

Robotic Platform Research

Explorative research into performance improvements for a greenhouse sensing robotic platform
Graduation Project, Technical Report



Author
Fesselet, L. (517107)

Date
08-04-2019

**Explorative research into performance improvements for a greenhouse sensing robotic platform
Graduation Project, Technical Report**

November 2018– April 2019

Published: T.B.D.

Abstract:

In the context of a graduation research at the faculty of Aeronautical Engineering, Inholland University of Applied Sciences Delft, this document reflects the process of answering the following research question:

How can HiPerGreen's pre-existing robotic platform be modified to effectively produce good imagery and be user efficient?

The answer is provided starting with a broad research into the environment of greenhouses and state-of-the-art robotic platforms. After an analysis of the current system, these requirements form the basis for a prototype design. The prototype in question is tested to validate meeting all requirements. This document is concluded with a set of recommendations for further research and development.

Feselet, L.		
		

Graduation Host: HiPerGreen
Graduation Assignment: HiPerGreen
Graduation Coordinator: M.J. Uiterwijk

Company Coach: A.H. Boode
Assignment Coach: C.J. Bes
Supervising Tutor: T. Hegberg

Cover image:
Courtesy of ADI

List of Abbreviations and Terms

ADI	Applied Drone Innovations
HiPerGreen	High Precision Greenhouse Farming
UAS	Unmanned Aerial System
R.S.	Rail System
PVC	Polyvinyl chloride
IMU	Inertial Measurement Unit
PWM	Pulse Width Modulation
CPU	Central Processing Unit
Ha	Hectares
DOF	Degrees of Freedom

List of Symbols

t	Time	\dot{Q}	Heat transfer
F	Force	T	Temperature
μ	Friction coefficient	ζ	Damping ratio
W	Power	δ	Damping delta
d	distance		
m	mass		
a	Acceleration		
c	Coefficient of efficiency		
Wh	Watt-hour		
ω	Angular frequency		
f	frequency		
A	Amplitude		
θ	Angle		
τ	Torque		
l	Length		
ν	Poisson's ratio		
E	Young's modulus		
J	Torsional Constant		
G	Modulus of rigidity		
π	Pi		
ω_p	Angular velocity of precession		
I_s	Moment of Inertia of spin		
ω_s	Angular velocity of spin		

Preface

I, Lucien Matola Fesselet, would personally like to thank M. Kampinga for his continued efforts in innovations which have led to the formation of HiPerGreen. I would also like to thank C. Heemskerk for the opportunities presented and the on-going guidance provided for the development of this assignment within HiPerGreen. I would like to send my regards and high esteems to C.J. Bes for the academic support provided. I would like to thank J. Westerhout for the contributed help during the design. Finally, I would like to thank my colleagues and friends C.C. Ramsay, W.G. Simmonds and T. Kearney-Michell for sharing and contributing the highs and lows throughout this graduation research.

I am a student from aeronautical engineering faculty from the Inholland University of Applied Sciences in Delft writing my graduation thesis. The degree covers a vast array of engineering disciplines including the chance to develop entrepreneurial skills. This is where Applied Drone Innovations was founded. In the spirit of innovation and being entrepreneurial, the opportunity was taken to expand the field of knowledge regarding the using robotic technologies inside greenhouses, leading to the creation of the HiPerGreen project.

This report is intended to the graduation examination board for assessment. It is also intended to be read by future students and participating entities within the HiPerGreen project. Any other persons are free to read, quote and refer this document for information, with future development in mind, to study material or simply to feed genuine curiosity.

Summary

Greenhouses are getting bigger, forcing a gap between grower and crop. Having to rely on manual labour, inspecting the crop is a strenuous and repetitive task, making the job undesirable. Outdoor agriculture is benefiting vastly from the use of Unmanned Aerial System to image crop, however, this technology is yet to add value inside greenhouse. HiPerGreen, a research project within InHolland University of Applied Sciences, has been exploring the use of a rail based robotic platform to image crop. Rail based platforms are already in use throughout horticulture, these can be seen to carry humans above crop or to transport watering system throughout greenhouse compartments. Naturally, HiPerGreen developed designs based on the status quo. However, the products suffered from oscillations rendering images taken too blurry for any good analysis or were simply too large for effective use and transport. Throughout the beginning of this document, several methodologies are considered, such as deblurring technics based on inertial movement readings or utilising known systems such as 3-axis gimbals, but ultimately, a more adequate design is chosen as methodology with the aim of reducing oscillations within the system.

Requirements are laid out and confirmed with HiPerGreen to ensure a proper design within the project scope. A clear three stage design can be derived: mechanical, electrical and software. All three aspects need to work together to produce the best crop imagery. Firstly, a trade-off is carried out over three mechanical skeleton concepts. These concepts are compared on parameters derived from the requirements. The winning concept is a relatively small and compact design theorised to have minimum oscillation. A more detailed section of the document describes in more depth the behaviour of the mechanics, as well as a re-design of the mounting and fixation of the wheels and motor unit. All these methods are passive methods to counter oscillations, but a proposal for using an active method is outlined: the usage of gyroscopic precession. By having a rotating mass, rotational force around an axis take effect at a 90-degree phase, a phenomenon that can be taken advantage of.

Usability improvement have been made on the electronics and software. Previous version of the robotic platform used a step by step approach where the user would have to input one by one each individual desired movement. For the new design, a more direct approach is taken: the user has a remote that dictates directly and proportionally desired movements. To achieve this, a radio-based transmission is established, with a controller that can accurately and timely translate the user's inputs into the motor. Limits are set to ensure a maximum speed in movement is not exceeded. Due to the limitation of the computing hardware, a variety of improvement were necessary to optimise the software to run as smoothly as possible. The electronic components are encased for protection and heat dissipation is considered and improved after reports of overheating on previous HiPerGreen rail based systems.

Once the design done and built, tests are carried out to verify and validate the solution. Tests on oscillatory motion and related images were carried out on the previous HiPerGreen design, from which a benchmark is derived. An algorithm is developed to determine a blurriness index within images, from which a minimum index is selected. The new design is subjected to the same tests, namely having to go through the various excitation scenarios. Data is recorded each time, studying the oscillatory response and imagery taken.

Based on the results of the tests, it can be concluded that under all scenarios formulated, the images taken always had a blurriness index above the minimum established. Moreover, test user reported a high ease of use of the product. Recommendation are established in research into reducing the weight of the product for further ease of use, as well as reducing manufacturing costs. In conclusion, the mechanical, electrical and software improvements have yielded a more stable, compact and user-friendly product.

TABLE OF CONTENTS

1	INTRODUCTION	2
1.1	Context	2
1.2	Problem Definition	2
1.3	Research Goal	3
1.4	Methodology	3
1.5	Report Structure	3
2	RAIL SYSTEMS & ANALYSIS	4
2.1	Operational Environment	4
2.2	State-of-the-Art Rail Systems	7
2.3	HiPerGreen Rail System Analysis	9
2.3.1	Rail System V1	9
2.3.2	Rail System V2	12
2.4	Imagery Analysis	17
3	SYSTEM REQUIREMENTS	20
3.1	Requirements	20
4	CONCEPTUAL SYSTEM CONFIGURATION	22
4.1	Criteria for Trade-Off	22
4.2	Configuration Concept 1	24
4.3	Configuration Concept 2	25
4.4	Configuration Concept 3	27
4.5	Trade-Off	28
5	DETAILED MECHATRONICS DESIGN	30
5.1	Mechanical Design	30
5.1.1	Oscillatory Motion	30
5.1.2	Gyroscopic Precession	33
5.1.3	Re-design of the Wheels	33
5.2	Electronic Design	34
5.2.1	Computing Unit	34
5.2.2	Communication	35
5.2.3	Controlled Movement	35
5.2.4	Overall Circuit	39
5.3	Software and Control Design	39
5.3.1	Control Approach	40
5.3.2	Software Architecture	40
5.3.3	Run Time Improvement	42
6	TESTING & VALIDATION	44
6.1	Testing of V3	44
6.1.1	Setup	44
6.1.2	Test	44
6.1.3	Results	45
6.2	Validation	48
7	CONCLUSIONS & RECOMMENDATIONS	50
7.1	Conclusion(s)	50
7.2	Recommendations	51

8 REFERENCES52

TABLE OF APPENDIXES

APPENDIX A TESTING FOR OSCILLATION OF R.S. V2.....54

A.1 Setup54

A.2 Test54

A.3 Result54

A.4 Analysis56

A.5 Source code of the microcontroller61

A.6 Snippet of the computer program for reading and saving the incoming data63

A.7 Snippet of the data recorded64

APPENDIX B LAPLACIAN VARIANCE COMPUTING SOURCE CODE65

APPENDIX C DATA TYPES IN C++66

APPENDIX D SOURCE CODE FOR R.S. V367

LIST OF FIGURES

Figure 1 Schematic of pipe hook at Hazeu, measurements in [mm] 5

Figure 2 Kink in heat pipes due to welding 5

Figure 3 Picture showing proximity of lamps 6

Figure 4 Cartesian Axis within greenhouse 6

Figure 5 Simple rail system 7

Figure 6 Picture of a spraying boom (Robur, 2019) 8

Figure 7 Picture of human boom (De Vette) 8

Figure 8 Picture of human boom (Robur, 2019) 9

Figure 9 HiPerGreen R.S. V1 CAD Model 10

Figure 10 Exploded View of Motor Unit, R.S. V1 11

Figure 11 Picture of HiPerGreen Rail System V1 11

Figure 12 Testing of R.S. V1 in Greenhouse 12

Figure 13 Side view of R.S. V1 wheel and R.S. V2 wheel 13

Figure 14 Motor unit R.S. V2 13

Figure 15 Motor housing R.S. V2 13

Figure 16 R.S. V2 prototype 14

Figure 17 Electrical Schematics of R.S. V2 15

Figure 18 Schematics of Software flow of R.S. V2 15

Figure 19 Timeline of synchronisation steps between Controller and Actuator 16

Figure 20 Schematic of Motion Blur (Stripek, et al., 2012) 17

Figure 21 Blurriness Comparison	18
Figure 22 Laplacian Kernel (Rosebrock, 2015)	18
Figure 23 Blurriness Comparison with Blur Index	19
Figure 24 Deblurring based on IMU. Left: Original image. Right: Deblurred image, with corner deblurring kernels (Joshi, et al., 2015)	19
Figure 25 Schematic of Configuration Concept 1	24
Figure 26 Schematic of Configuration Concept 2	25
Figure 27 Schematic of Configuration Concept 3	27
Figure 28 – Simple Harmonic Motion (Nave, 2019)	30
Figure 29 Schematics of Gyroscopic Precession (Maaz, 2016)	33
Figure 30 New wheel mount	34
Figure 31 Schematics of a PWM signal (Android Developers, 2016)	35
Figure 32 Sequence of stepper motor (Dejan, 2019)	36
Figure 33 Stepper motor sequence of coils in Single-coil Excitation (Dejan, 2019)	36
Figure 34 Stepper motor sequence of coils in full step drive (Dejan, 2019)	36
Figure 35 Stepper motor sequence of coils in half step drive (Dejan, 2019)	37
Figure 36 Stepper motor sequence in coils in microstepping (Dejan, 2019)	37
Figure 37 Diagram of Stepper Motor Drive (electronics-course.com, 2019)	38
Figure 38 Electronic schematic proposition	39
Figure 39 photo of electronics box	39
Figure 40 Software flow diagram for R.S. V3	40
Figure 41 PWM Signal processing time comparison schematic	43
Figure 42 IMU data after induced oscillation around z axis	45
Figure 43 Image taken during the oscillation, with blur index of 502	46
Figure 44 Gyroscope data of platform passing over a welding point around Y axis	46
Figure 45 Gyroscope data of platform passing over a welding point around Z axis	47
Figure 46 Image taken during the oscillation of the second wheel passing over the welding point, with blur index of 575	47
Figure 47 Graph of Amplitude over time	48
Figure 48 Graph of gyroscope around Z-axis, placed at the bottom	55
Figure 49 Graph of gyroscope around Z-axis, placed at the top	55
Figure 50 Graph of compass in X direction, placed at the bottom	56
Figure 51 Graph of example for Savitzky-Golay filtering result	56
Figure 52 Graph of gyroscope around Z-axis, placed at the bottom, in the time domain and frequency domain	57
Figure 53 Graph of gyroscope around Z-axis, placed at the top, in the time domain and frequency domain	58

Figure 54 Graph of compass in X direction, placed at the bottom, , in the time domain and frequency domain 59
Figure 55 Amplitude vs time, bottom and top..... 60

LIST OF TABLES

Table 1 Heat Pipe Specification 4
Table 2 List of Requirements 20
Table 3 Trade off table 28
Table 4 Zeta Calculation R.S. V3 48
Table 5 Requirement compliance table 49
Table 6 Zeta Calculation R.S. V2 61

1 INTRODUCTION

The background of this graduation research project is outlined in section 1.1, from which a problem definition is then described regarding technical aspect in section 1.2. The main research question and sub-questions are then formed in section 1.3. An insight into how this research was carried can be found in section 1.4. Finally, an overview of the chapters and readability can be found in section 1.5.

1.1 Context

The average area cultivated area under glass in a single greenhouse has grown from 1.9 ha to 10 ha between 2004-2006 (Breukers, et al., 2008). This is mainly due to a higher concentration of greenhouses hectareage per company, for example in 2000 there were approx. 100 million ha spread across 11000 companies, by 2016, 92 million ha spread across 3800 companies (Centraal Bureau voor de Statistiek , 2019). Greenhouses are getting bigger, forcing a gap between grower and crop. Having to rely on manual labour, inspecting the crop is a strenuous and repetitive task, making the job undesirable. This can be seen by labour being hard to find, hard to train and results are subjective, and quality is very subjective (de Beer, 2018). This calls for an increase in data driven decision making and automation within horticultural processes.

Outdoor agriculture is benefiting vastly from the use of Unmanned Aerial System (UAS) to image crops, however, this technology is yet to add value inside greenhouse. HiPerGreen (High Precision Greenhouse Farming), a research project within InHolland University of Applied Sciences, has been exploring the use of a rail based robotic platform to image crops (Lalwani, 2019).

Rail systems are used extensively throughout horticulture, these can be seen to carry humans above the crop at Ter Laak (Robur, 2019) and used as a logistics tool for removing ill crops at Hazeu Orchids (Figure 5). These systems can provide design inspiration but are not yet suitable for autonomous remote imaging of crops. Rail systems offer a less complex means of roving above crop and could easily be implemented as a standardised way of pushing the industry towards data driven crop growth market.

1.2 Problem Definition

In order to use a rail based robotic platform inside a greenhouse it is necessary to take into consideration the operational environment, this includes geometry of the heating pipes from which the system is supported, the temperature of the heating pipes, humidity of the air and other greenhouse obstacles are also important. Once the operational environment is clearly defined and understood it is possible to set specific requirements for the rail based robotic platform.

The aim of the rail based robotic platform is to capture images. These images are used for analysis of the crop. It is important to retain a maximum quality within the image. However, the rail based robotic platform is a moving platform. The more movement there is when a camera is taking a picture, the blurrier the resulting image. It is therefore critical to keep movement to a minimum when a picture is taken. To achieve this, several challenges need to be overcome.

The first main technical challenge regards the mechanical structure, considering the rail system is a moving system with rotary components, it is essential to have sufficient structural integrity in all axles and bearing housings. The rail system is balancing on its wheels, therefore, centre of gravity and balancing of the system is important for stability. Dynamic stability is also of concern due to oscillatory motion of the system which can be introduced by the drive train, by welds in the heating pipe, or many other external or internal excitation forces.

Once there is a robust mechanical base for the robotic platform the next technical hurdle is regarding electronics. The system must be powered, communication to the system must be

possible and of course function in a controlled manner. This must operate in a high humidity, high temperature environment and safety must be insured.

Finally, to be used in various greenhouses, the system needs to be directly controllable with little to no latency and designed in such a way to be as easy to use as possible.

1.3 Research Goal

Considering the problem definition outlined in section 1.2, this research can be concluded by answering the following main research question:

Main Research Question

How can HiPerGreen's pre-existing robotic platform be modified to effectively produce good imagery and be user efficient?

In order to answer the main research question, the following sub research questions have been formulated. It is necessary to fulfil these project objectives if a conclusion is possible within the timeframe of this project.

Sub Research Questions

- I. What are the restrictions and limitations imposed by the operational environment?
- II. How are state-of-the-art robotic platforms currently used in greenhouses?
- III. What are the requirements of the robotic platform?
- IV. What is the current state of HiPerGreen's robotic platform?
- V. What structural mechanic modifications are required?
- VI. What electronics and software modifications are required?
- VII. How can the system be validated?

1.4 Methodology

Tackling these research questions begins with a literature study, this study is regarding the horticultural industry, state-of-the-art rail systems, HiPerGreen's current system, vibrations and oscillatory motion, electronics, software and control-software. Once the literature study is concluded an analysis of HiPerGreen's existing system is carried out, including system tests. From this initial literature study and testing of the existing system it is possible to define all requirements. After requirements have been established work can begin on conceptual configuration designs, the most appropriate configuration will determine the configuration to be detailed in this research. The detailing of the system regards mechanical aspects, electronics, software and control. Finally, it is possible to test and validate the research findings and detailed mechatronic design.

1.5 Report Structure

After being introduced to the thesis in section 1, the reader can find a study into state-of-the-art rail systems and an analysis of HiPerGreen's existing system in section 2. Section 3 sets out the technical requirements and wishes for the system. This sets the stage for generating conceptual designs in section 4. A trade-off is executed between the conceptual designs and the winning configuration is further detailed in section 5 on mechanical aspect, electronics aspect and finally on software. In order to check the assumptions and design choices made throughout this thesis, section 6 details the tests performed for validation. Finally, the thesis is concluded in section 7 with recommendations.

2 RAIL SYSTEMS & ANALYSIS

After the introducing the research project in section 1, section 2 establishes the environment in which the robotic platform is operating and overviews current systems. This is done through an introduction to the physical environment in section 2.1. Followed by a presentation of the relevant state-of-the-art machinery in the horticultural industry in section 2.2 before finishing on an in-depth analysis of the current systems built by HiPerGreen in section 2.3 and an analysis of the imagery in section 2.4.

2.1 Operational Environment

Rail system run along heat pipes installed near the roof inside greenhouses. Their primary function is to regulate the temperature inside compartment, with water at various temperature running inside it. Depending on the greenhouse and the type of crop grown, the water inside the pipes can go up to 70 [°C] (Brander, 2018).

After carrying out measurements in greenhouses, it can be concluded that pipes are made out of steel, with an external diameter ranging from 57 [mm] to 76 [mm], laid in a “S” pattern in the horizontal plane hanging from the roof. The hooks are *welded to the roof* structure, but the pipes are “just sitting” in the hooks, using gravity to keep it in place. A schematic can be found in Figure 1 with dimensions. The pipes are generally always at the same distance from the roof; however, the height of the roof can vary significantly. In low roof compartment, such as the buffer room in Hazeu, the pipes are at 1.5 [m] from the walking level. Compared to the high roof compartment in the same company, the pipes are at 4.8 [m] from the walking level.

There is limited space to be used around the pipes. All of the hooks are laid in the same direction. As can be seen in Figure 1, the hook itself protrude on both side of the pipe. Moreover, there is a variety of obstacle near the pipes that need to be taken into account, such as sprinklers, lamps and fans. Measuring at Hazeu on the open side of the hook, lamps are located at 250 [mm], sprinklers at 600 [mm] and fans at 310 [mm]. A picture of the pipes and objects can be seen in Figure 3.

It is important to know the climate greenhouses are operating as electronics are susceptible to climate parameters, namely temperature and humidity. However, due to the nature of these parameters (the best climate setting for plant growth is confidential company knowledge), it has not been permitted to disclose the information, but a range can be determined. Regarding temperature, a range between 5 [°C] and 35 [°C] is assumed; and for humidity a range between 20 [%] and 99[%] relative humidity is assumed.

All data is summarised in Table 1.

Table 1 Heat Pipe Specification

Heating pipe diameters	57mm – 76mm
Heating pipe temperature	70 °C
Environmental temperature	5 °C – 35 °C
Environmental relative humidity	20% - 99%
Heating pipe height from ground	1.5m – 4.8m

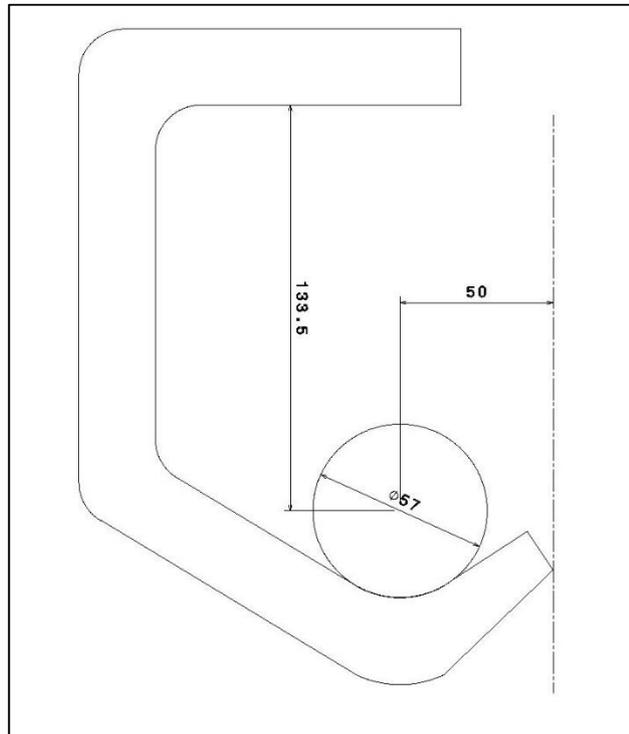


Figure 1 Schematic of pipe hook at Hazeu, measurements in [mm]

Another aspect to be considered is the nonuniformity of these pipes. During installation, the hooks can be placed with irregularities, this can lead to the pipes not being entirely straight. Moreover, the pipes are assembled during installation and welded together, leaving a welding “kink” that can introduce disturbance on a rolling wheel, as can be seen in Figure 2. The pipes undergo severe temperature changes, ranging from 70 [°C] to 10 [°C]. These variations introduce expansion and contraction of the pipes. The pipes are also never cleaned, leaving a layer of dust on the top side, which can introduce slippage, as shown in Figure 3. Lastly, common land subsidence in the Netherlands changes slowly the structure of the greenhouse (Meijer, 2018).



Figure 2 Kink in heat pipes due to welding



Figure 3 Picture showing proximity of lamps

Throughout the report many directions are discussed, such as direction of oscillation. It is therefore important to define a constant axis system. An x,y,z cartesian system is used, with x pointing perpendicular the heating pipes, y pointing parallel to the heating pipes and z going out of the figure, starting on row 1 table 1, as illustrated in Figure 4, and following the right hand rule.

