DEVELOPMENT OF AUTOMATED WALL CLIMBING ROBOT FOR GLASS CLEANING

By

Geraldi Kolin
12001024

BACHELOR'S DEGREE
in

MECHANICAL ENGINEERING STUDY PROGRAM
MECHATRONICS CONCENTRATION
FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

SWISS GERMAN UNIVERSITY
The Prominence Tower
Jalan Jalur Sutera Barat No. 15, Alam Sutera
Tangerang, Banten 15143 - Indonesia

Revision after the Thesis Defense on 2nd of July 2024

STATEMENT BY THE AUTHOR

I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgement is made in the thesis.

Geraldi Kolin — 16/07/2024

Student — Date

Approved by:

Edward Boris P. Manurung, M.Eng — 16/07/2024

Thesis Advisor — Date

Yohanes Fredhi S.P. — 16/07/2024

Thesis Co-Advisor — Date

Dr. Maulahikmah Galinium, S.Kom, M.Sc

Dean — Date

ABSTRACT

DEVELOPMENT OF AUTOMATED WALL CLIMBING ROBOT FOR GLASS CLEANING

By

Geraldi Kolin
Edward Boris P. Manurung, M.Eng, Advisor
Yohanes Fredhi, S.P., Co-Advisor

SWISS GERMAN UNIVERSITY

This study explores the development of an autonomous skyscraper cleaning robot, which involves various aspects crucial for its functionality. The study provides an overview of the research, emphasizing the need for automated cleaning solutions on high-rise buildings. The research on window cleaning robots involves surveys, and includes notable projects, highlighting advancements in sensor technology, control systems, and adhesion mechanisms. To outline the systematic approach used in the autonomous cleaning robot, the design justifications cover aspects such as conceptual design, component selection, prototype construction, and testing methodologies. The findings from experimental tests and analyses conducted, includes evaluations of adhesion systems, locomotion mechanisms, cleaning modules, and navigation algorithms. The overall load that the adhesion system can handle is on 11.569kg. On average the robot itself can manoeuvre a horizontal surface at around 60cm per minute, with the main hindrance being the testing method. Challenges encountered during development and troubleshooting strategies are also discussed. Overall, the result of this thesis proves that the adhesion system is very reliable, while the result of the locomotion system requires the design to be adjusted.

Keywords: Automation, Robotics, Window Cleaning, Wall Climbing Robot, Autonomous Mobile Robot

DEDICATION

I dedicate this thesis project for myself, for my parents, and also for the lecturers, friends, and engineers whom I respect.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor, Edward Boris Manurung M.Eng, and co-advisor, Mr. Yohanes Fredhi S.P. for seeing potential in this project, and supporting me throughout the development process. I would like to thank Mr. Yohanes Fredhi S.P. in particular for being able to give many recommendations both on the design and methodology of the project. They have given much insight on developing this technology as a whole.

Secondly, I would like to thank my family and closest friends for supporting me both emotionally and financially during the making of this project, and throughout my career in Swiss German University. They have been the best support for me to be able to push through difficult times.

Finally, I would like to thank my lecturers, collogues, and friend, who have helped me during this project, be it through the decision making, design selection, fabrication process, or even emotional support. Without them, I would say that this project wouldn’t have been completed as it is now.

TABLE OF CONTENTS

LIST OF FIGURES

Figures Page

LIST OF TABLES

CHAPTER 1 – INTRODUCTION

1.1. Background

Automation of tasks have become widely applied in various areas of the industry to cut manpower or reduce safety hazards. One of the tasks that can be solved using automation technology is window cleaning on skyscrapers. Since skyscrapers are tall buildings that are always exposed to the elements, it is necessary to maintain the surface area of these buildings, mainly the window panes that are degraded by pollution, rain, and dust. By cleaning the windows of these buildings, it can mainly mitigate the corrosion of these materials due to contaminants, as well as maintaining a clean appearance and helps with safety regulations. This type of task is considered to be hazardous due to its environment in a relatively high-altitude setting. Accidents on these types of situations can prove fatal to the workers and victims that might be present in the area, particularly on the ground. Along with that, the rise of pollution in Indonesia, particularly in the Jabodetabek area where most skyscrapers are located and where the pollution is mostly concentrated, the amount of cleaning that needs to be conducted has risen. Not only in Indonesia, but studies also show that with the increasing number of high-rise buildings, the tenants of said structures also seem to be increasingly demanding a higher frequency of window cleaning. There is also the matter of privacy, where high-rise buildings often use clear window panels, where most window cleaners can see through. Some may feel uncomfortable on having window cleaners on the outside of the window, especially during private events or situations.

Current methods of cleaning these buildings use manual labour, where cleaners are lowered using Bosun’s Chair or Suspended Work Platforms in front of the buildings to wipe the glass one by one which is done slowly over the course of time. This process can take several hours to complete just one side, and depending on the size of the building, could even take up over a day. The surfaces of the building are cleaned from the top to bottom, or vice versa. Typically, windows are cleaned one pane at a time using a rubber wiper paired with a wiping cloth and spray bottle with some cleaning solution. The mechanism of lowering and rising the chair itself is also done either by hand or automatically using motors.