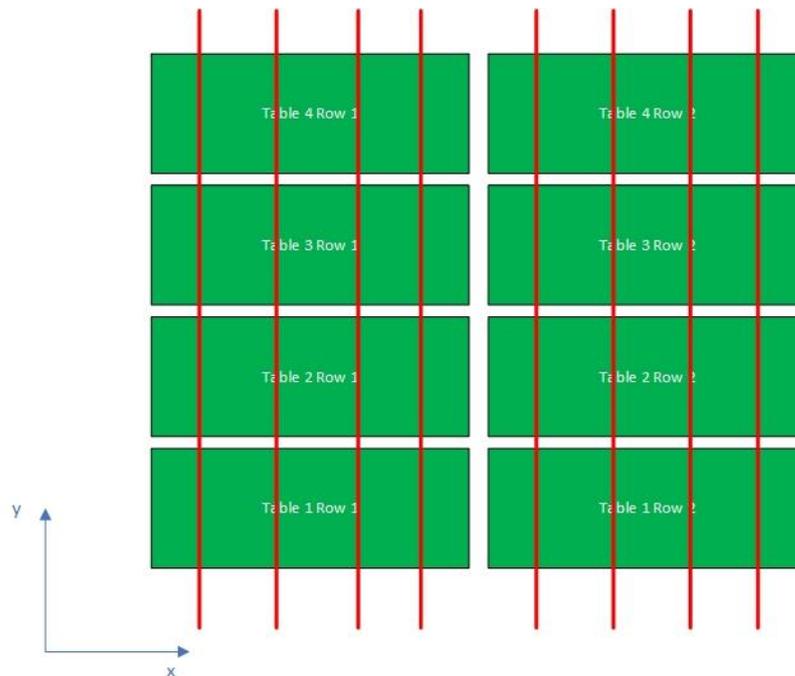


Figure 4 Cartesian Axis within greenhouse

2.2 State-of-the-Art Rail Systems

The horticultural industry already uses the heat pipes to transport systems across compartments. These systems range from simple manual platform to transport small items to full bridge certified for 450 kg capable of moving people across huge fields. This sub-chapter show cases some examples.

The simplest form of such platform is a simple mono-rail platform used as a logistical tool for removing ill crops. The scouts drag it with them along to place ill plants. Such system can be seen in Figure 5. It is very lightweight and is handled manually to install it from one rail to the other.



Figure 5 Simple rail system

The most common platform is the spraying boom, as can be seen in Figure 6. A heavy platform, usually hanging from more than one heat pipe traverses a lane in one go, spraying fluid on the crop. Some come with its own container, others have a reeling system to hook the nozzles to a central fluid unit hundreds of meters away from the system. In both cases, a so called “mother ship” is always present on the end of the heat pipes that can transport the spraying system from one alley of crop to the next one.



Figure 6 Picture of a spraying boom (Robur, 2019)

Certain greenhouses highly optimise the space for plants, leaving little to no space for humans to walk around. This leads to needing machines to transport humans around. Such an example can be seen in Figure 7 and Figure 8. These machines are used to monitor the crop and infrastructure of the greenhouse, are semi-automated and are certified to carry over 450 [kg].



Figure 7 Picture of human boom (De Vette)

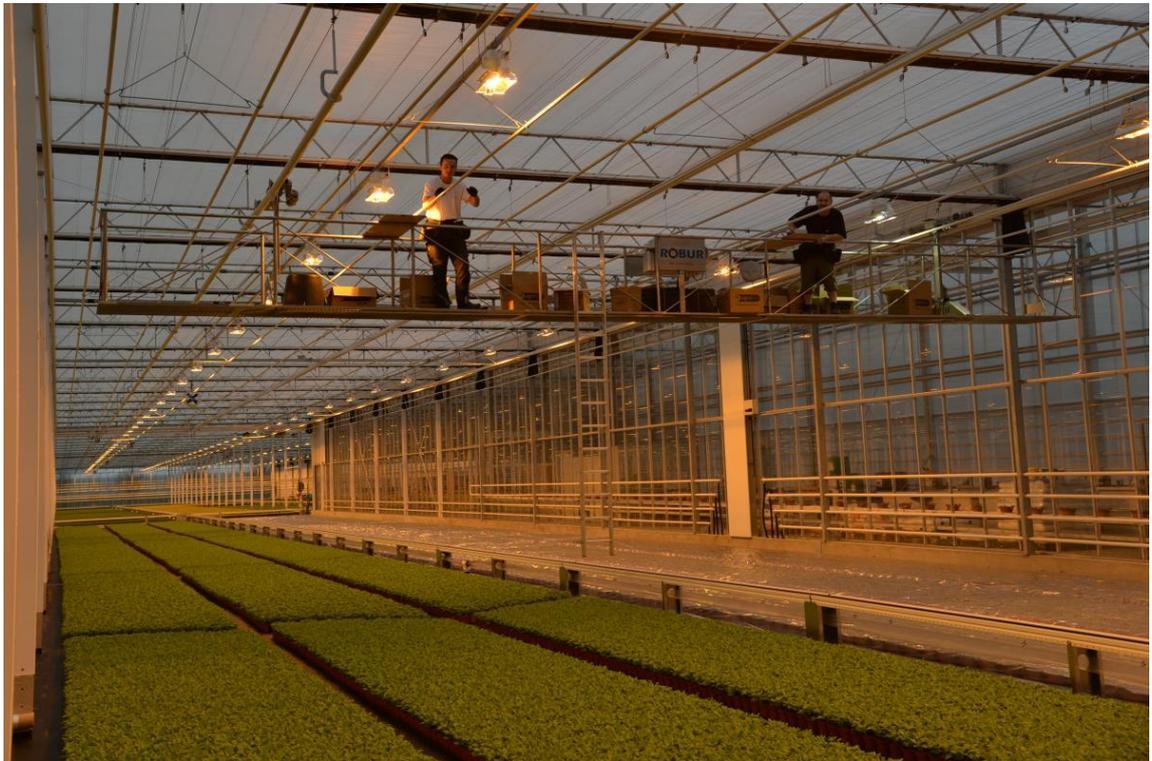


Figure 8 Picture of human boom (Robur, 2019)

2.3 HiPerGreen Rail System Analysis

HiPerGreen has already been through 2 iterations of a rail system. The first iteration, Rail System V1, was developed by Bernard Srivastava, 3rd year internship during the period of February 2018 to July 2018 (Srivastava, 2018). The second iteration was developed by Mohit Lalwani, 3rd year internship and HiPerGreen's researcher W.G. Simmonds during the period of September 2018 to February 2019 (Lalwani, 2019).

The second iteration development period overlaps the development period of this graduation assignment by two and half months. This allowed for more extensive analysis with various tests on the V2 compared to V1. Moreover, V1 was scrapped for parts to build V2, rendering tests impossible. All information on V1 is derived from testimonials and reports.

2.3.1 Rail System V1

Rail System V1 was the first attempt to build a robotic platform. It follows the construction of a hanging bridge, as can be seen in Figure 9, comprising of a lateral beam; a payload cart capable of going along the lateral beam; vertical skis with motor units on the top. The system was at a total length of 6 [m] and a height of 2.5 [m].

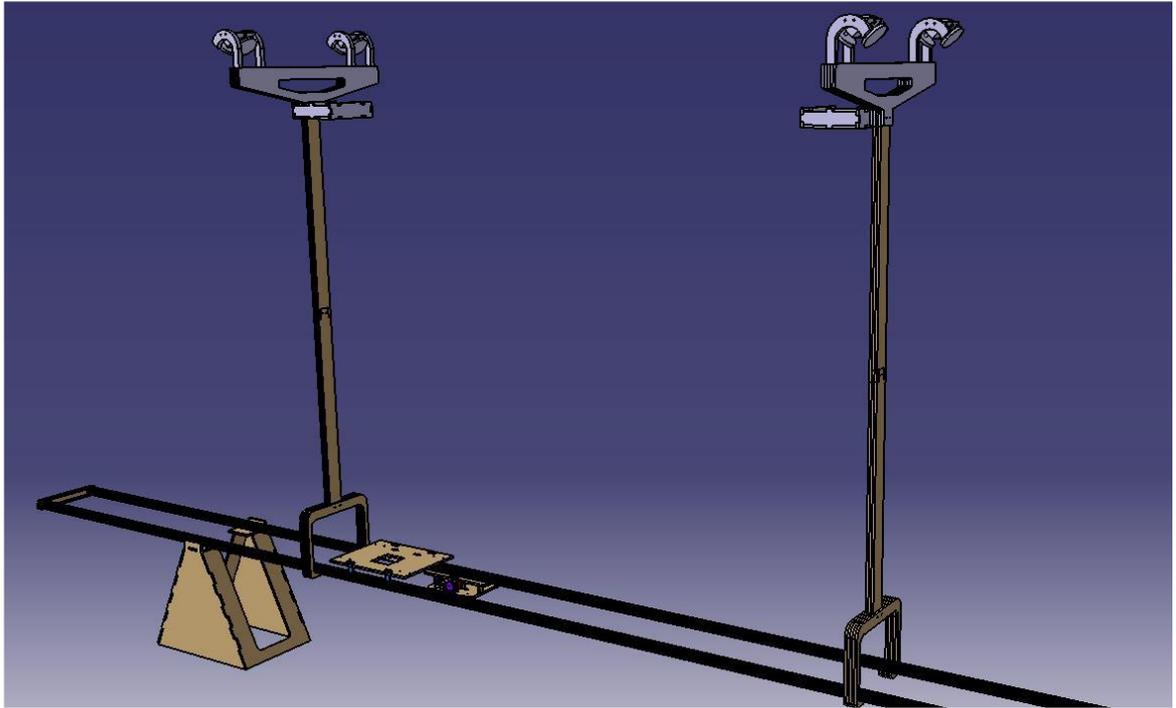


Figure 9 HiPerGreen R.S. V1 CAD Model

The lateral beam is made of a set of two parallel beams made of three 2 [m] Bosch aluminium profiles for transportability, interconnected by an aluminium bar. This results in hanging ends of the lateral beam. To remedy this, strings are attached in between the skis, and between skis and ends of the lateral beam, as can be seen in Figure 11.

The payload cart is where sensory equipment is placed (e.g. cameras, temperature sensor etc..) and is capable of moving along the lateral beam, placing itself anywhere above crop table. This introduces many mechatronic challenges: synchronisation of forward motion of the whole rail system with the lateral motion of the payload cart; powering everything from the same battery; countering voltage drops on relatively low voltages across long distances; complexity of accurate positioning of the payload cart on the lateral beam.

The motor units on top of the skis were inspired by the simple rail system described in section 2.2 (Figure 5), where the wheels are in a “V” configuration, shown in Figure 10. The wheels are manufactured with the additive manufacturing process (commonly known as 3D printing), making them susceptible to interlayer shear, especially in the stress concentrated zone between the shaft and the wheel itself.

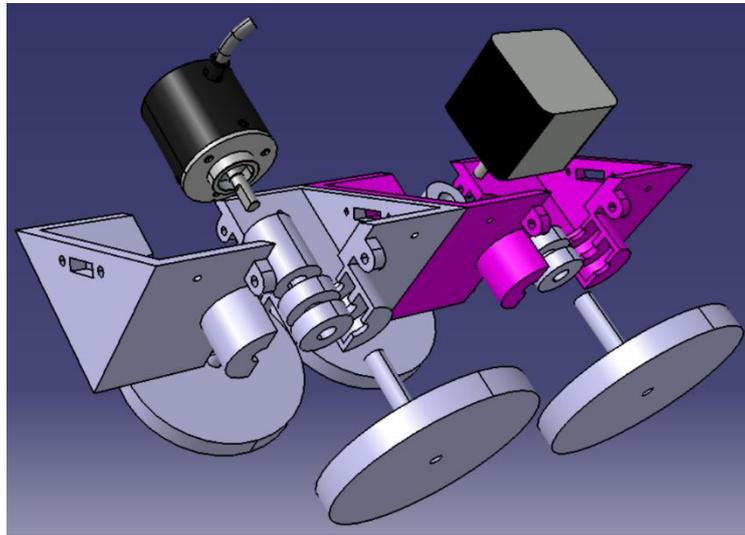


Figure 10 Exploded View of Motor Unit, R.S. V1

A prototype was built and can be seen in Figure 11.



Figure 11 Picture of HiPerGreen Rail System V1

Testing was attempted inside a greenhouse. The prototype did not get to be tried due to practicality issues. The system was too heavy, too long and too complex to be comfortably manoeuvred in place. The motor units were damaged during installations of skis onto the heat pipes (inter layer shear on the wheel axle; motor housing breaking in half; wires unplugged and broken in half), nuts and bolts were lost after dropping them or slipping out of hands, staff were uncomfortable having to balance themselves and the prototype standing on precarious surfaces. Figure 12 shows the precarity during assembly of R.S. V1 in the greenhouse.



Figure 12 Testing of R.S. V1 in Greenhouse

Clearly, there is a need to re-design the system. This is where M. Lalwani's assignment starts and is discussed in section 2.3.2.

2.3.2 Rail System V2

The analysis of R.S. V2 is split into two parts, a mechanical analysis and an electronics and software analysis.

Mechanical Analysis

R.S. V2 design aimed for the system to perform better than the previous one, from the issues outlined in section 2.3.1. The first big modification is the removal of the payload cart. By placing the remote sensors at a higher level above the crop, enough area can be observed with enough resolution, and thus eliminating the need for a complex lateral movement mechanism.

The second big modification happened in the wheel configuration, the motor unit and the motor housing. The "V" configuration of the wheels is replaced by a concave "diabolo" wheel. This allows for a sturdy and strong construction with the limited space allowed. A schematic can be found in Figure 13. The motor unit is re-designed accordingly to the new wheel configuration, as can be seen in Figure 14. A sturdier motor mount is made of laser cut wood instead of tie-rop and a bigger spacing between the bearings is introduced to reduce moment on the shaft. Finally, the whole assembly is re-designed to fit within the limited space due to the hooks and other obstacle. The aim of the assembly is to keep the weight vector aligned with the central axis of the pipe. A complete schematic can be found in Figure 15.

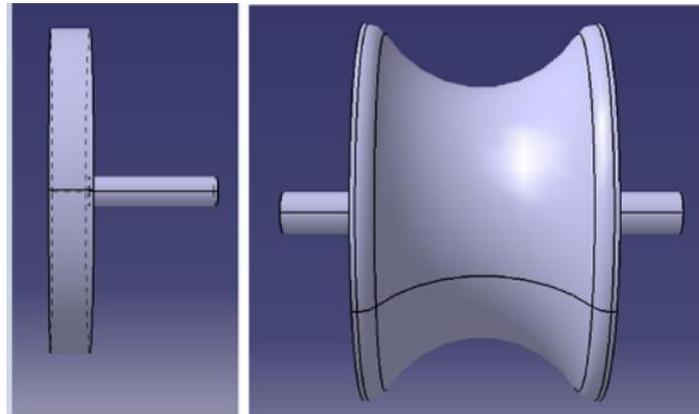


Figure 13 Side view of R.S. V1 wheel and R.S. V2 wheel

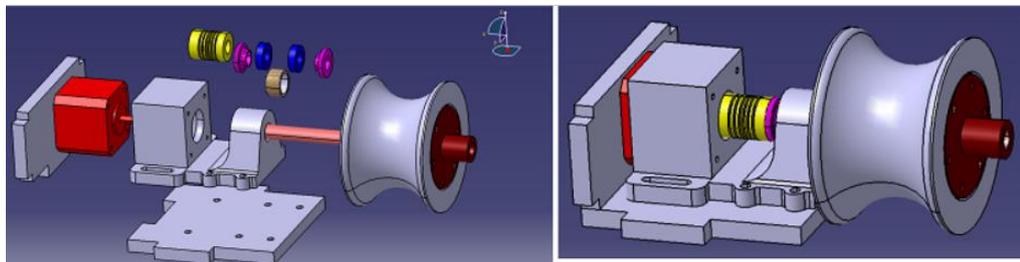


Figure 14 Motor unit R.S. V2

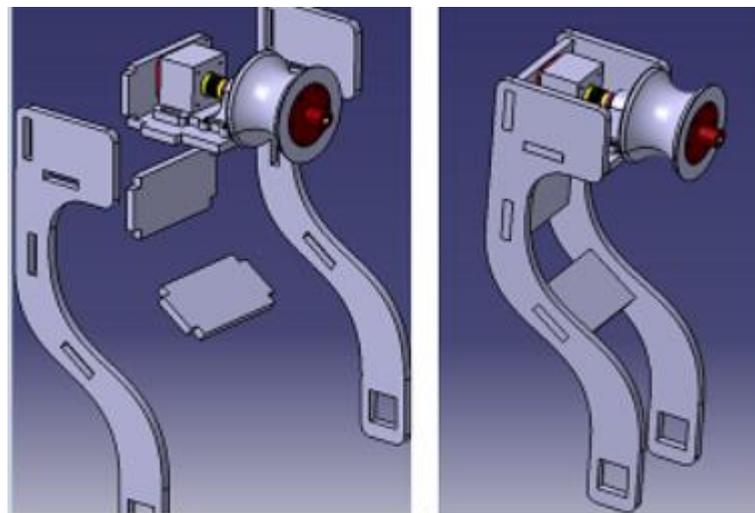


Figure 15 Motor housing R.S. V2

Two motor housing are made, one powered and the other one passive. They are connected by a beam, from which a vertical beam is attached. At the bottom a lateral beam is attached, onto which sensor packages can be attached at various location. A picture of the final prototype can be found in Figure 16.



Figure 16 R.S. V2 prototype

During testing, oscillations were observed around the vertical axis. These oscillations occurred by external excitation mainly caused by un-even pipes and obstacles on the pipe (i.e. dust and welding joints). Although the aforementioned situation induced small amplitude oscillation, human interaction caused the biggest oscillations, whilst installing or trying to adjust the system manually. Further investigation is performed for understanding. The investigation done through testing is explained in detail in APPENDIX A.

The conclusion of the test reveals that most of the oscillation is happening within the vertical beam.

Electrical and Software Analysis

The mechatronics to enable movement consists of two parts: the controller and the actuator. It is derived from the electronics on board the R.S. V1. Since V1 had never been dynamically tested, no reporting on the performance can be found. V2 follows the same principle, where controlling and actuating units are communicating wirelessly. After interviewing the makers of V1 (Srivastava, 2018), the wireless configuration was to eliminate bad communication over relatively long distances – over 6 [m] from controlling unit to actuating unit makes the wire act as an antenna and voltage fluctuation interfere with communication protocol chosen – and to offer possibility to add any number of actuating unit based on one controlling unit. A schematic of the overview on the electrical system of V2 can be found in Figure 17.

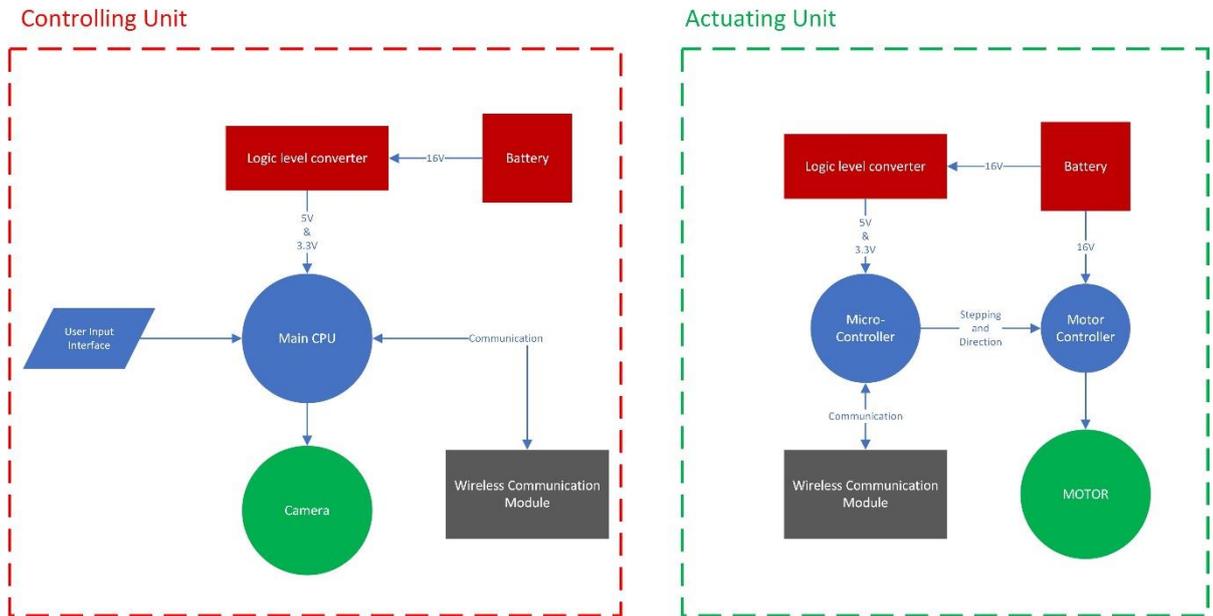


Figure 17 Electrical Schematics of R.S. V2

This configuration results in a certain level of complexity. Each module needs a separate power source and power regulation, with additional weight. Each module has its own computing units where both need to work in a synchronic manner, as reflected from the software flow diagram in Figure 18. Moreover, the actuator module computing unit is a micro-controller, which is not capable of multi-tasking. From a practical view, this translate to whenever the whole system starts to move, there is no way of interrupting the system, or to include a safety into the system.

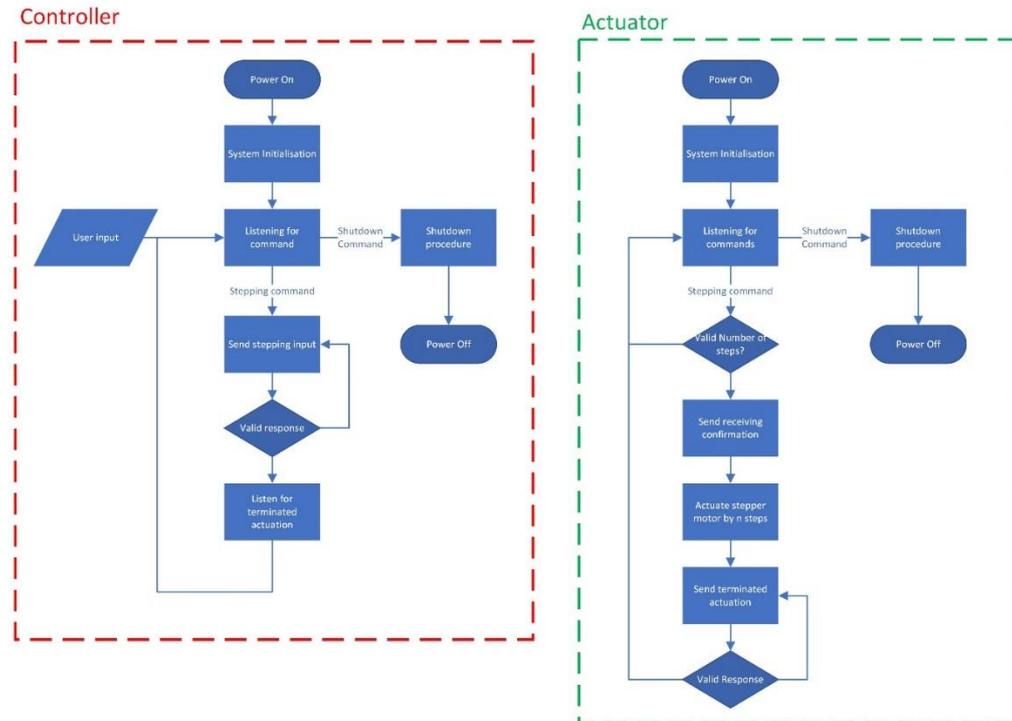


Figure 18 Schematics of Software flow of R.S. V2

The system works with user input commands, with two choices: “move” by n number of steps and “shutdown system”. These commands are given through a written command, where the user needs to type letter by letter the command, e.g. “move 12.34” where 12.34 is the distance in [m]. With the given radius of the drive wheel, the controller module converts the distance into a number of steps and forward it to the actuator module. Once the actuator module confirms receiving correctly the number of steps, it proceeds to step. A timeline schematic can be found in Figure 19.

This methodology implies that the user knows precisely the distances he wants the system to move by. It also offers no response for correction during movement, which during testing was quickly seen as a wasting time element.

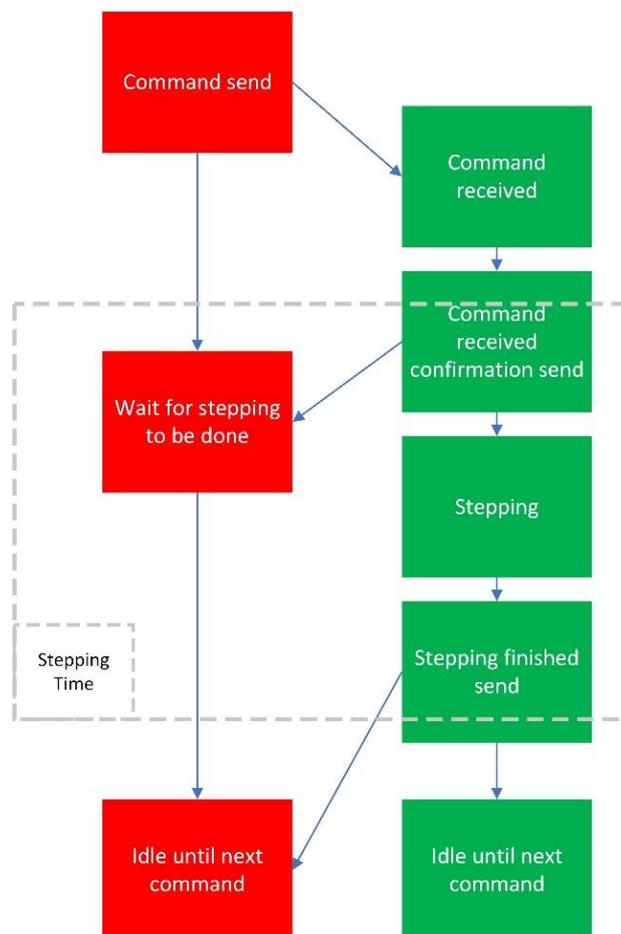


Figure 19 Timeline of synchronisation steps between Controller and Actuator

Another problem outlined is the continuity of operation of the system. The motor controller has been tested extensively in office, with no problems. However, after 5 to 8 minutes of continuous use in the greenhouse, it worked intermittently, with 20 seconds of working before stopping for 1 minute to 1 minute and a half. The data sheet suggests adding a heatsink for any current above 1 [A]. Although there is already in place a passive heatsink, it has been suggested to increase the heat dissipation for continuous use of the current setup.

2.4 Imagery Analysis

The rail system is equipped with a high-resolution camera to image the crop. The images collected is then processed to find abnormalities in the crop. However, due to the workings of a camera, fast relative movement between the camera and the subject photographed yields blurry images.

To understand blurriness in images, it is necessary to understand how cameras work. A camera has a light sensitive sensor, composed of pixels. Generally speaking, a camera works as follows: once triggered, the aperture opens and lets light hit individual pixels. After a certain amount of time, the aperture closes again. While the aperture is open, pixels act as “buckets of light”, transforming the amount of light into a digital number representing the amount of light. With pixels arranged such to be sensitive to certain colours, an image can then be re-constructed based on the numbers of each individual pixels.

To work in different environment, the exposure of the light sensitive sensor can be modified on three levels: the aperture, the shutter speed and the ISO. The aperture can be modified to be bigger or smaller, letting more or less light at once onto the sensor. The shutter speed dictates how long the sensor is exposed to light. ISO influences how sensitive the sensor is to light.

The aperture of the camera selected for the rail system is fixed. Increasing the ISO makes the sensor more sensitive to the incoming light, but increases the noise levels as well, making picture appear “grainier”. It is therefore always set to as low as possible. The last parameter left is the shutter speed. Finding the right shutter speed is a thin line to have correct, too low and the images will be too dark, too high and motion blur will render images useless.

Image blur is a phenomenon where objects or scenery appears blur due to motion during exposure time. An example image can be found in Figure 20. For a given shutter speed, the bigger the motion the bigger the blur.

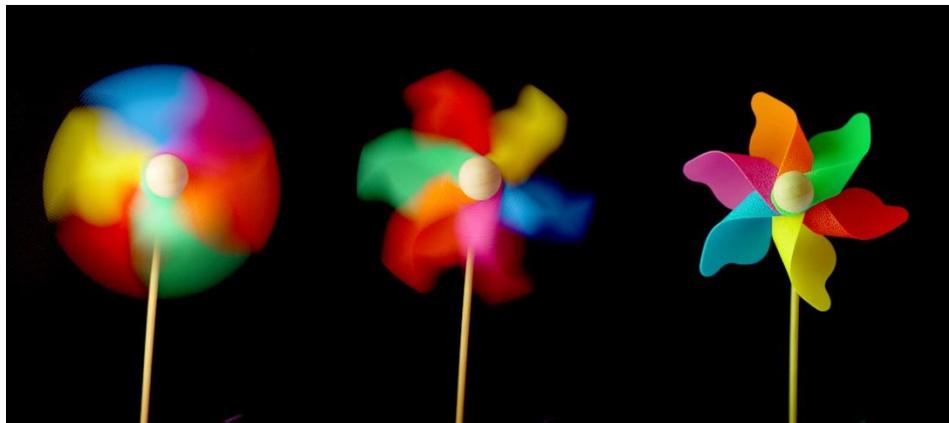


Figure 20 Schematic of Motion Blur (Stripek, et al., 2012)

The motion blur occurs also if the camera itself is in motion. That said, the rail system is continuously in movement leaving little room for extra movement. When stable, images are decent enough, however, after excitation and when in oscillation, the relative movement of the camera renders the images blurry, too blurry for analysis, as shown in Figure 21. From the figure, image c) is taken at rest, image b) is taken during oscillation, and image a) is purposefully over-blurred for understanding.



a) very blurry

b) moderate blurry

c) none blurry

Figure 21 Blurriness Comparison

It is important to be able to quantify the blurriness. A good measure is the variation of Laplacian transform in an image, discovered by Pech-Pacheco et al. where, as Adrian Rosebrock explains:

“You simply take a single channel of an image (presumably grayscale) and convolve it with the following 3 x 3 kernel (Figure 22)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Figure 22 Laplacian Kernel (Rosebrock, 2015)

If the variance falls below a pre-defined threshold, then the image is considered blurry; otherwise, the image is not blurry.

The reason this method works is due to the definition of the Laplacian operator itself, which is used to measure the 2nd derivative of an image. The Laplacian highlights regions of an image containing rapid intensity changes, much like the Sobel and Scharr operators. And, just like these operators, the Laplacian is often used for edge detection. The assumption here is that if an image contains high variance then there is a wide spread of responses, both edge-like and non-edge like, representative of a normal, in-focus image. But if there is very low variance, then there is a tiny spread of responses, indicating there are very little edges in the image. As we know, the more an image is blurred, the less edges there are.” (Rosebrock, 2015)

The source code to compute it can be found in APPENDIX B. The program gives a single number representing the variance across the image. After showing and discussing with HiPerGreen’s biologist T. Kearney-Mitchell, the threshold has been established at 300. Below can be found example images with their indices in Figure 23.



Figure 23 Blurriness Comparison with Blur Index

Three ways can be foreseen to reduce blurriness: a software approach, using a gimbal or a mechanical approach. The software approach would be using inertial measurements while the camera is moving, and post-process the images based on the motion recorded during the exposure. This technic has been tried by Joshi N. et al. Although showing good results, there are still some artefacts left in the resulting image: “*There is still some residually ringing that is unavoidable due to frequency loss during blurring.*” (Joshi, et al., 2015). Example image can be found in Figure 24.



Figure 24 Deblurring based on IMU. Left: Original image. Right: Deblurred image, with corner deblurring kernels (Joshi, et al., 2015)

Unfortunately, these small artefacts are too much for plant analysis, and the potential for confusing the analysis program is too high.

A gimbal is a pivoted support that allows the rotation of an object about one axis. On a more common practise in photography, a 3-axis gimbal is a device that ensure the orientation and attitude of a camera to always be the same. 3-axis gimbals are very popular to ensure stable imagery. However, gimbals only counter rotation in which the centre of rotation is located in space within the camera. In the situation of the rail based robotic platform, the camera is located on an arm at several meters from the centre of rotation. Thus, the camera not only is subjected to rotation, but also to translation, movements the gimbal cannot counter. For a gimbal to be effective, the entire robotic platform would need to act as a gimbal.

The third method, the mechanical approach, is to analyse the mechanical properties and re-design the system to reduce the oscillations.

3 SYSTEM REQUIREMENTS

After introducing the context of this research and analysing HiPerGreen’s current systems, section 3 presents the requirements and wishes for the system.

3.1 Requirements

Table 2 presents hard requirements the system must meet. These requirements are determined by the student together with HiPerGreen researchers.

Table 2 List of Requirements

Reference	Description	Additional Information
Req-01	The structure shall be usable by maximum of one person	The system can be used by only one person throughout a session: carrying it from the transportation vehicle to the greenhouse, ease of installing onto a heat pipe, regardless of the height of pipe, and removing from the heat pipe
Req-02	The system shall have a maximum weight of 10 [kg]	This is to limit the effort of installing the system
Req-03	The system shall be able to scout a lane in less than 10 minutes	The longest lane known to HiPerGreen is 140 [m] long, this means that the system shall go the entire length AND back within 600 [s].
Req-04	The system shall have oscillations around any axis with an amplitude of less than 3 degree	
Req-05	When oscillation occurs, the settling time shall be below 1 [s]	The camera takes approximately 1 image per second.
Req-06	The system shall not interfere with any other object in the vicinity of the heat-pipes	Objects such as fans, lights and other greenhouse installations.
Req-07	The system shall be operable by one person	During operations, the system will not require more than 1 person, e.g. a pilot and a sensory equipment is not permitted.
Req-08	The system shall have as longest dimension in any direction a maximum length of 2.5 [m]	This is to make sure it can fit within a van for transportation.
Req-09	The system shall be able to operate for a continuous time of at least 30 [min] without human intervention	Some of the greenhouses have a harsh climate, with temperature around 30 [°C] and 90 % relative humidity. The system needs to be sufficiently isolated and cooled to be able to continuously operate without failure.
Req-10	The system shall react to user input at any given time with a latency no bigger than 2 [s]	The user needs to be able to react to any unforeseen elements, such as the system moving too fast, or realising something is in the path of the system.
Req-11	Produced images shall have a blur index above 300	The index is calculated using variance of Laplacian transform, outlined in 2.4.
Req-12	Images shall be consistent through a same run, with a maximum deviation of landmarks of 1 % within the image.	Alignment is an important aspect, and therefore consistency is important throughout a run. A landmark such as the edge of a table has to be consistent throughout images. With an image size of 4500 pixels, 1 % is 45 pixels.
Req-13	The communication between the user and the system shall be uninterrupted over a distance of at least 200 [m]	Greenhouses compartment have been recorded to be up to 150 [m] long and 100 [m] wide, making a diagonal of just over 180 [m].

4 CONCEPTUAL SYSTEM CONFIGURATION

In this section mechanical configurations for the platform skeletons are outlined, a total of three configurations are considered. All three will have very similar mechatronics and software, but the configuration will dictate the complexity and determine if requirements are met or not.

Sub section 4.1 presents the criteria, sub sections 4.2, 4.3 and 4.4 presents three concepts to be subjected to a trade-off study to determine the best configuration, and finally sub-section 4.5 presents the result of the trade-off.

4.1 Criteria for Trade-Off

Criteria are formulated to determine an optimal configuration. These criteria are determined based on the requirements outlined in section 3. Not all requirements are considered as some of them do not influence the configuration. Seven criteria are derived and presented in this sub-section with a description of the criterion, its origin, its scale and its weight.

All criteria are scored on a scale of 1 to 5, with a weight ranging from 1 to 3, 1 being the least important and 3 the most. Some criteria have linear scale, to which linear interpolation is used to determine the score followed by rounding the score. The score on each criterion is then multiplied by the weight, and the final score is the sum of all weighted scores. The configuration with the highest final score is chosen for further development.

The criteria are as follows:

1. Stability

Oscillations are undesirable when imaging crop. Thus, criteria is concerning the oscillatory stability around all three axes, namely around the x, y and z axis. This is derived from Req-04, *the system shall have oscillations in any direction with an amplitude of less than 3 degree*. The requirement states a maximum oscillation amplitude, however, if oscillation around an axis can already be determined to be naturally stable, the configuration scores higher.

The scoring is done as follows: if the configuration is stable in all three direction, it scores a 5; in two direction, it scores a 3 and in only one direction, it scores a 1. As stability adds a lot of further development, this criterion has a weight of 3.

2. Estimated weight

The weight is to be limited as stated in Req-02, *The system shall have a maximum weight of 10 [kg]*. This correlates also to the Req-01, for ease of operations. For all conceptual configuration, the motor unit is the same. The unit is taken from R.S. V2 for estimating weight, which weighs 2.3 [kg].

This criterion is scored on a linear scale, ranging from 1 [kg] to 10 [kg], with the scoring of 5 for 1 [kg] and 0 for 10 [kg] and above. Due to the relative importance, this criterion is given the weight of 2.

3. Ease of assembly/disassembly

The ease of assembly is considered based on how long and how many steps is required to assemble the configuration. As the system will be in small form for transportation, it is required to assemble it before operations. This criterion is derived from Req-01, *The structure shall be usable by maximum of one person*.

A configuration requiring no assembly/disassembly scores 5, and a point is deducted for every assembly step. As this presents mostly time consumption during operations, this criterion has a weight of 2.

4. Versatility

The versatility criterion describes how easily can the system be adapted to have different sensors. With a lateral beam, multiple sensors can be attached at various locations, and thus surveying in different ways. This criterion is not derived from the requirements, but rather from the purpose of the system.

The scale for this criterion is based on the number of sensors it can hold at any given time, or in other words, the length of space available to hold sensors. A total length of 4 [m] scores 5 and a length of 0.1 [m] scores a 1. Anything in between is scored linearly.

This criterion is fitted with a weight of 1.

5. Power consumption

The power consumption is derived from Req-10, *The system shall be able to operate for a continuous time of at least 30 [min] without human intervention.* A higher power consumption forces a bigger battery pack to meet Req-10, which in turn makes the system heavier.

Work is described by the following formula:

$$W = F * d$$

Equation 1 Work equation

And

$$F = m * a$$

Equation 2 Force equation

Lastly, to take into account losses due to friction and conversion, each wheel will have a coefficient of 0.01 (typically between 0.008 and 0.02 (Barbir, 2013)) (μ_{wheel}) and electric gates tend to have a 0.95 coefficient of efficiency ($c_{electric\ conversion}$) (Scherz, et al., 2013).

Since each ski consists of two wheels and one motor unit, the total coefficient can be described:

$$k = (1 + \mu_{wheel}) * (1 + \mu_{wheel}) * \frac{1}{c_{electric\ conversion}}$$

Equation 3 Coefficient combined

Combining Equation 1, Equation 2 and Equation 3, we get:

$$W = (m * a) * d * (k * n)$$

Equation 4 Work approximation equation

With n the number of skis.

As the system is moving at 1 [m/s], and it is assumed for a to be 1, the formula to estimate power consumption is:

$$W = 1.074 * m * n$$

The current battery used in R.S.V2 has a capacity of 10 [Wh]. To calculate how long the battery will last, the following equation can be used:

$$t = \frac{Wh}{W} = \frac{10}{1.074 * m * n} [hours]$$

Equation 5 Time estimation

The score is based on a linear scale, ranging from 3 hours as a score of 5 to 30 minutes as a score of 1. As power consumption is dependent on other factors, this criterion is weighted with a factor of 2.

4.2 Configuration Concept 1

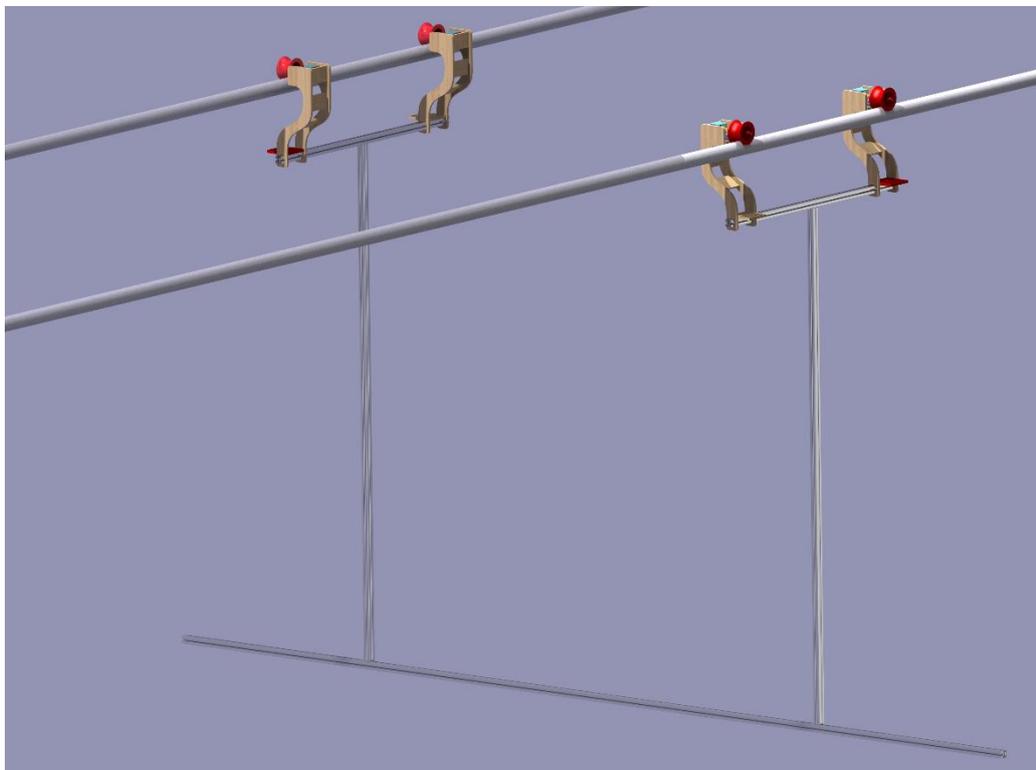


Figure 25 Schematic of Configuration Concept 1

Configuration concept 1 is based on the bridge configuration already found in state-of-the-art rail systems, such as the one made by De Vette (Figure 7). It is also the basis on which R.S. V1 was constructed. It uses two skis to hang the lateral beam and uses a hanging bridge approach to hold itself stable, as can be seen in Figure 25. It is interesting to re-visit the concept in order to understand better how it differs from other conceptualised ideas.

Stability

This concept has four contact points, making it very stable around all axis. With heat pipes being 4.5 [m] apart, and assuming fully rigid body and corners, this gives almost no free oscillation around the y-axis. In the same manner with two wheels on each ski, almost no free oscillation around the x-axis can be found. Lastly, with all four wheels being concave around the heat pipes and assuming no derailing situation, there is no free oscillation around the z-axis.

This conceptual configuration receives therefore a score of 5 for the criterion stability.

Estimated weight

The concept has 8 [m] of beams, 4 [m] as the lateral beam and two 2 [m] vertical beams. With a weight of 0.9 [kg] per meter, the total beam weight is to be of 7.2 [kg]. This concept needs both skis to be motorised. With a motor unit weight of 2.3 [kg] each, this concept has a total estimated weight of 11.8 [kg].

This conceptual configuration receives the score of 0 as the weight is above the 10 [kg] for the criterion estimated weight.

Ease of assembly

This concept consists of two skis and a lateral beam. Each ski needs one assembly step on their own, followed by the attachment between the lateral beam and each ski. This brings to a total of 4 assembly steps.

This conceptual configuration receives therefore a score of 1 for the criterion ease of assembly.

Versatility

The total length available on this concept is 4 [m], allowing multiple sensor packages.

This conceptual configuration receives therefore a score of 5 for the criterion versatility.

Power Consumption

Following the formula outlined in Equation 5, the estimated time for this concept is 0.39 [hrs]

$$t = \frac{Wh}{W} = \frac{10}{1.074 * 11.8 * 2} = 0.39 [hours]$$

This conceptual configuration receives therefore a score of 1 for the criterion power consumption.

4.3 Configuration Concept 2

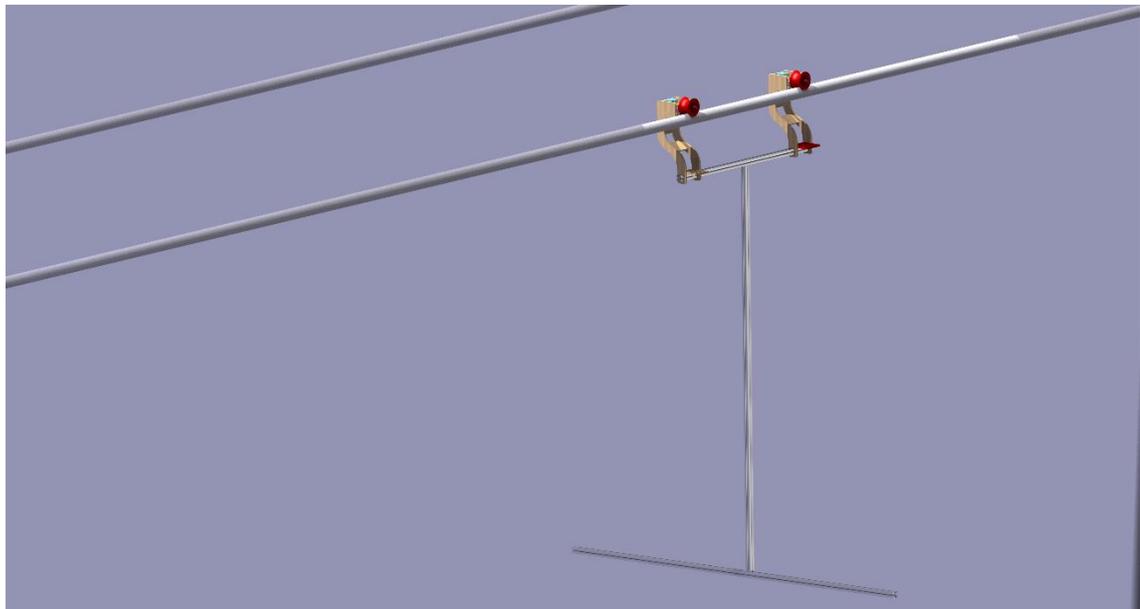


Figure 26 Schematic of Configuration Concept 2

Configuration concept 2 is based on the mono rail system already found in state-of-the-art rail systems, such as shown in Figure 5. It also was the basis on which R.S. V2 was constructed. It consists of one simple ski, with a lateral beam to hold cameras and other sensors. The lateral

beam offers relative flexibility with respect to sensor placement, beneficial for yet known greenhouses. A CAD based model can be seen in Figure 26.

Stability

This concept has two contact points based on the only ski used. This provides stability around the x-axis. However, with a relatively long vertical beam, in the same manner as R.S. V2, oscillation can occur around the z-axis. With two aligned contact point on a single rounded beam (heat pipe), oscillation will occur around the y-axis.

This conceptual configuration receives therefore a score of 1 for the criterion stability.

Estimated weight

The concept has 4 [m] of beams, with one vertical beam of 2 [m] and a horizontal beam of 2 [m]. With a weight of 0.9 [kg] per meter, the total beam weight is to be of 3.6 [kg]. This concept has only one ski. With an estimated motor unit weight of 2.3 [kg], this concept has a total estimated weight of 5.9 [kg].

This conceptual configuration receives the score of 3 for the criterion estimated weight.

Ease of assembly

This concept consists of one ski and a lateral beam. This means this concept needs only one step: connecting the lateral beam with the ski.

This conceptual configuration receives therefore a score of 4 for the criterion ease of assembly.

Versatility

The total length available on this concept is 2 [m].

This conceptual configuration receives therefore a score of 3 for the criterion versatility.

Power Consumption

Following the formula outlined in Equation 5, the estimated time for this concept is 1.57 [hrs]

$$t = \frac{Wh}{W} = \frac{10}{1.074 * 5.9 * 1} = 1.57 [hours]$$

This conceptual configuration receives therefore a score of 3 for the criterion power consumption.

4.4 Configuration Concept 3

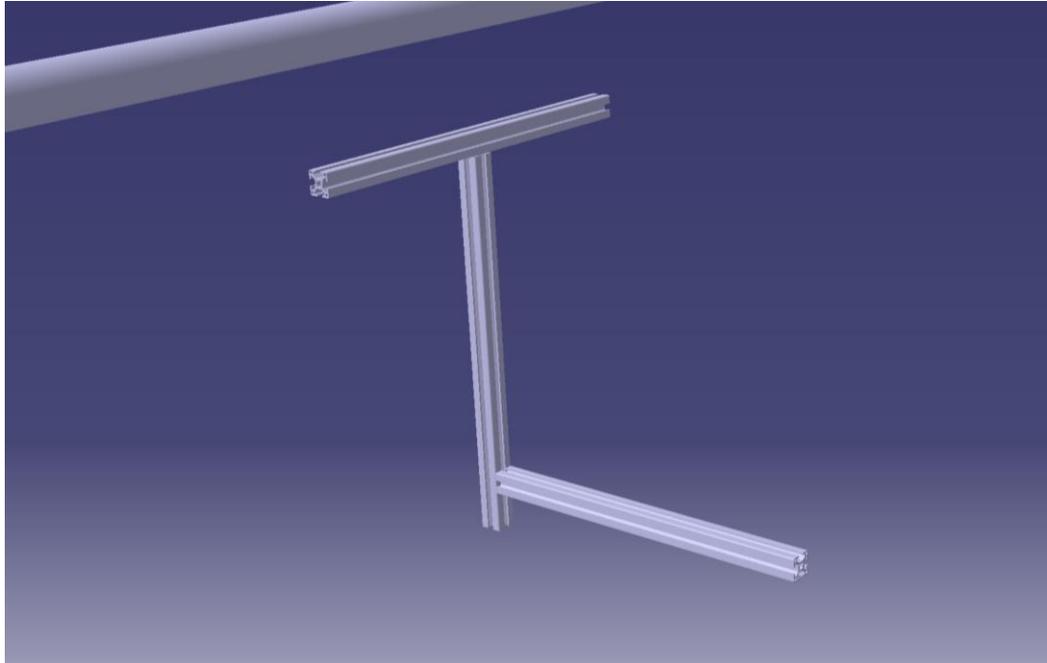


Figure 27 Schematic of Configuration Concept 3

Configuration concept 3 is a similar version of concept 2. Also based on the mono rail system found in section 2.2 “*State-of-the-Art Rail Systems*”, it consists of one ski with a lateral beam. However, after discussions with HiPerGreen and considering the usage of high resolution cameras, a short vertical beam is chosen for this concept as well as a short lateral beam. This reduces drastically the weight and oscillations. On the otherhand, it limits by far the versatility in placement for sensors. A CAD based schematic can be found in Figure 27.

Stability

This concept has two contact points based on the only ski used. This provides stability around the x-axis. With a relatively short vertical beam, it is estimated that the oscillation around the z-axis is minimal. With two aligned contact point on a single rounded beam (heat pipe), oscillation will occur around the y-axis.

This conceptual configuration receives therefore a score of 3 for the criterion stability.

Estimated weight

The concept has 1.45 [m] of beams, with two bars of 0.5 [m] and a horizontal bar of 0.45 [m]. With a weight of 0.9 [kg] per meter, the total beam weight is to be of 1.305 [kg]. This concept has only one ski. With an estimated motor unit weight of 2.3 [kg], this concept has a total estimated weight of 3.6 [kg].

This conceptual configuration receives the score of 4 for the criterion estimated weight.

Ease of assembly

This concept requires no assembly steps.

This conceptual configuration receives therefore a score of 5 for the criterion ease of assembly.

Versatility

The total length available on this concept is 0.45 [m].

This conceptual configuration receives therefore a score of 2 for the criterion versatility.

Power Consumption

Following the formula outlined in Equation 5, the estimated time for this concept is 2.58 [hrs]

$$t = \frac{Wh}{W} = \frac{10}{1.074 * 3.6 * 1} = 2.58 [hours]$$

This conceptual configuration receives therefore a score of 4 for the criterion power consumption.

4.5 Trade-Off

The scores of each configuration concepts are grouped and compared in Table 3.

Table 3 Trade off table

Criterion	Criterion Weight	Concept 1		Concept 2		Concept 3		Max score
		Score	Score * Weight	Score	Score * Weight	Score	Score * Weight	
Stability	3	5	15	1	3	3	9	15
Estimated Weight	2	0	0	3	6	4	8	10
Ease of Assembly	2	1	2	4	8	5	10	10
Versatility	1	5	5	3	3	1	1	5
Power Consumption	2	1	2	3	6	4	8	10
		Total Score Concept 1	24	Total Score Concept 2	26	Total Score Concept 3	36	50

It is clear that configuration concept 3 is the winner.

5 DETAILED MECHATRONICS DESIGN

This section concerns the detailed mechatronic design of the selected rail system configuration. The mechatronic design phase focuses on the engineering of both electrical and mechanical systems, and also includes a combination of robotics, electronics, systems and control.

5.1 Mechanical Design

The mechanical design starts by exploring the oscillatory modes, providing a solution to counter them, and finally presents a final mechanical skeleton.

5.1.1 Oscillatory Motion

Oscillatory motion in a system can be modelled and predicated using vibration theory. All bodies possessing mass and elasticity are capable of vibration. There are broadly speaking two families of oscillatory motion; *linear* and *non-linear*. Linear systems can be modelled using relatively simple mathematics and the principle of superposition holds. Techniques for analysing non-linear systems are less well known and difficult to apply. It's important to note that all systems tend to become nonlinear with increasing amplitude of oscillation.

There are two classes of vibrations; *free* and *forced*. Free vibrations take place due to the forces inherent in the system, a system under free vibration will vibrate at one or more of its natural frequencies. Natural frequencies are properties of the dynamical system depending on mass and stiffness. Vibrations taking place under forced vibration are excited by an external force. If the system is excited by an oscillatory excitation, then the system will be forced to vibrate at the excitation frequency. If the *excitation frequency* coincides with one of the *natural frequencies* of the system, then *resonance* will occur. Resonance can cause structural failure due to dangerously large oscillation.

In order to limit the amplitude of oscillations it is necessary to increase *damping* throughout the system, this should be done for all the systems *degrees of freedom (DOF)*. The DOFs are the number of independent coordinates required to describe the motion of a system. (Thompson, 2014). An illustration of simple harmonic motion can be seen in Figure 28.

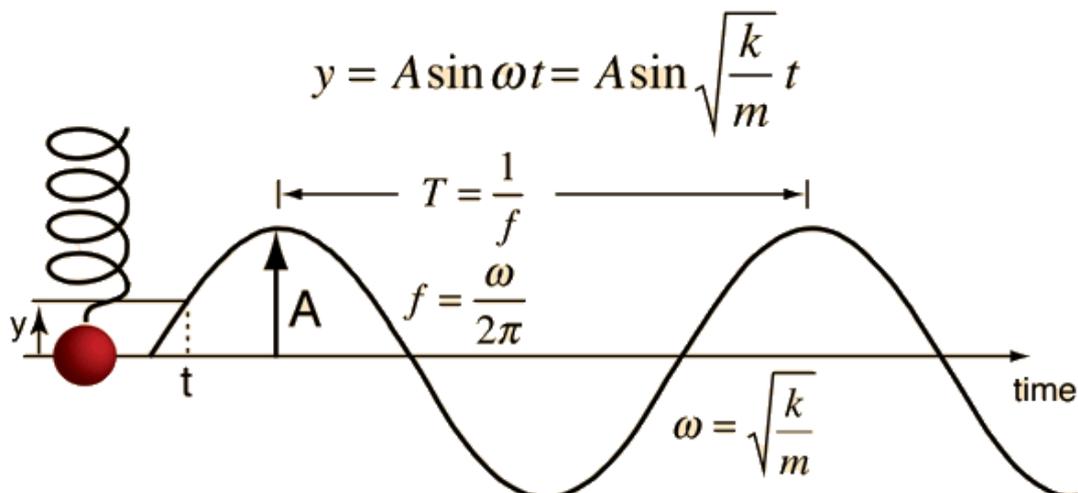


Figure 28 – Simple Harmonic Motion (Nave, 2019)

ω = angular frequency, k = spring stiffness, m = mass, A = amplitude, f = frequency

In this chapter the rail system is analysed to identify the degrees of freedom (DOF) of interest and for each conceptual design the modes of vibration will be determined. The mechanical system design must have sufficient damping to reduce oscillatory motion about these DOFs. We will consider the system to behave linearly and will analyse two conditions, the free vibration of the system of the system when for example a displacement is introduced and the forced vibrations which will occur. These forced vibrations will be excitations due to the motor driving the wheel (uneven axle which introduces oscillation), the start and stop motion of the system and deformations in the heating pipe which introduce an excitation force.

In the R.S. V2, there are two modes of oscillations: around the y-axis and around the z-axis. The first mode will be further researched in section 5.1.2. The second mode, around the z-axis, can be seen from the tests, outlined in APPENDIX A, is due to two main parts: the vertical beam and the wheel mount. However, from the two, the vertical beam is allowing for most of the oscillation. The wheel mount is explained with more details in section 5.1.3.

To understand the behaviour of the vertical beam, it is necessary to investigate the model for torsion, based on R.S. V2. The governing equations for torsion are as follows:

$$\theta = \frac{\tau * l}{G * J}$$

6 Eq. Torsion equation

$$G = \frac{E}{2(1 + \nu)}$$

7 Eq. Modulus of Rigidity relation with Young's Modulus

With:

$\theta = \text{Angle [Rad]}$

$\tau = \text{Torque } \left[\frac{N}{m} \right]$

$l = \text{Length [m]}$

$G = \text{modulus of regidity } \left[\frac{N}{mm^2} \right]$

$J = \text{Torsional constant [m}^4\text{]}$

$E = \text{Young's modulus } \left[\frac{N}{m^2} \right]$

$\nu = \text{Poisson's ratio}$

Combining equations 6 and 7, the equation is:

$$\theta = 2 * (1 + \nu) * \frac{\tau * L}{E * J}$$

8 Eq. Torsion equation (2)

The choice for material is limited. The Bosch profiles used are one of the best options as it offers much versatility for attachment and standardisation. The material and profile of the beam is thus fixed.

The material of the beam is an aluminium alloy, denoted EN-AW 6060 - T66 (Rexroth, Bosch Group, 2016). The beam profile is dubbed 30x30, with the following material and profile properties:

$$J = 0.29[cm^4] = 2.9 * 10^{-9} [m^4]$$

$$E = 70\,000 \left[\frac{N}{mm^2} \right] = 7 * 10^{10} \left[\frac{N}{m^2} \right]$$

$$\nu = 0.34$$

Rearranging equation 8 with the known values, the equation becomes:

$$\theta = \frac{67}{5075} * \tau * L$$

9 Eq. Torsion equation for the vertical beam

From equation 9, it is clear to see that the angle is directly proportional to the torque induced and the length of the beam.

As the torque induced is depended on the load case, the last parameter for change is the length of the beam. To comply with Req. 04, the amplitude of oscillations must be under 3 [°]. From testing of R.S. V2 with a vertical beam length of 2 [m], the amplitude is max of 12.6 [°]. To come within the requirement, the new vertical beam must be:

$$new\ beam \leq length_{old\ beam} * \frac{3}{12.6} = \frac{6}{12.6} = 0.47 [m]$$

Given the space available outlined in section 2, the beam is selected to have a length of 0.45 [m]

However, with a smaller length, the frequency increases. The frequency of a weight on a beam is dictated by the following formula:

$$freq = \frac{1}{2\pi} \sqrt{\frac{GK}{(I_{end\ mass} + I_{beam}) * L}}$$

With:

$$G = modulus\ of\ regidity \left[\frac{N}{m^2} \right]$$

$$K = Polar\ second\ moment\ of\ inertia [m^4]$$

$$I_{end\ mass} = mass\ moment\ of\ inertia\ of\ end\ mass [kg * m^2]$$

$$I_{beam} = mass\ moment\ of\ inertia\ of\ beam [kg * m^2]$$

$$L = Length [m]$$

Looking at the formula, two factors play a role in increasing the frequency, the length of the vertical beam and the reduction of the mass moment of inertia of the 'end mass'. The second factor is because of the camera module placed closer to the vertical beam. The mass moment of inertia of the 'end mass' is dictated by the following formula:

$$I = mr^2$$

The new length is 0.45 [m], or 0.225 * original length. The camera module is placed at 0.3 [m] from the vertical beam, or it is 0.15 * original r . Assuming the mass moment of inertia of the beam is negligible, the new frequency can be re-written as:

$$freq_{R.S. V3} = \frac{1}{2\pi} \sqrt{\frac{GK}{I_{end\ mass} * 0.15^2 * L * 0.225}}$$

$$freq_{R.S. V3} = freq_{R.S. V2} * \sqrt{\frac{1}{0.15^2 * 0.225}} = 0.819 * \sqrt{\frac{1}{0.15^2 * 0.225}} \approx 11.51 [Hz]$$

Looking back onto equation 9, with the new length and same load case of torque 8.3287 [Nm], this yields to a maximum amplitude angle of 0.04947 [rad], or 2.83[°].

Unfortunately, there is not sufficient data on the system to accurately estimate the change in the damping coefficient. However, having talks with experts within HiPerGreen, it is estimated that the damping is to become sharper, or in other words, a damping ratio closer to 1 is expected compared to the previous system (see APPENDIX A).

5.1.2 Gyroscopic Precession

The re-design explained in the previous section are passive measures for the oscillation. However, a more active solution can be considered. The oscillations around the y-axis are expected to be much less because of the new smaller beam, to the point that an active counter measure is not needed. That said, research is done in case of the need for active counter measure. A high candidate is the use of gyroscopic precession.

Gyroscopic precession is a phenomenon that happens when a relatively big enough mass rotates relatively fast where any torque induced perpendicular to the spin axis, that torque takes effects in a 90 [°] phase (Nagaraja, 2019). An explanatory schematic can be found in Figure 29.

Gyroscopic precession is governed by the following equation:

$$\omega_p = \frac{\tau}{I_s * \omega_s}$$

Where:

ω_p = Angular velocity of precession [Rad/s]

τ = Torque $\left[\frac{N}{m}\right]$

I_s = Moment of Inertia of spin [kg m²]

ω_s = Angular velocity of spin [Rad/s]

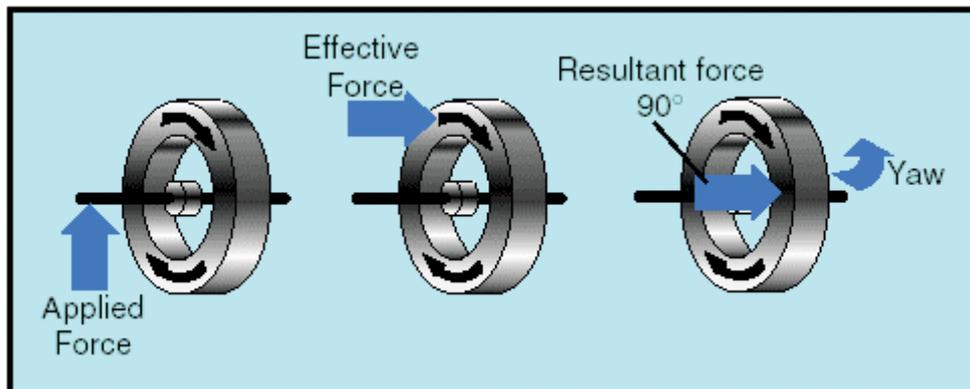


Figure 29 Schematics of Gyroscopic Precession (Maaz, 2016)

Through the gyroscopic precession, moments around the y-axis will act in a 90 [°] phase, and thus act as a torque around the z-axis. With two concave contact wheels and relatively low mass, the system is expected to stay in place.

5.1.3 Re-design of the Wheels

The top of the skis, where the wheels are mounted in R.S. V2 are susceptible to oscillations. Made out of multiplex wood, the material and construction allow for much deflection, although not

as much as the vertical beam. Comparing Figure 48 and Figure 49 in Appendix A.4, there is a clear factor of ten difference between the allowed torsion from the wheel mounts and the vertical beam. That said, a re-design of the wheel mount is necessary to comply with Req-04.

The re-design of the wheel mount has been done by a colleague within HiPerGreen, Westerhout J., mechanical engineer. The approach taken is to reduce the number of parts within the mount with a as direct connection as possible to the vertical beam. The wheel shafts are directly bolted to an aluminium piece, called the shaft holder, which in turn is locked to the rest of the system. Pictures can be found in Figure 30.

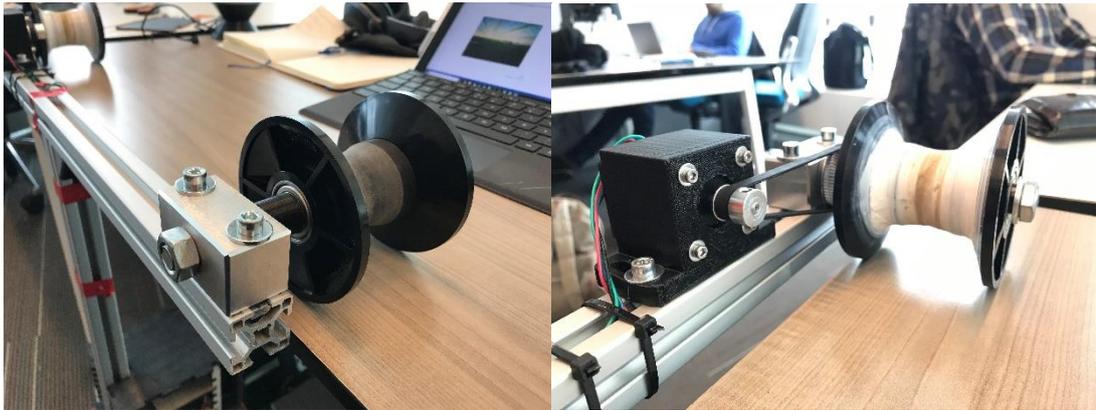


Figure 30 New wheel mount

The shaft does not rotate, rather bearings are placed within the wheel that let the wheel rotate around the shaft. The shaft is bolted within the shaft holder. Contrary from the R.S. V2 where the wheels were 3D printed, in the new design the wheels are made of PVC Nylon and bought from a manufacturer. This ensures no deformation due to heat. Lastly, the motor is not directly connected to the shaft, but a belt is in place to drive the wheel. This reduces the length needed, and this making the system more compact.

5.2 Electronic Design

The electronics must be as simple as possible. The design consists of keeping the bare minimum but as functional as possible.

5.2.1 Computing Unit

Only one computing unit is used for the design to keep the complexity to a minimum. The computing unit is the Arduino Mega board. As P. Scherz describes it: *“Arduino is an open source hardware platform for microcontrollers prototyping. It encompasses both a microcontroller development board and an integrated development environment (IDE). The IDE is simple to use and available for Mac, Linux, and Windows computers. Arduino boards are extremely popular as a starting point for using microcontrollers technologies. Their popularity is due to a number of factors, including the following:*

- *Low cost (around \$30)*
- *Open source design*
- *Easy-to-use and cross-platform IDE*
- *Availability of plug-in shields (expansion hardware)” (Scherz, et al., 2013)*

Working on 5[V], the Arduino is compatible with the receiver, the voltage regulator and the stepper motor driver.

5.2.2 Communication

Pulse Width Modulation (PWM) is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. This signal type is very commonly used to simulate voltage differences in LEDs. By reducing the duty cycle, the average “on” time appears to be lower, and thus a human would perceive the light to be dimmer. An example of a PWM signal is given in Figure 31.

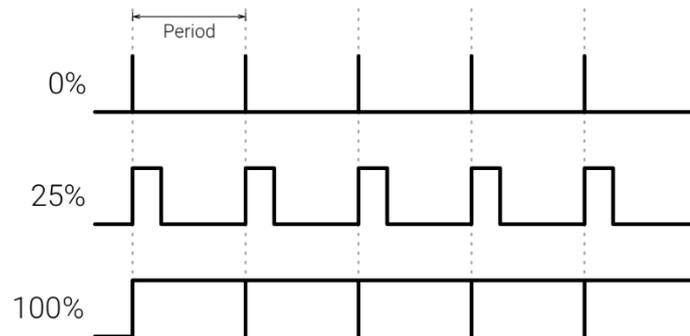


Figure 31 Schematics of a PWM signal (Android Developers, 2016)

PWM signal is commonly found in RC planes as a communication signal, i.e. from remote control to device, where positions of command sticks are embedded in a minimum of 4 channels, although most controllers offer more than 4 channels. Commonly found is 6 channel remotes, with 2 auxiliary channels, but more complex remote can have up to 12 channels. This is transmitted over the 2.4 [GHz] band where each channel represents either pitch, roll, yaw or throttle. The frequency of the PWM is standardised at 50 [Hz], with command pulses ranging from 1000 [μ s] to 2000 [μ s], with 1500 [μ s] as the midpoint.

The official specification of the module acquired specifies a range of 2.7 [km] with direct line of sight. With objects (such as buildings, animals, plants etc..) in between, the range decreases. No data can be found on relation on how much object reduces the range. However, in greenhouses the space is mostly un-occupied, with the biggest obstructing object being metallic poles. Signal cannot go through metal, but it can ‘bounce’ of it. The radio remote is equipped with omnidirectional antennas, therefore it is expected that the range of the communication is able to meet the requirement of at least 150 [m]

5.2.3 Controlled Movement

Stepper motor is a brushless DC motor that rotates in steps. This is very useful because it can be precisely positioned without any feedback sensor, which represents an open-loop controller. The stepper motor consists of a rotor that is generally a permanent magnet and it is surrounded by the windings of the stator. As the windings are activated step by step in a particular order and a current flows through them, they will magnetize the stator and make electromagnetic poles respectively that will cause propulsion to the motor. (Scherz, et al., 2013)

There are several different ways of driving the stepper motor. The first one is the Wave Drive or Single-Coil Excitation. In this mode just one coil is active at a time which means that for this example of motor with 4 coils, the rotor will make a full cycle in 4 steps.

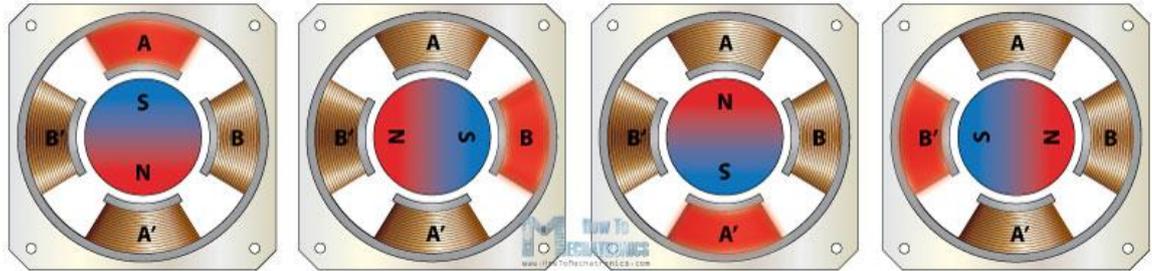


Figure 32 Sequence of stepper motor (Dejan, 2019)

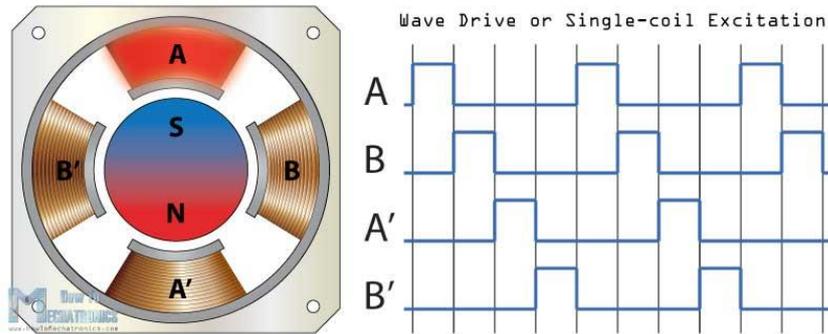


Figure 33 Stepper motor sequence of coils in Single-coil Excitation (Dejan, 2019)

Next is the full step drive mode which provides much higher torque output because there always is 2 active coils at a given time. However, this doesn't improve the resolution of the stepper and again the rotor will make a full cycle in 4 steps.

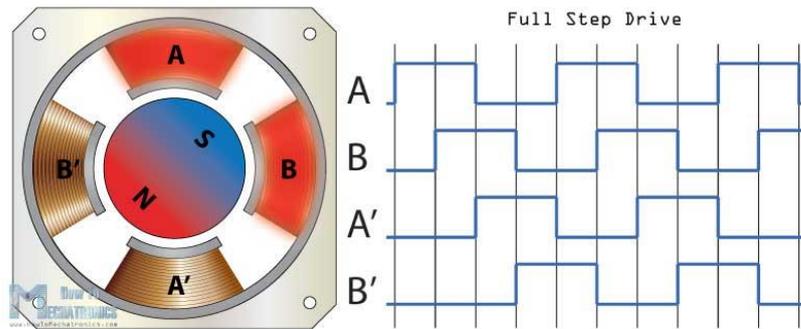


Figure 34 Stepper motor sequence of coils in full step drive (Dejan, 2019)

For increasing the resolution of the stepper motor the Half Step Drive mode is used. This mode is actually a combination of the previous two modes. Here a coil is active followed by 2 active coils and then again one active coil followed by 2 active coils and so on. With this mode double the resolution can be obtained with the same construction. Now the rotor will make a full cycle in 8 steps.

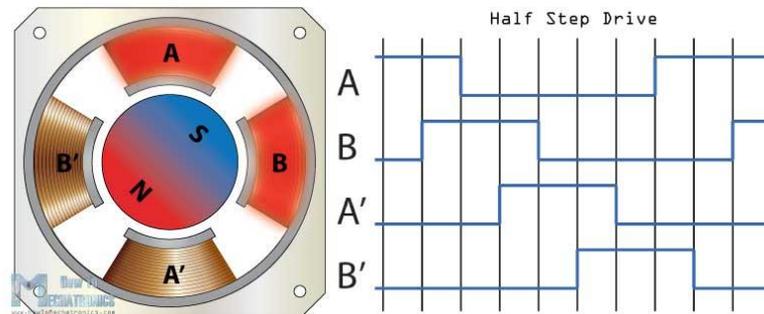


Figure 35 Stepper motor sequence of coils in half step drive (Dejan, 2019)

However, the most common method of controlling stepper motors nowadays is the Microstepping. In this mode a variable controlled current is provided to the coils in form of sin wave. This will provide smooth motion of the rotor, decrease the stress of the parts and increase the accuracy of the stepper motor.

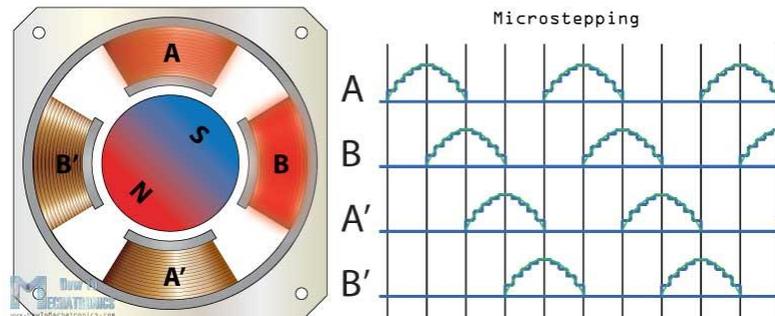


Figure 36 Stepper motor sequence in coils in microstepping (Dejan, 2019)

Many stepper motors are available on the market, but stepper motor Trinamic QMot 1.8° is selected, as it is easily accessible through the accredited supplier for HiPerGreen and comes in a small form factor. Moreover, it is easily accessible through HiPerGreen. This motor operates at 200 steps per revolution.

The stepper motor cannot be controlled directly by the microcontroller as they do not operate at the same voltage: 5 [V] vs 16 [V]. A motor driver is necessary. Building a driver is possible, but stepper motor drivers are widely available. However, they all work on the basis of gates, MOFETS and diodes. An example of a simple stepper motor drive can be found in Figure 37.

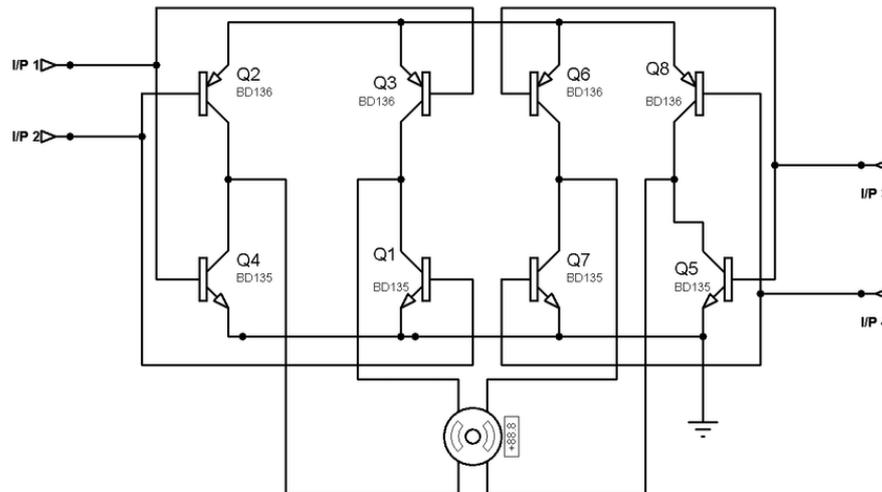


Figure 37 Diagram of Stepper Motor Drive (electronics-course.com, 2019)

The motor driver selected is the Pololu A4988. It can easily be used by many microcontrollers, Arduino Mega included, and accepts a wide range of voltages and currents, within the limitations of the battery and motor selected. It also offers multiple levels of microstepping internally.

To mitigate the overheating issue revealed during testing of R.S. V2, a new heat sink is fitted. In this situation, the dominant heat transfer mode is conduction. This mode is governed by the following equation:

$$\dot{Q} = -kA\left(\frac{dT}{dt}\right)$$

Equation 10 Heat transfer through conduction (Scherz, et al., 2013)

It is clear that by increasing the area, more heat can be dissipated. However, due to the limiting area available on the motor driver, a heatsink only twice the size of the original one is fitted. Furthermore, another solution is incorporated: airflow. To hold the optimum heat transfer rates, dT needs to be as large as possible. As heat is transferred from body 1 to body 2, dT decreases and thus heat transfer slows down. By introducing airflow, dT is kept as large as the airflow can keep up with the heat transfer.

5.2.4 Overall Circuit

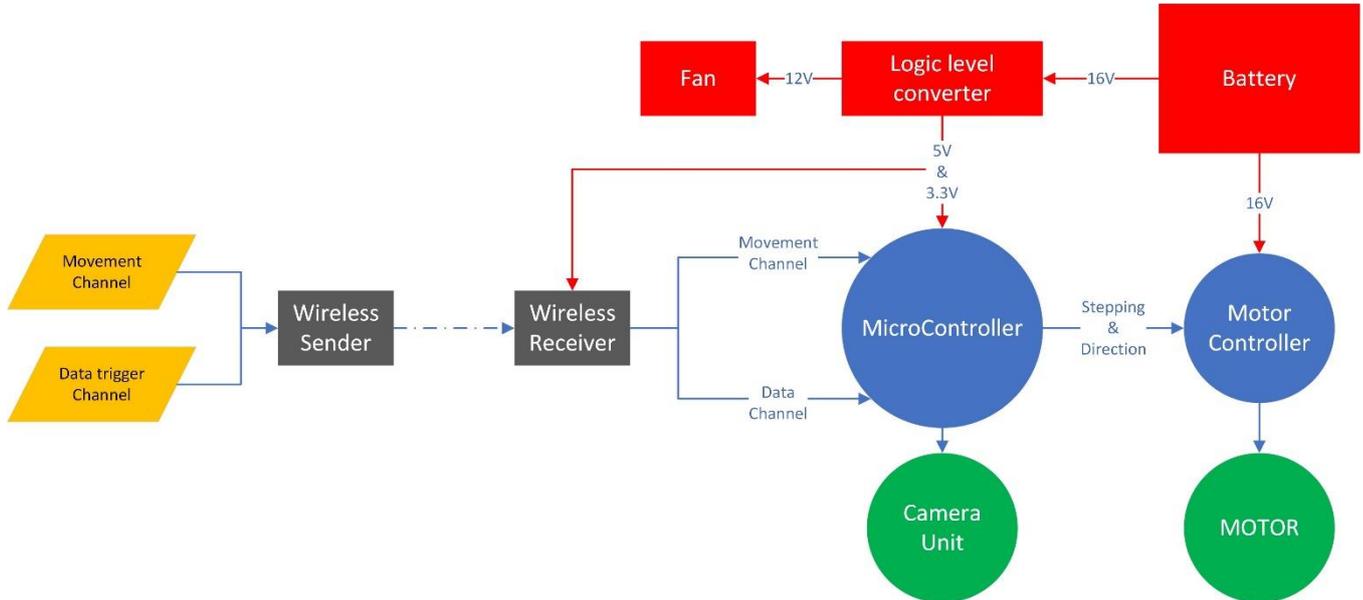


Figure 38 Electronic schematic proposition

The overall circuit diagram can be found in Figure 38. All of the electronics is placed in a multi-levelled box. The box offers protection from the occasional droplets from the greenhouse roof, keeps the electronics in place, and with the fan strategically placed on one side and wholes placed on the other side the airflow is continuous and directed through the electronics for cooling. A photo can be found in Figure 39.

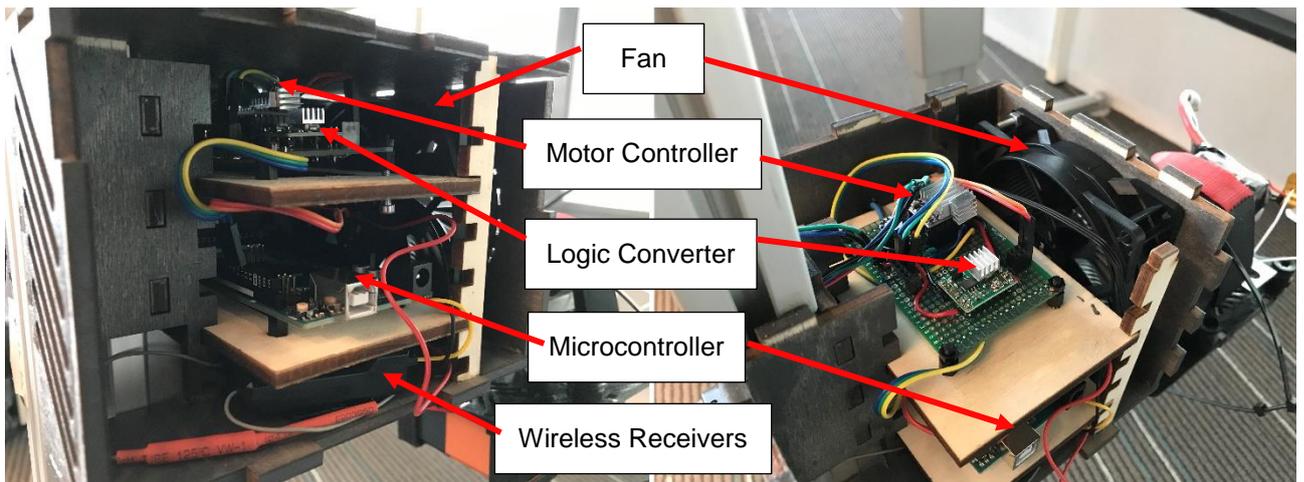


Figure 39 photo of electronics box

5.3 Software and Control Design

The design of the software is highly dependent on the control approach. The control approach is first described before designing the software.

5.3.1 Control Approach

As the environment is continuously changing (growers adding materials, system used in various unknown greenhouses) and a variety of test are expected through HiPerGreen, a user of the system is always present during operations. The control therefore needs to be direct user input. A methodology as per R.S. V2 is not suitable, as the user needs to manually type in the command, which takes relatively long. The approach here is to have a stick input dictates the speed of the system.

This can be achieved by a short loop in the software where at each iteration the speed can be adjusted. In this manner, a near real-time response is achieved by the system.

One can distinguish four different kinds of timeliness for systems (Boode, 2018): hard, firm, soft and near real time. Hard real time means that whenever a deadline is missed, the consequences are to be catastrophic. Firm real time allows a certain number of deadline-misses before the situation becomes catastrophic. Soft real time means that after a deadline is missed, there is still some revenue while dropping to zero (no revenue but no catastrophe as well). Near real time refers to delays for example in network management systems, where no specific deadlines are given but the revenue degrades to zero over time.

In this part, in the description of the control of the rail system, we deal with near real time because we do not have specific deadlines while controlling the rail system. A user's input leads to some action to be performed by the rail system but the amount of time it takes for the rail system to reach the required state is time independent but must be eventually successful.

5.3.2 Software Architecture

To keep the loop runtime as minimal as possible, it is important to keep the steps as simple as possible. The loop consists of three main steps: read the radio signal inputs, convert signal input to a speed equivalent and finally move one step at the found speed. A flow diagram can be found in Figure 40.

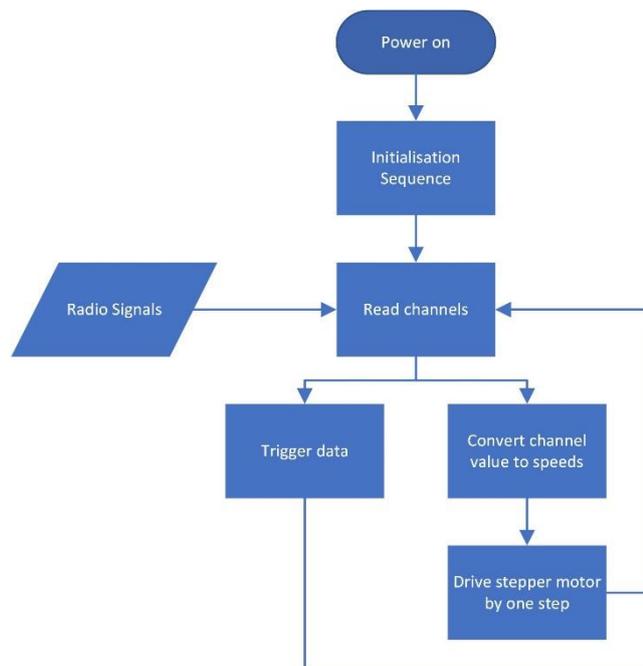


Figure 40 Software flow diagram for R.S. V3

The Arduino environment offers a function to read pulses, called “*pulseIn()*”, which returns the time of positive pulses [μs]. With this function, a read-out of the command by the user can be used to control the motor directly or trigger the sensor. The channels of throttle and auxiliary 01 is used as throttle for the stepper motor and trigger for the data. The source code is as such:

```
1. long pulseLength = pulseIn(Throttle_Input_Pin,HIGH);
```

With:

Long: data type
pulseLength: the length of the pulse in μs
Throttle_Input_Pin: the pin the controller needs to read it from
HIGH: the type of pulse to look for, between high voltage or low voltage

The stepper motor driver works on one single line changing from logic high to logic low to move by one step. The time in between steps determines the speed at which the motor rotates. The aim is to step fast enough to create a continuous motion and correlate linearly to the throttle input of the user. Arduino environment offers a function to do that called “*map()*” as follows:

```
1. newValue = map(value, fromLow, fromHigh, toLow, toHigh)
```

With:

value: input value
fromLow: the lower bound of the value's current range
fromHigh: the upper bound of the value's current range
toLow: the lower bound of the value's target range
toHigh: the upper bound of the value's target range
newValue: the mapped value

For smooth motion, the stepper motor driver is set to the mode where each step calling, the motor is doing 1/8 of a step. However, due to its internals, that means that there is a maximum speed at which the driver is able to keep up with the input stepping speed. Through testing, it was found to be an in-between step delay of 8 [μs]. Through same testing, the minimum speed the user can drive the motor is set to be an in-between step delay of 300 [μs].

To give possibility to the user to be able to go forward and backwards, the throttle is set to be forward when the throttle pulses are between 1600 and 2000 [μs], with 1600 as low speed and 2000 as high speed, and backwards when the throttle pulses are between 1000 [μs] and 1400 [μs], 1000 as high speed and 1400 as low speed. Between 1400 and 1600 [μs], the motor is stationary. This area of inactivity is called a deadband. As the command sticks do not come back to the midpoint on their own, the deadband ensures that no unwanted movement occurs due to inaccuracy of the sticks. The stepping part of the code is as follows:

```
1. digitalWrite(StepPin, HIGH);  
2. delayMicroseconds(long(SteppingInput/2));  
3. digitalWrite(StepPin, LOW);  
4. delayMicroseconds(long(SteppingInput/2));
```

Lastly, a debug function is added, to print out the values read from the radio signal and the outputting stepping times.

During testing of this software setup, the motor was running in a jittering manner, suggesting the time the loop took to iterate was too long. The microcontroller is not a powerful computing unit and requires efficient scripts to improve performance.

After using debugging tools, the cycle of the program has been measured to be between 0.4 [s] and 0.2 [s], far too long to drive the stepper motor.

5.3.3 Run Time Improvement

The microcontroller is an 8-bit based CPU (Central Processing Unit), meaning that it can only make operations on 8 bits at every clock cycle. A byte consists of 8 bits and has a limited range from 0 to 255. To use higher number, one can use multiple bytes together to be able to manipulate higher numbers. With two bytes, the range is 0 to 65 535. However, the CPU needs to compute it in several cycles, increasing CPU time.

It is thus necessary to be efficient in data type declaration. Any variables with expected values to be below 255, positive and whole are declared as “byte” type, variables with expected values below 65 535, positive and whole are declared as “unsigned int” type and so on. A table summarising data types can be found in APPENDIX B.

From some research on the Arduino community forums, it appears that the “*map()*” function is a slow process. A conversion of data type is made, then mathematic mapping is done, followed by another type conversion. This is done to ensure compatibility with any data type however, it comes at the expense of CPU time. Moreover, according to many forum debates, the function is prone for overflow.

The “*map()*” function is thus replaced by two less flexible but fixed linear interpolation functions for driving forward and driving backwards:

```
1. SteppingInput = (MaxDelay*(2000 - Throttle_Input_Value) + MinDelay*(Throttle_Input
_Value - 1600))/(1600); //driving forward
2. SteppingInput = (MinDelay*(1400 - Throttle_Input_Value) + MaxDelay*(Throttle_Input
_Value - 1000))/1600; //driving backwards
```

With no data type changes, this performs much faster.

Reading the radio signal is a slow process. The function “*pulseIn()*” makes multiple readings before returning the value. Furthermore, for each reading, the function will wait until the signal is in the appropriate logic level before starting the timer. In practise, say you want to read the logic high duration, but the microcontroller happens to start enquiring while the signal is already logic high, the CPU waits until the signal goes to logic low, then awaits the change to logic high to start a timer. PWM frequency is at 50Hz, meaning that the CPU can be in idling time for as much as 20 [ms]. The stepper motor is set to be running with a delay in-between steps of as low as 8 [µs]. It is suspected that this is causing most of the delays within the original software.

Interrupts on microcontrollers are a good way to mitigate the issue. Interrupts work on basis of two streams of programs, the main program and the ISR (Interrupt Service Routine). Upon a pre-define change, the main program gets paused to run the ISR. However, it is important to keep the ISR as short as possible, as it can make the main program unstable if dependent on time accurate operations. As Arduino says: “*Generally, an ISR should be as short and fast as possible.*” (Arduino Corp., 2019). In the design, the ISR is set to occur only when logic level changes. If a positive change is registered, a timer is started, and on a negative change, the timer is stopped, and the timing is saved. Figure 41 shows a comparative schematic of the doing.

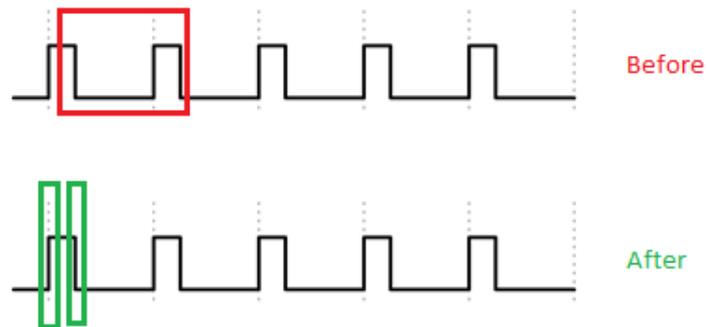


Figure 41 PWM Signal processing time comparison schematic

Lastly, the debug function is made toggleable. Printing characters is time consuming. With a transfer rate of 115200 [bps], and printing messages of 42 characters, with 8 bits per character, the printing time is:

$$printing\ time = \frac{1}{\frac{115200}{42 * 8}} \approx 0.003\ [s]$$

This is again a considerable amount of time.

Implementing all the improvements, a smooth operation is attained. The cycle time of the program is measured to be between 4 and 6 [μ s].

The latency on command input is forecasted to be at maximum of the length of one pulse cycle, 1/50 [s].

The full source code can be found in APPENDIX D for further review.

6 TESTING & VALIDATION

This section presents the testing of the design. After the design completed in the previous section, the product is put to the test to confirm and validate the requirements set out. The outline of the test is presented through a description of the setup, the test and the results. Finally, a discussion is held on validating the results.

6.1 Testing of V3

The testing of V3 is setup to assess three key components: oscillations, imagery and latency over a long distance. The oscillation aspect is tested in two scenarios, the first with induced oscillation in the same manner as done in V2 and the second by driving over the welded joints of the heating pipes. In both cases, data is recorded and images are taken for assessment. The distance aspect is tested over a long distance by timing differences in input command and reaction of the system.

6.1.1 Setup

Both stages of the test are conducted in the greenhouse, on the heating pipes. The heating pipes are hanging 4 [m] above the crop. R.S. V3 is fitted with a sensing module inside the electronics box, right beside the vertical beam of R.S. V3 to register IMU information. A camera is linked to the sensing module to trigger photos as the system is in movement.

The sensing module consists of two elements: a microcontroller (Arduino Uno R3) and an Inertial Measurement Unit (IMU) (TDK-InvenSense MPU9250). The microcontroller reads data from the IMU at roughly 200 [Hz] and transfers it to a computer, on which a program reads the data and saves it in an orderly fashion. A snippet of the microcontroller source code, the source code for saving the data and a snippet of the recorded data can be found in sections A.5 through A.7.

The IMU consists of three sensors: accelerometer, gyroscope and compass. The accelerometer reads the linear acceleration in three axes, the compass reads the earth magnetic field strength and direction to deduce the heading of the IMU, and the gyroscope reads rotational accelerations around all three axes.

The camera is linked to a micro-computer (Raspberry-Pi), which saves the images on an interval. The sensing module is plugged to the micro-computer. The micro-computer is programmed to register the input of the sensing module as well as saving images. This will ensure perfect synchronisation between the reading time stamps and the image time stamps. However, as the camera takes time to take an image, process the incoming data and save it, images are not always at the highest amplitude of oscillation.

It is worth mentioning that generally, except for high performance IMUs, the accelerometer and the compass tend to be very noisy but stable, and the gyroscope tend to be very accurate but will drift with time (Inertial Navigation – Forty Years of Evolution, 1998).

6.1.2 Test

Once the system is stable on the pipe, readings are started and recorded. The system is then excited around the z-axis by manual means in such a way that no other oscillations around the other axes are induced. The system is let to oscillate until the system appears to be relatively stable. Waiting longer, the signal to noise ratio becomes too great for meaningful data interpretation.

The system is also put through going over the pipes welding kinks. The R.S. V3 is set to roll over multiple different ones. The data recorded showing the worst behaviour is taken for analysis.

Lastly the system is brought to an abrupt halt to assess the responsiveness. The aim is to test the system reacts within the requirement of 2 [s]. This test is done at close range (within 5 [m])

and across multiple compartment at a distance of roughly 220 [m]. The distance is to assess the range requirement of 200 [m].

6.1.3 Results

The results of the tests are presented below. All images shown are zoomed-in for clear view of the blurriness level.

Induced Oscillations

In the same manner as for testing of V2, oscillation around the z-axis is induced. The most significant information from the IMU is displayed in Figure 42. From the frequency graph, it can be deduced that a frequency of 6.9 [Hz] is dominant. The corresponding image, Figure 43, has the timestamp of 4.6 [s], within high oscillatory motion. The image is fed to the blurriness index algorithm and has an index of 502.

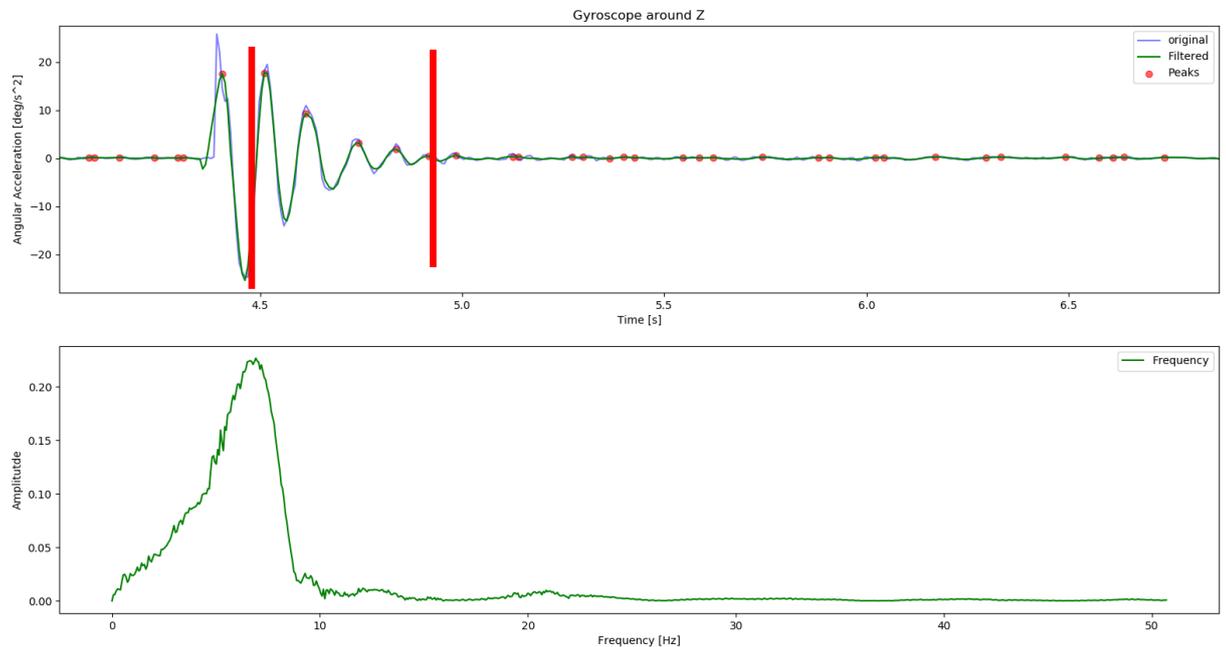


Figure 42 IMU data after induced oscillation around z axis



Figure 43 Image taken during the oscillation, with blur index of 502

Driving over the Weld

The system is driven over multiple welds, with the worst one presented here. It can be seen that both wheel passing over the weld induces each oscillations, with the second wheel creating the worst vibration. The first wheel is the one coated with rubber, explaining the more damp response. Figure 44 shows the IMU response around the Y axis Figure 45 around the Z axis. Figure 46 shows the image taken during the second wheel at time stamp 5.7 [s], and is computed to have a blurriness index of 575.

Form the IMU graphs, there is a multitude of frequencies appearing. Around the Y axis, there are spikes at 9.8, 12.5, 17.0 and 26.2 [Hz]. Around the Z axis, there are spikes at 1.5 and 17 [Hz].

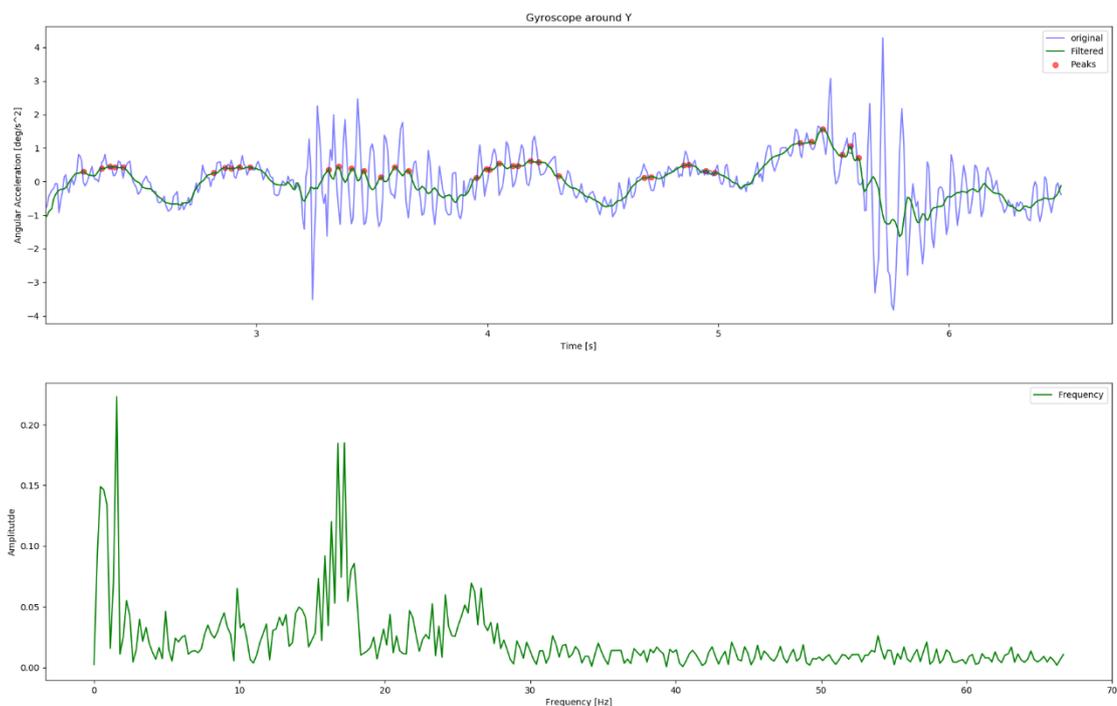


Figure 44 Gyroscope data of platform passing over a welding point around Y axis

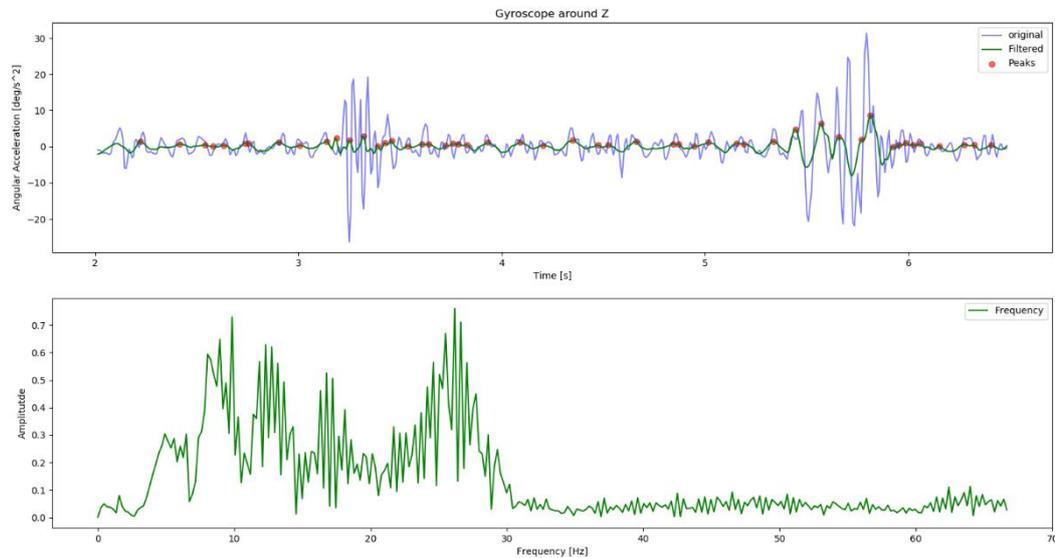


Figure 45 Gyroscope data of platform passing over a welding point around Z axis



Figure 46 Image taken during the oscillation of the second wheel passing over the welding point, with blur index of 575

Latency Assessment

The latency assessment at close range appeared to be instantaneous. For at longer range, the test was performed involving two persons, one issuing the command, and the other observing the system. Both persons are in communication over radio communication equipment. As the command is issued, the observer tells when the system stops. The communication equipment introduces an unknown delay, however, the latency in the system appeared to be below 0.5 [s].

6.2 Validation

During the induced oscillation test, the frequency recorded spikes at 6.9 [Hz], lower than the predicted 11.51 [Hz]. In the prediction, the mass moment of the beam is no taken into account, a mass that explains the difference. However, the frequency did increase from the original R.S. V2 which is at 0.819 [Hz].

The settling time is deduced from Figure 42, denoted in red lines. It is measured to be 0.6 [s], a major improvement from R.S. V2's 62.8 [s].

The damping ratio is calculated in the same manner as for R.S. V2, shown in APPENDIX A. The calculation table is shown below in Table 3. Since the oscillations are much less, less nodes are taken into consideration. That said, the ratio is of 0.125, an almost 10x increase in damping sharpness.

Table 4 Zeta Calculation R.S. V3

node	Amplitude X	delta	Zeta
0	17.73		
1	9.27	0.648475	0.102663
2	3.18	0.859188	0.135483
3	1.81	0.760644	0.120183
4	0.47	0.90757	0.142961
		Average	0.125322

The amplitude is calculated on the same nodes as for the damping ratio. The maximum amplitude encountered is 0.17 [deg], drastic reduction from previous . A graph of the amplitude over time is found in Figure 47

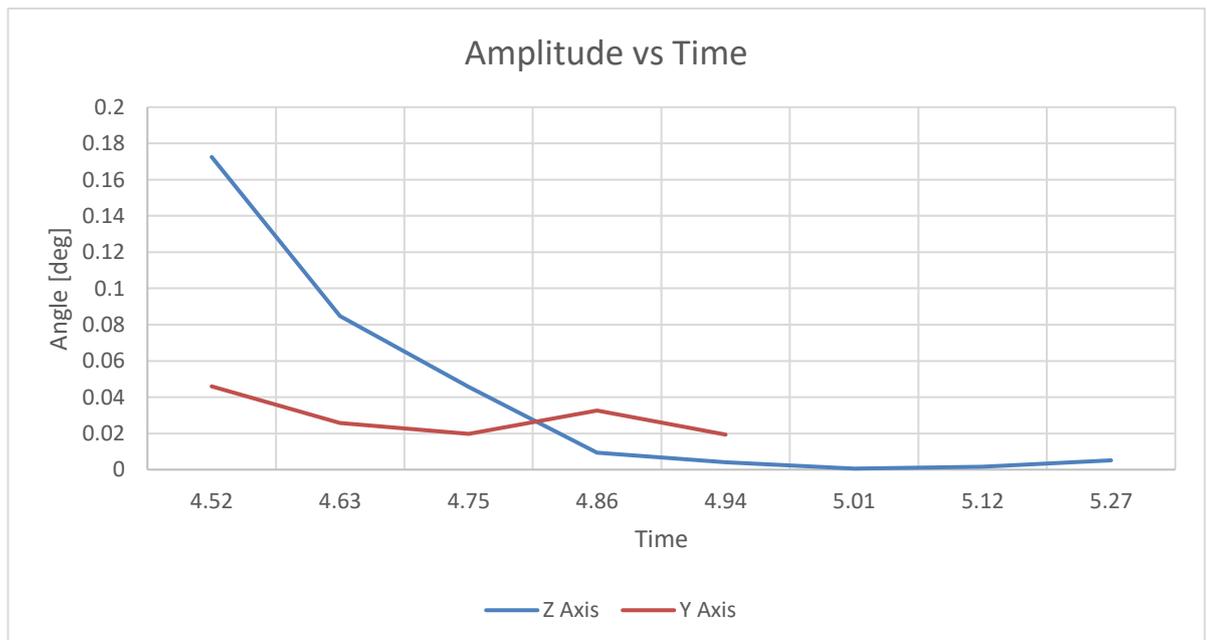


Figure 47 Graph of Amplitude over time

The images taken during oscillation have the blurriness computed, with the lowest value during induced oscillation. The index has been calculated to be 502, almost double then the previous R.S. V2's 254.

When following a key landmark across a series of images, the variation of pixel location differs by a maximum of 11 pixels.

Table 5 Requirement compliance table

Reference	Description	Requirement achievement
Req-01	The structure shall be usable by maximum of one person	During testing, the system has been handled only by one person
Req-02	The system shall have a maximum weight of 10 [kg]	The weight of the system is weighted at 3.84 [kg]
Req-03	The system shall be able to scout a lane in less than 10 minutes	The system took 6.2 [min] to go through an entire lane.
Req-04	The system shall have oscillations in any direction with an amplitude of less than 3 degree	The oscillations have been calculated to be 0.17 [deg]
Req-05	When oscillation occurs, the settling time shall be below 1 [s]	The system has a settling time of 0.6 [s]
Req-06	The system shall not interfere with any other object in the vicinity of the heat-pipes	No object has interfered during runs
Req-07	The system shall be operable by one person	During testing, the system was operated by only one person
Req-08	The system shall have as longest dimension in any direction a maximum length of 2.5 [m]	The longest dimension, in diagonal, is of 0.56 [m]
Req-09	The system shall be able to operate for a continuous time of at least 30 [min] without human intervention	The system has run in the greenhouse for 1.5 hours with no interruption
Req-10	The system shall react to user input at any given time with a latency no bigger than 2 [s]	The latency measured is of maximum 0.5 [s]
Req-11	Produced images shall have a blurriness index above 300	The lowest index recorded is 502
Req-12	Images shall be consistent through a same run, with a maximum deviation of landmarks of 1 % within the image.	The variation of pixels is maximum of 11 pixels, less than the 45 required
Req-13	The communication between the user and the system shall be uninterrupted over a distance of at least 150 [m]	The maximum range tested was of 220 [m]

All in all, almost all requirements are met. Only Req-11 is not met. That said, the results from tests shows that it is very close to the desired value.

7 CONCLUSIONS & RECOMMENDATIONS

This chapter presents the conclusions and recommendations for the robotic platform re-design, in section 7.1 and 7.2 respectively.

7.1 Conclusion(s)

HiPerGreen's Rail System came a long way from V1 to the V3. V3's re-design of the structure, the electronics and the software has proven that a row can be inspected easily and with meaningful images, thus answering the main research question:

How can HiPerGreen's pre-existing robotic platform be modified to effectively produce good imagery and be user efficient?

Analysis of the greenhouse pointed towards some potential difficulties: the pipes on which the platform hangs tend to be dirty, have kinks due to welding and aren't straight. Several elements are placed near the pipes, leaving limited space to incorporate a motorised platform. Lastly, the climate is worth taking note, as the temperature is as high as 30 [°C] and the relative humidity can exceed 95%. Altogether, this has created a good understanding of the restrictions and limitation imposed by the environment, and thus answering sub-question I.

Sub-question II is answered through a research of existing robotic platform within the industry. Current HiPerGreen's platforms are heavily inspired by those existing platforms. The first version was too big, not strong enough and never got to be tested as it broke apart upon installation. The second version, although much smaller, produced blurry images and was difficult to operate, especially in unknown environments. To fully understand the second version, an in-depth test is performed, from which the vertical beam is pointed to be the main source of oscillations, a major source for causing the blurred unusable images. This test concludes the answer of sub-question IV.

Based on the problems outlined through testing, a set of 13 requirements are made. The set of requirements are the basis for the new design, denoted version 3. The set of requirements answers the sub-question III.

The new design took a simplification approach to meet the requirements. After a literature research into torsion, the overall system is designed to be compact using short beams. The mount for the wheels is re-designed to be much sturdier and a new wheel mount design is fitted. Oscillations are reduced drastically, from 12.6 [deg] to 0.17 [deg]. Furthermore, the settling time has also been drastically reduced from 62.8 [s] to 0.6 [s]. The design answers sub-question V.

An algorithm to determine blurriness in images has been created. The algorithm outputs an index based on sharpness of structure in the image. Through testing and discussions with HiPerGreen researchers, the index threshold has been set at 300. A crisp clear image tends to have an index above 600, whereas unacceptable images are towards 250. During the worst oscillations, the new system produced images with an index around 500.

The electronics have been modified mostly around the computing unit. It consisted previously of more than one CPU, making synchronisation very complex and made it all very prone for error. The new design has only one CPU, that reacts directly to the users input. This answers the sub-research question VI. To enable fast reaction without any delay in actuation, it was found that efficient programming is important. Data types are an essential start, followed by optimised routines to get rid of delays. Lastly, the introduction of interrupt routines allows for a simple computing unit to handle a main routine and a (very light) secondary routine. Combining those for-mentioned methods, the software and control are able to serve the user in the right manner, and thus answering sub-research question VI.

Through testing, the design was validated to comply with all but one requirement and thus answering sub-question VII. The system reacts to the user input with a maximum lag of 0.5 [s]

and is able to go through a whole row with no issues. Rolling over kinks made little oscillation in the system, and dirt made no opposition.

Although one requirement is very close to being met, all other have been met. The robotic platform is well accepted by the researchers at HiPerGreen and is considered for commercialisation.

7.2 Recommendations

The skeleton of the system is using aluminium Bosch profile, which offers high versatility in terms of attachment. However, these are relatively heavy and are bad at taking moments. As soon as the positions of elements are set, the next step would be to manufacture the skeleton in a different material and a different cross-section. An idea would be a tubular cross-section made from glass fibre.

The wheels have been bought. However, they come at a relatively high price for a relatively simple wheel. Research can be made towards using additive manufacturing technics with materials such as nylon. This would decrease the cost of manufacturing while still using material that can withstand the pipe temperature of up to 70 [°C].

8 REFERENCES

- Diatom autofocusing in brightfield microscopy: a comparative study.* **Pech-Pacheco, J. L. and G., Cristobal.** 2016. Granada, Spain : Imaging & Vision Dept. , 2016.
- Android Developers.** 2016. PWM. *developer for android.* [Online] Android, 2016. [Cited: 08 March 2019.] <https://developer.android.com/things/sdk/pio/pwm>.
- Arduino Corp.** 2019. About Interrupt Service Routines . *Attach Interrupt on External input.* s.l. : Arduino, 2019.
- Barbir, F.** 2013. *PEM Fuel Cell (Second Edition).* Delft : Elsevier, 2013.
- Beards, C.** 1995. *Engineering Vibration Analysis with Application to Control Systems, 1st edition.* s.l. : Butterworth-Heinemann; 1 edition (July 1, 1995), 1995. 9780340631836.
- Boode, A.H.** 2018. *On the Automations of Periodic Hard Real-Time Processes.* Enschede : University of Twente, 2018. 978-90-365-4551-8.
- Brander, T.** 2018. *Interview on Greenhouses.* Delfgauw, 18 December 2018.
- Breukers, A., Hietbrink, O. and Ruijs, M.** 2008. *The power of Dutch greenhouse, An analysis of the private sector and its institutional framework.* The Hague : LEI Wageningen UR, 2008.
- Centraal Bureau voor de Statistiek .** 2019. StatLine, Bedrijven; Bedrijfstak. CBS. [Online] CBS, 2019. [Cited: 08 February 2019.] <https://opendata.cbs.nl/statline/#/CBS/nl/dataset/81589ned/table?ts=1549631279302>.
- de Beer, C.** 2018. Informative Session on Greenhouse Scouting Procedure. *HiPerGreen Validation.* Delft, June 2018.
- Dejan, P.** 2019. How a Stepper Motor Works. *How to Mechatronics.* [Online] HowToMechatronics.com, January 2019. [Cited: 07 March 2019.] <https://howtomechatronics.com/how-it-works/electrical-engineering/stepper-motor/>.
- electronics-course.com.** 2019. Stepper Motor Circuit. *electronics Course.* [Online] Enabt PTe Ltd, 2019. [Cited: 07 March 2019.] <http://electronics-course.com/stepper-motor-circuit.200210155R>.
- Inertial Navigation – Forty Years of Evolution.* **King, A.D.** 1998. NO. 3, London : Marconi Electronics Systems, Ltd., 1998, Vols. GEC REVIEW, VOL. 13.
- Joshi, N, et al.** 2015. *Image Deblurring using Inertial Measurement Sensors.* s.l. : Microsoft Research, 2015.
- Kooten, Olaf van.** 2016. Green eyes: Turning growers into scientists. *Greentech Community.* [Online] 22 4 2016. <https://community.greentech.nl/crops/b/knowledge-inspiration/posts/green-eyes-turning-growers-into-scientists>.
- Lalwani, M.** 2019. *Technical Report, Greenhouse Sensing Robotic Platform.* Delft : Inholland, 2019.
- Maaz, A.** 2016. Is gyroscopic effect in aeroplane during turning desirable? *Quora.* [Online] Quora, 15 May 2016. [Cited: 27 January 2019.] <https://www.quora.com/Is-gyroscopic-effect-in-aeroplane-during-turning-desirable>.
- Meijer, Roy.** 2018. Subsidence in the Netherlands greater than expected. *Delft University of Technology.* 2018.
- Nagaraja, Dr. Mamta Patel.** 2019. Geodetic Precession. *NASA Science.* [Online] NASA, March 2019. [Cited: 29 March 2019.] <https://science.nasa.gov/science-news/science-at-nasa/2000/geodetic>.
- Naman-Bhalla and Rex, Abhi.** 2016. C++ Data Types. *Geeks for Geeks, a computer science portal for geeks.* [Online] 2016. [Cited: 07 March 2019.] <https://www.geeksforgeeks.org/c-data-types/>.
- Nave, R.** 2019. Hyper Physics. [Online] 2019. [Cited: 03 01 2019.] <http://hyperphysics.phy-astr.gsu.edu/hbase/shm.html>.
- Rexroth, Bosch Group.** 2016. *Aluminium Structural Framing System.* Stuttgart : Bosch Rexroth AG, 2016.
- Robur.** 2019. Robur Holland Afbeeldingen. *Robur Holland.* [Online] 2019. [Cited: 21 01 2019.] <http://roburholland.nl/>.

Rosebrock, Adrian. 2015. Blur detection with OpenCV. *PyImageSearch*. [Online] PyImageSearch, 7 September 2015. [Cited: 19 March 2019.] <https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/>.

Scherz, P. and S., Monk. 2013. *Practical Electronics for Inventors, Third Edition*. Chicago, USA : McGraw-Hill Education, LLC., 2013. 978-0-07-177133-7.

Srivastava, B. 2018. *Technical Report, 3rd Year Internship at HiPerGreen*. Delft : Inholland , 2018.

Stripek, Jan and Spielberg, Steven. 2012. Cinematography. *Shutter speed in Cinema*. [Online] Cinema Shock, 30 July 2012. [Cited: 19 March 2019.] <https://cinemashock.org/2012/07/30/45-degree-shutter-in-saving-private-ryan/>.

Thompson, Dehleh &. 2014. *Theory of Vibrations with Applications*. 2014.

Whittaker, E.T and Robinson, G. 2005. *The Calculus Of Observations*. s.l. : Blackie And Son Limited, 2005. ark:/13960/t2988397k.

APPENDIX A TESTING FOR OSCILLATION OF R.S. V2

During system instalment tests, oscillations has been observed and need deeper understanding. This section outlines the tests performed to further understand the oscillations occurring.

A.1 Setup

The system is tested in office, on a demo pipe. The pipe is a Polyvinyl chloride (PVC) tube of 67 [mm] diameter, secured at 3.4 [m] above the floor. The system is placed on it, and a sensing module is placed at two location: at the bottom T intersection between the vertical beam and the lateral beam, and at the top T intersection of the same two beams. The aim if to understand the period of the osculation and its amplitude.

The sensing module consists of two elements: a microcontroller (Arduino Uno R3) and an Inertial Measurement Unit (IMU) (TDK-InvenSense MPU9250). The microcontroller reads data from the IMU at roughly 200 [Hz] and transfers it to a computer, on which a program reads the data and saves it in an orderly fashion. A snippet of the microcontroller source code, the source code for saving the data and a snippet of the recorded data can be found in sections A.5 through A.7.

The IMU consists of three sensors: accelerometer, gyroscope and compass. The accelerometer reads the linear acceleration in three axes, the compass reads the earth magnetic field strength and direction to deduce the heading of the IMU, and the gyroscope reads rotational accelerations around all three axes.

It is worth mentioning that generally, except for high performance IMUs, the accelerometer and the compass tend to be very noisy but stable, and the gyroscope tend to be very accurate but will drift with time (Inertial Navigation – Forty Years of Evolution, 1998).

A.2 Test

Once the system is stable on the pipe, readings are started and recorded. The system is then excited around the z-axis by manual means in such a way that no other oscillations around the other axes are induced.

The system is let to oscillate for around 1.5 [min], until the system appears to be relatively stable. Waiting longer, the signal-to-noise ratio becomes to great for meaningful data interpretation.

The test is performed multiple times until clean data sets are acquired.

A.3 Result

Although linear acceleration, compass heading and rotational acceleration is measured, only a few of the parameters are presented. The gyroscope around the z-axis in both positions shows clearly the behaviour. The compass when placed at the bottom shows the best result for compass readings. All other parameters are well into the noise ratio, and thus no clear conclusion can be obtained.

The results are presented in the form of graphs in Figure 48, Figure 48 and Figure 50. These figures present the original data, filtered data and the positive peaks for analysis. The scale on the y axis of these graphs are not constant and worth paying attention to.

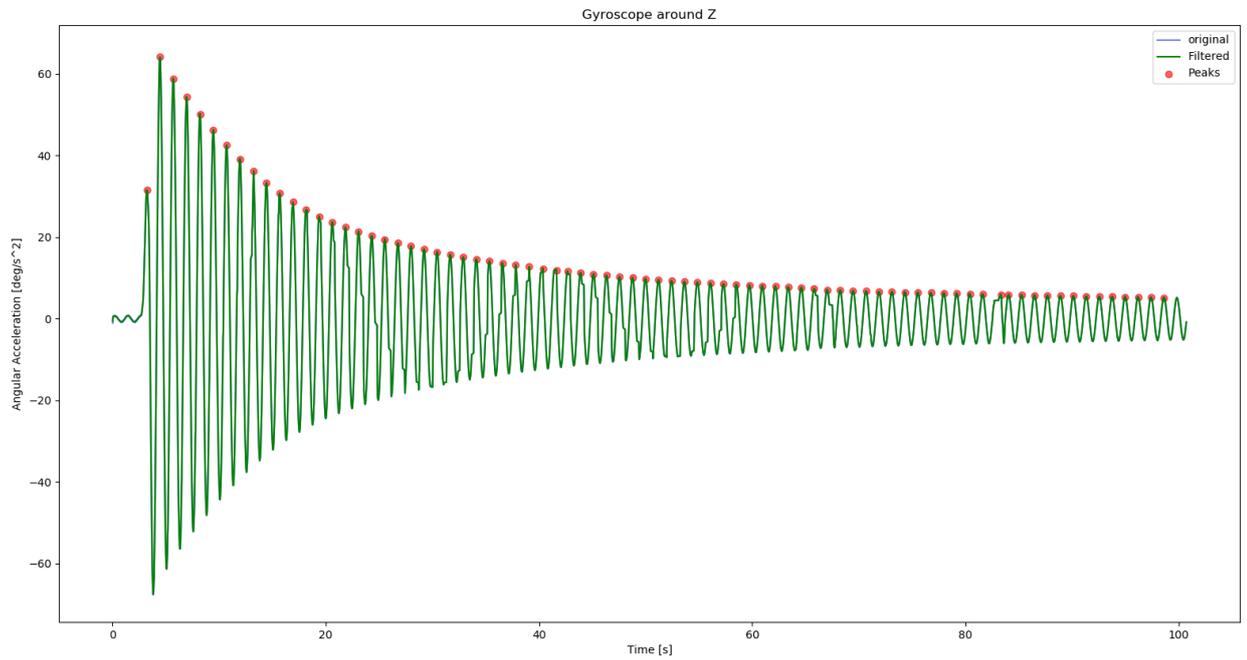


Figure 48 Graph of gyroscope around Z-axis, placed at the bottom

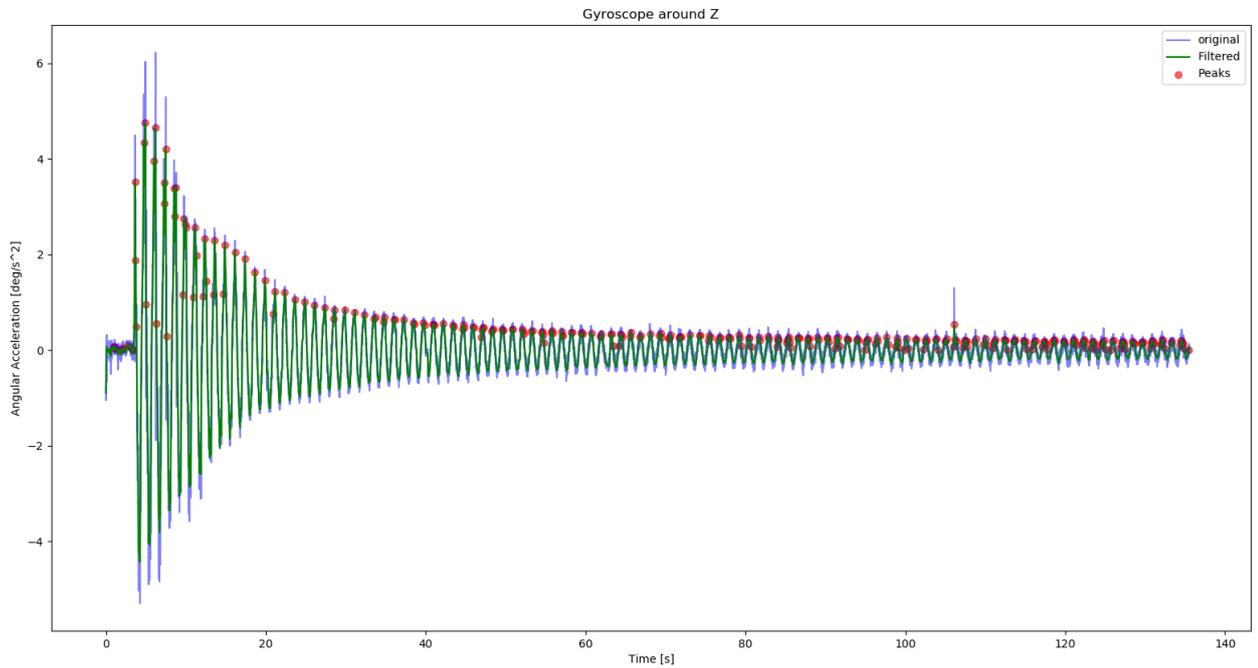


Figure 49 Graph of gyroscope around Z-axis, placed at the top

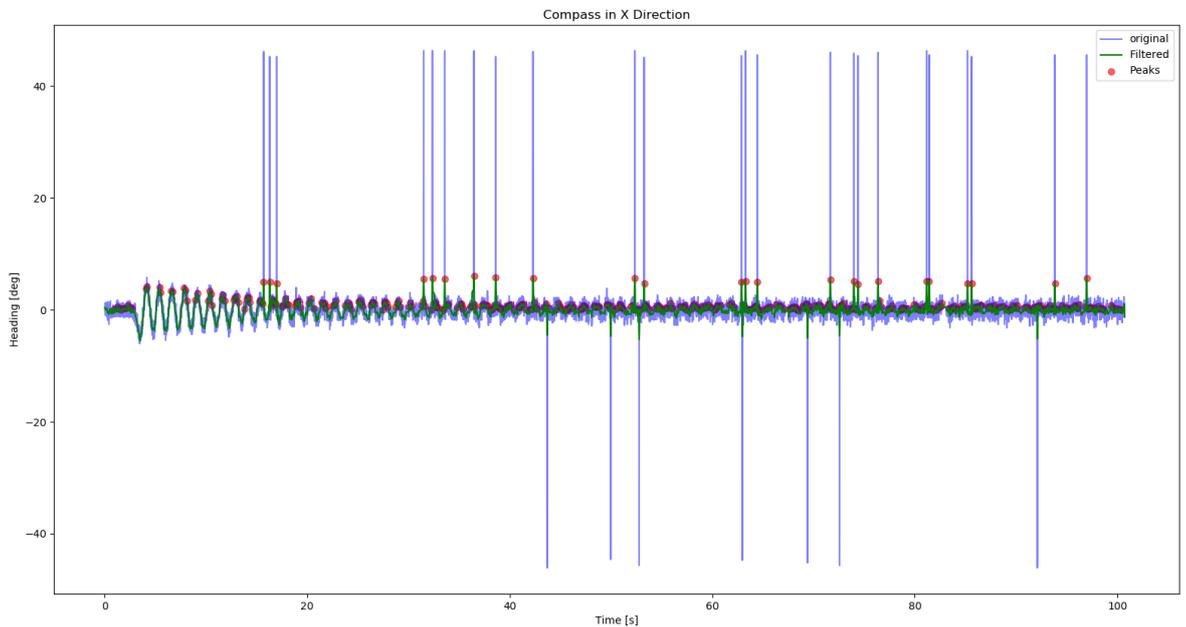


Figure 50 Graph of compass in X direction, placed at the bottom

A.4 Analysis

The first step in the data interpretation is filtering for better reading and concluding. The method of Savitzky-Golay, commonly known as Savgol, is used. *A Savitzky-Golay filter is a digital filter that can be applied to a set of digital data points for the purpose of smoothing the data, that is, to increase the signal-to-noise ratio without greatly distorting the signal. This is achieved, in a process known as convolution, by fitting successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares.* (Whittaker, et al., 2005). An example of it can be seen in Figure 51.

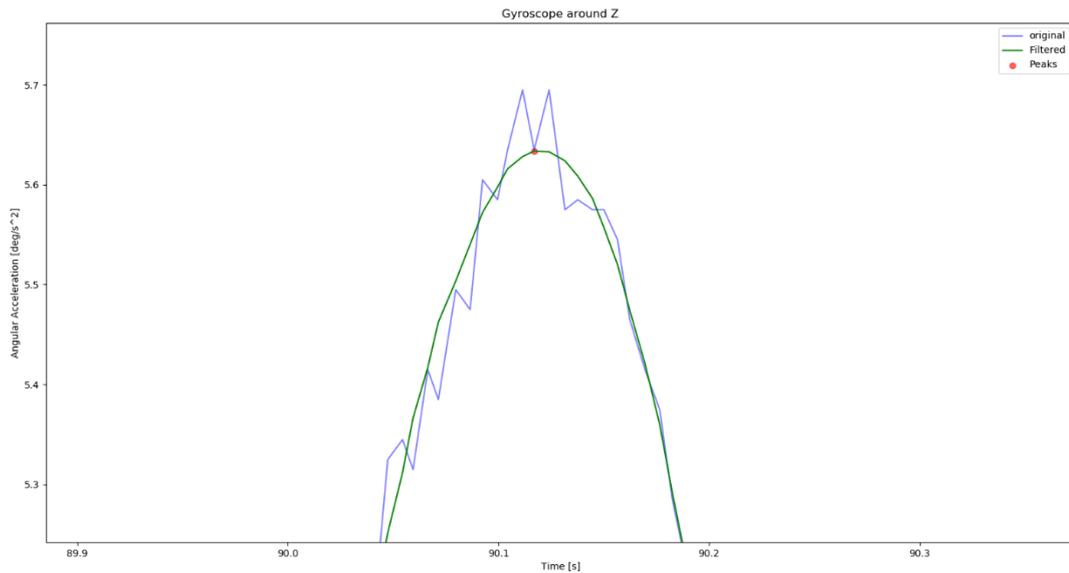


Figure 51 Graph of example for Savitzky-Golay filtering result

The period can be determined by looking at the average time between the peaks. With the data at the bottom, the period is of 1.22 [s], with a standard deviation of 0.084 [s]. With the data in the top, the period is of 1.23, with a standard deviation of 0.10 [s]. It can be seen that with lower data points, the noise to signal ratio is higher, and thus the data from the bottom has a lower standard deviation. The period is therefore taken from the data taken at the bottom, with a period of 1.22 [s].

A second method to calculate the period is to use the Fourier Transform. The Fourier Transform transfers the signal from the time domain to the frequency domain. Previous graph are plotted with the frequency domain version in Figure 52, Figure 53 and Figure 54. From both the graph from the gyroscope and compass placed at the bottom a frequency of 0.819 [Hz] can be read, corresponding to a period of 1.22 [s]. From the gyroscope placed on the top, a frequency of 0.808 [Hz] can be read, corresponding to a period of 1.23 [s].

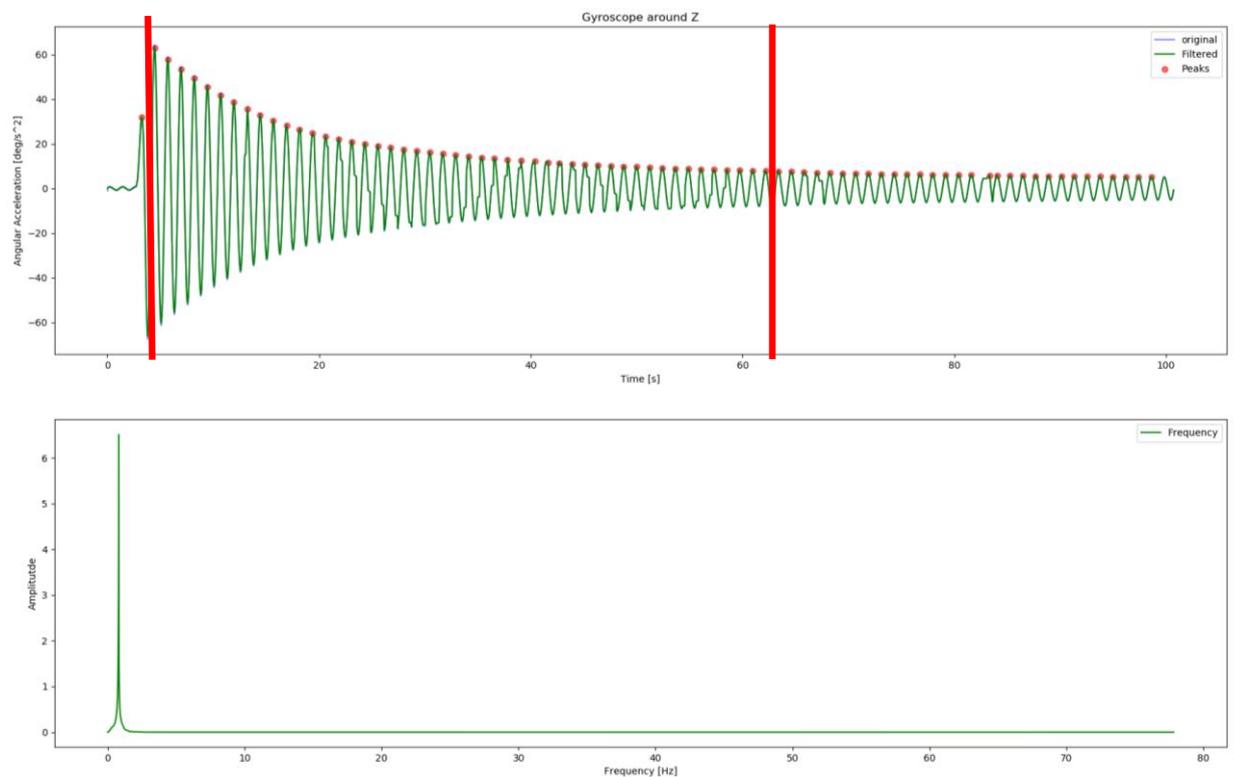


Figure 52 Graph of gyroscope around Z-axis, placed at the bottom, in the time domain and frequency domain

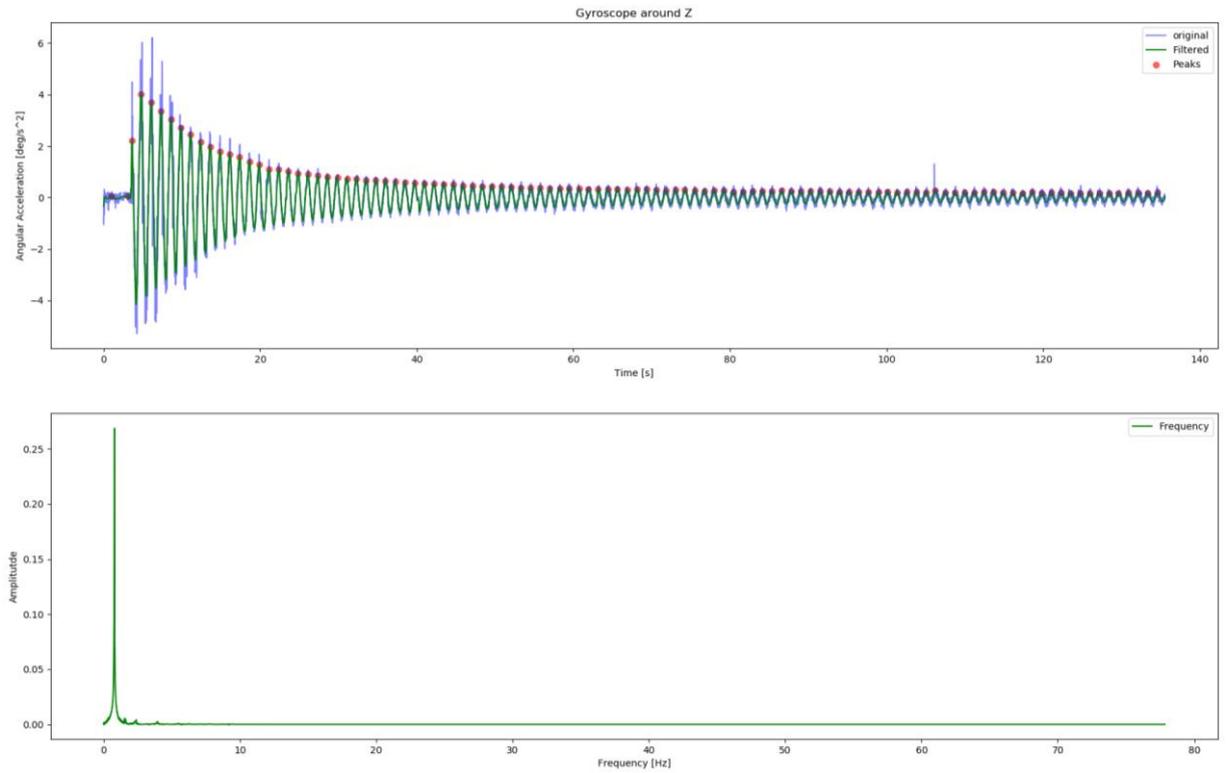


Figure 53 Graph of gyroscope around Z-axis, placed at the top, in the time domain and frequency domain

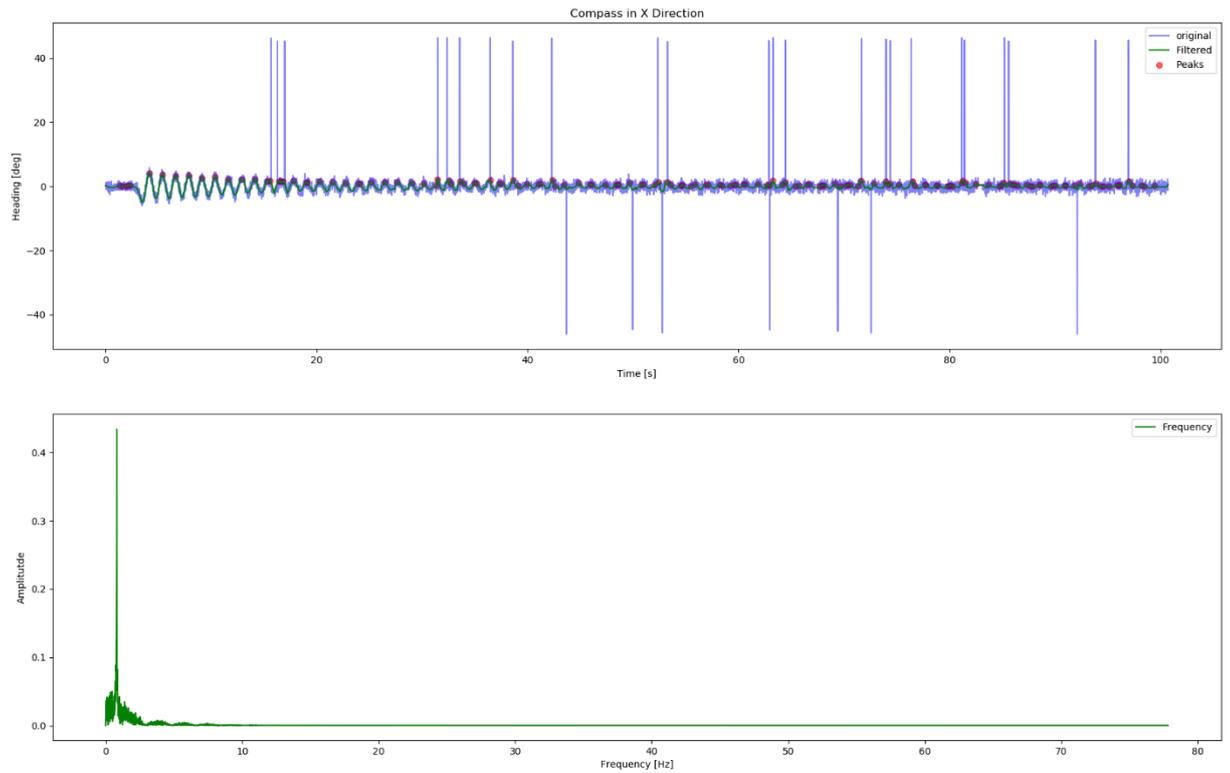


Figure 54 Graph of compass in X direction, placed at the bottom, , in the time domain and frequency domain

The amplitude is calculated by integrating the data from the gyroscope during falling time. The maximum is read to be 12.6 [deg] Graphs are derived and presented in Figure 55. The y axis has a logarithmic scale.

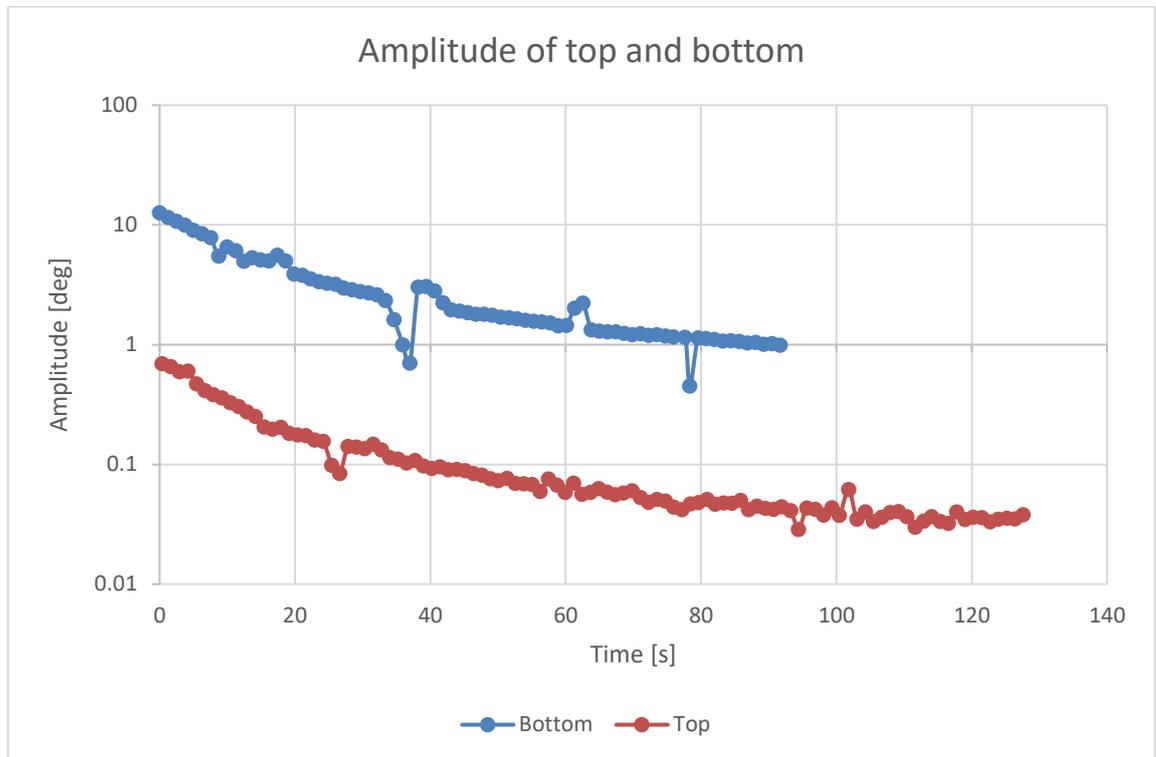


Figure 55 Amplitude vs time, bottom and top

The settling can be deduced from Figure 52, denoted in the red lines. The settling time is measured to be 62.8 [s].

By comparing the data set taken at the top and at the bottom, it can be concluded that a lot of the oscillation is happening in the vertical beam. The period is the same across the whole system, with a very similar damping speed. However, the amplitude is vastly bigger at the bottom of the beam compared to the top.

The damping ratio can be calculated by means of experimental determination. This is done by calculating the reduction of amplitude over time (Beards, 1995). This is performed from the data from the bottom, as seen in Figure 48 and Figure 52. The damping ratio can be calculated using the following formula:

$$\zeta = \frac{1}{2\pi} \ln \frac{X_1}{X_2}$$

However, this is valid only for adjacent amplitudes. For over “n” number of amplitudes, a delta function is added. The peak amplitudes are recorded, then delta is calculated with the following formula:

$$n\delta = \ln \frac{X_i}{X_{i+n}}$$

Damping ratio can then be calculated using the following formula:

$$\zeta = \frac{\delta}{\sqrt{(2\pi)^2 + \delta^2}}$$

As the measurements are never too accurate, an average over the first 12 nodes are taken to produce the final result, as shown in Table 6. The final Zeta value is 0.013, confirming that the system is under-damped.

Table 6 Zeta Calculation R.S. V2

node	Amplitude X	delta	Zeta
0	63.15		
1	57.95	0.085932	0.013675
2	53.57	0.082262	0.013091
3	49.52	0.081045	0.012898
4	45.81	0.080253	0.012772
5	41.88	0.082141	0.013072
6	38.74	0.08144	0.01296
7	35.59	0.081921	0.013037
8	33.01	0.081088	0.012904
9	30.31	0.081559	0.012979
10	28.06	0.081117	0.012909
11	26.27	0.079735	0.012689
		Average	0.012999

A.5 Source code of the microcontroller

```
1. #include <SparkFunMPU9250-DMP.h>
2.
3. // #define SerialPort SerialUSB
4.
5. MPU9250_DMP imu;
6.
7. void setup()
8. {
9.     Serial.begin(115200);
10.
11.     // Call imu.begin() to verify communication with and
12.     // initialize the MPU-9250 to it's default values.
13.     // Most functions return an error code - INV_SUCCESS (0)
14.     // indicates the IMU was present and successfully set up
15.     if (imu.begin() != INV_SUCCESS)
16.     {
17.         while (1)
18.         {
19.             Serial.println("Unable to communicate with MPU-9250");
20.             Serial.println("Check connections, and try again.");
21.             Serial.println();
22.             delay(5000);
23.         }
24.     }
25.
```

```
26. // Use setSensors to turn on or off MPU-9250 sensors.
27. // Any of the following defines can be combined:
28. // INV_XYZ_GYRO, INV_XYZ_ACCEL, INV_XYZ_COMPASS,
29. // INV_X_GYRO, INV_Y_GYRO, or INV_Z_GYRO
30. // Enable all sensors:
31. imu.setSensors(INV_XYZ_GYRO | INV_XYZ_ACCEL | INV_XYZ_COMPASS);
32.
33. // Use setGyroFSR() and setAccelFSR() to configure the
34. // gyroscope and accelerometer full scale ranges.
35. // Gyro options are +/- 250, 500, 1000, or 2000 dps
36. imu.setGyroFSR(500); // Set gyro to 500 dps
37. // Accel options are +/- 2, 4, 8, or 16 g
38. imu.setAccelFSR(2); // Set accel to +/-2g
39. // Note: the MPU-9250's magnetometer FSR is set at
40. // +/- 4912 uT (micro-tesla's)
41.
42. // setLPF() can be used to set the digital low-pass filter
43. // of the accelerometer and gyroscope.
44. // Can be any of the following: 188, 98, 42, 20, 10, 5
45. // (values are in Hz).
46. imu.setLPF(98); // Set LPF corner frequency to 98Hz
47.
48. // The sample rate of the accel/gyro can be set using
49. // setSampleRate. Acceptable values range from 4Hz to 1kHz
50. imu.setSampleRate(200); // Set sample rate to 200Hz
51.
52. // Likewise, the compass (magnetometer) sample rate can be
53. // set using the setCompassSampleRate() function.
54. // This value can range between: 1-100Hz
55. imu.setCompassSampleRate(100); // Set mag rate to 100Hz
56. }
57.
58. void loop()
59. {
60. // dataReady() checks to see if new accel/gyro data
61. // is available. It will return a boolean true or false
62. // (New magnetometer data cannot be checked, as the library
63. // runs that sensor in single-conversion mode.)
64. if ( imu.dataReady() )
65. {
66. // Call update() to update the imu objects sensor data.
67. // You can specify which sensors to update by combining
68. // UPDATE_ACCEL, UPDATE_GYRO, UPDATE_COMPASS, and/or
69. // UPDATE_TEMPERATURE.
70. // (The update function defaults to accel, gyro, compass,
71. // so you don't have to specify these values.)
72. imu.update(UPDATE_ACCEL | UPDATE_GYRO | UPDATE_COMPASS);
73. printIMUData();
74. }
75. }
76.
77. void printIMUData(void)
78. {
79. // After calling update() the ax, ay, az, gx, gy, gz, mx,
80. // my, mz, time, and/or temperature class variables are all
81. // updated. Access them by placing the object. in front:
82.
83. // Use the calcAccel, calcGyro, and calcMag functions to
84. // convert the raw sensor readings (signed 16-bit values)
```

```
85. // to their respective units.
86. float accelX = imu.calcAccel(imu.ax);
87. float accelY = imu.calcAccel(imu.ay);
88. float accelZ = imu.calcAccel(imu.az);
89. float gyroX = imu.calcGyro(imu.gx);
90. float gyroY = imu.calcGyro(imu.gy);
91. float gyroZ = imu.calcGyro(imu.gz);
92. float magX = imu.calcMag(imu.mx);
93. float magY = imu.calcMag(imu.my);
94. float magZ = imu.calcMag(imu.mz);
95.
96. Serial.print(accelX);
97. Serial.print(',');
98. Serial.print(accelY);
99. Serial.print(',');
100. Serial.print(accelZ);
101. Serial.print(',');
102. Serial.print(gyroX);
103. Serial.print(',');
104. Serial.print(gyroY);
105. Serial.print(',');
106. Serial.print(gyroZ);
107. Serial.print(',');
108. Serial.print(magX);
109. Serial.print(',');
110. Serial.print(magY);
111. Serial.print(',');
112. Serial.print(magZ);
113. Serial.print('\n');
114. }
```

A.6 Snippet of the computer program for reading and saving the incoming data

```
1. import serial
2.
3.     ser = serial.Serial(port = 'COM15', baudrate = 115200, timeout = 0.5)
4.     Dataset = []
5.     for i in range(5):
6.         ser.readline() # clearing the incoming buffer
7.         print('\t\tRecording!!!!')
8.         start = time.time()
9.         while 1:
10.
11.             try:
12.                 temp = ser.readline()
13.                 temp = temp[:-2]
14.                 temp = str(temp)
15.                 temp = temp[2:-1]
16.                 temp += ','+str(round(time.time()-start,4))
17.             try:
18.                 Dataset.append(temp)
19.                 # timestamp.append(round(time.time()-start,4))
20.                 if len(Dataset) > 200:
21.                     print (temp)
22.                     foo = open(File, 'a')
23.                     for temp in Dataset:
```

```

24.         foo.write(temp)
25.         foo.write('\n')
26.         foo.close()
27.         Dataset = []
28.     except:
29.         continue
30. except KeyboardInterrupt:
31.     break

```

A.7 Snippet of the data recorded

accelX	accelY	accelZ	gyroX	gyroY	gyroZ	magX	magY	magZ	Timestamp
0.13	0	1	-1.01	0.02	0.63	-0.3	12.45	19.9	1.5035
0.14	0	0.99	0	0	0	-0.3	12.45	19.9	1.5204
0.14	0	0.99	-2.76	0.05	1.6	-1.2	17.4	20.2	1.5254
0.14	0	1	-3.8	0	2.4	-1.2	17.4	20.2	1.5324
0.14	0	1.01	-3.91	0.02	2.52	3.15	17.1	17.1	1.5374
0.14	-0.01	1	-3.79	0.03	2.58	6.3	13.65	19.9	1.5454
0.14	0	1	-3.92	0.09	2.64	6.3	13.65	19.9	1.5514
0.13	0	1	-3.83	0.06	2.49	1.95	9.9	23.1	1.5573
0.14	-0.01	1	-3.98	-0.05	2.53	1.5	14.25	18.3	1.5623
0.15	0	0.99	-3.82	0	2.5	1.5	14.25	18.3	1.5693
0.14	-0.01	0.98	-3.85	-0.09	2.52	3.75	16.8	15.7	1.5774
0.13	-0.01	0.98	-3.94	0.02	2.46	0.3	14.1	18.4	1.5823
0.13	0	0.99	-3.8	0.02	2.41	0.3	14.1	18.4	1.5903
0.13	0	0.99	-3.86	0.08	2.41	3.75	10.65	18.3	1.5952
0.14	-0.01	1	-3.94	0.02	2.35	4.05	7.8	20.7	1.6022
0.14	-0.01	1.01	-3.94	0.02	2.4	4.05	7.8	20.7	1.6072
0.14	0	1	-3.85	-0.05	2.49	5.7	12	15.3	1.6152
0.14	0	1	-4.02	0.03	2.4	5.7	12	15.3	1.6202
0.14	0	0.99	-3.86	0.11	2.27	0.3	18.15	15.6	1.6272
0.13	0.01	1	-3.83	0	2.23	1.05	11.55	17.7	1.6351
0.15	0	0.99	-3.95	0.03	2.27	1.05	11.55	17.7	1.6397

APPENDIX B LAPLACIAN VARIANCE COMPUTING SOURCE CODE

```
1. import cv2
2.
3. def variance_of_laplacian(image):
4.     # compute the Laplacian of the image and then return the focus measure, which is simply the variance of the Laplacian
5.     return cv2.Laplacian(image, cv2.CV_64F).var()
6.
7.
8. imagePath = input('drag and drop Image\n->').strip('')
9.
10. # Load image in RAM
11. image = cv2.imread(imagePath)
12.
13. # Convert to Grayscale
14. gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
15.
16. # Compute variance
17. fm = variance_of_laplacian(gray)
18.
19. print("Variance: ",fm)
```

APPENDIX C DATA TYPES IN C++

DATA TYPE	SIZE (IN BYTES)	RANGE
short int	2	-32,768 to 32,767
unsigned short int	2	0 to 65,535
unsigned int	4	0 to 4,294,967,295
int	4	-2,147,483,648 to 2,147,483,647
long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295
long long int	8	$-(2^{63})$ to $(2^{63})-1$
unsigned long long int	8	0 to 18,446,744,073,709,551,615
signed char	1	-128 to 127
unsigned char	1	0 to 255
float	4	
double	8	
long double	12	

Source: (Naman-Bhalla, et al., 2016)

APPENDIX D SOURCE CODE FOR R.S. V3

```
1. #include "AlexInterrupt.h"
2.
3. //#define DEBBUG
4.
5. byte Throttle_Input_Pin = 20;
6. byte DirPin = 8;
7. byte StepPin = 9;
8. byte EnPin = 10;
9.
10. AlexInterrupt THROTTLE_INPUT(Throttle_Input_Pin);
11.
12. long Throttle_Input_Value = 1500;
13. float SteppingInput;
14. byte Direction = 0;
15.
16. long MaxDelay = 300;
17. long MinDelay = 8;
18.
19. void setup() {
20. #ifdef DEBBUG
21. Serial.begin(115200);
22. #endif
23. pinMode(DirPin, OUTPUT);
24. pinMode(StepPin, OUTPUT);
25. attachInterrupt(digitalPinToInterrupt(Throttle_Input_Pin), ThrottleLoop, CHANGE);
26.
27. }
28.
29. void loop() {
30. ReadInput(); // Get radio signal
31. GetThrottle(); // Convert radio signal to stepping speed
32. Stepping(); // Step
33. #ifdef DEBBUG
34. PrintState(); // used only if debug mode is enabled
35. #endif
36. }
37.
38. void ReadInput(){
39. Throttle_Input_Value = THROTTLE_INPUT.duration;
40.
41. }
42.
43. void GetThrottle(){
44. if (Throttle_Input_Value >= 1600){
45. Direction = 0;
46. SteppingInput = (MaxDelay*(2000 - Throttle_Input_Value) + MinDelay*(Throttle_Input_Val
ue - 1600))/1600;
47. digitalWrite(EnPin, LOW); //Enabling the motor
48.
49. }
50. else if(Throttle_Input_Value <= 1400){
51. Direction = 1;
52. SteppingInput = (MinDelay*(1400 - Throttle_Input_Value) + MaxDelay*(Throttle_Input_Val
ue - 1000))/1600;
53. digitalWrite(EnPin, LOW); //Enabling the motor
54. }
```

```
55. else {
56.     SteppingInput = 0;
57.     digitalWrite(EnPin, HIGH);
58. }
59. }
60.
61. void ThrottleLoop(){
62.     THROTTLE_INPUT.Checker();
63. }
64.
65. void Stepping(){
66.     if (SteppingInput > 0){
67.         if (Direction > 0){
68.             digitalWrite(DirPin, LOW);
69.         }
70.         else{
71.             digitalWrite(DirPin, HIGH);
72.         }
73.         digitalWrite(StepPin, HIGH);
74.         delayMicroseconds(long(SteppingInput));
75.         digitalWrite(StepPin, LOW);
76.         delayMicroseconds(long(SteppingInput/2));
77.     }
78.     else{
79.     }
80. }
81.
82. void PrintState(){
83.     Serial.print("T Input:");
84.     Serial.print(Throttle_Input_Value);
85.     Serial.print("\tDirection:");
86.     Serial.print(Direction);
87.     Serial.print("\tT Output:");
88.     Serial.print(SteppingInput);
89.     Serial.print("\n");
90. }
```