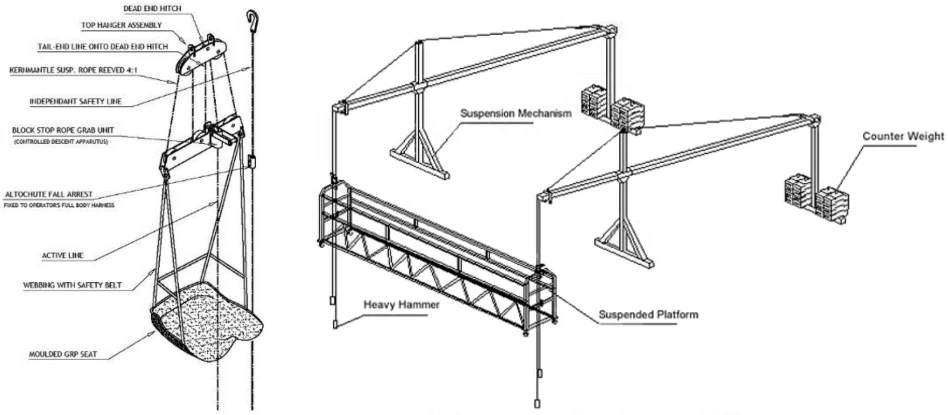


Figure 1.1. Bosun's Chair and Suspended
Platform System (Nadine Oosthuizen. 2020.).

To further maximize efficiency and safety, an automated robot can be utilized to replace the manual process of cleaning. This autonomous robot would be equipped with vacuum gripping system that would enable it to attach itself onto the vertical surface of a building. For the cleaning process, a cleaning mechanism with wiping and spraying is also installed on the side of the robot which faces the building surface. Automation of the task is sequenced using internal mapping and obstacle evasion so it can navigate itself on the building, with the help of auxiliary sensors providing feedback of both distance, position, and awareness. There are already a few commercialized and experimental systems that try to provide a solution to this problem; however, many of them cannot be considered a modern development, and many are still manually controlled.

This study will focus on the overall build of an autonomous robot fit to climb high-rise buildings. The author will contribute to the build, tuning, and testing of the robot in all aspects such as the mechanics, electrics, and programming. The main focus of this thesis is to further push the transition of manual labour into automated systems, utilizing modern materials, mechanical principles, and machine learning. It is also meant to ease the cost by having set singular expense instead of a subscription or wage of manual labour. It also reduces the dangers of cleaning high-rise buildings by eliminating hazards to and by the worker.

1.2. Research Problems

From the solution that has been proposed, the problems that need to be addressed are as follows,

1. How to attach the robot onto the surface of a skyscraper or building, specifically on the vertical glass panes.
2. How to effectively create the robot movement to ensure safe and reliable motion while also avoid obstacles that might be present.
3. How to clean the surface effectively from filth such as dust, grimes, and other contaminants.
4. How to navigate the machine autonomously so that it does not require constant surveillance.

1.3. Research Objectives

Based on the research problem that is mentioned, the formulated objectives are as follows,

1. To create a sufficient attaching force using vacuum suction to attach a robot onto a vertical surface.
2. To create a robot using a movement that allows it to move freely on a vertical surface, which allows obstacle evasion such as a different angled surface, and corners.
3. To create a cleaning module that can be used to remove filth on vertical surfaces.
4. To develop an algorithm for navigation of an autonomous robot on vertical surfaces.

1.4. Significance of Study

The main significance of this study is due to,

1. The creation of a safer environment by removing humans from hazardous places, which in this case is the high altitude of skyscrapers.
2. Since there is a rise of pollution in the greater area of Jakarta with skyscrapers, it is crucial to constantly maintain the cleanliness of high-rise buildings.
3. The development will further progress the transition from manual labour to automated systems.

1.5. Research Questions

Question #1: What is the ideal mechanism and structure for an automated window cleaning robot?

Question #2: What mechanism would be used for cleaning glass pane windows?

Question #3: What sensors would be beneficial for a robot on a high-rise building?

Question #4: What algorithm should be used for the navigation and decision making?

Question #5: What type of hardware would be compatible for developing the cleaning robot?

Question #6: What is the performance of the robot in quantifiable terms?

1.6. Hypothesis

Hypothesis # 1: A light weight body, combined with a negative pressure adhesion mechanism would be the most ideal system.

Hypothesis #2: A combination of rotary wiping with a cleaning solution would be the most ideal method in cleaning high-rise windows.

Hypothesis #3: A combination of proximity sensors, cameras, and gyro sensors would benefit the robot in to adapt in navigation and situational awareness.

Hypothesis #4: A machine learning algorithm with internal mapping would be the best method for navigation on a high—rise building.

Hypothesis #5: A combination of microcontrollers with custom PCBs would be ideal in making a good performing robot.

Hypothesis #6: The robot should be measured on the effectiveness, efficiency, and lifetime that it is able to handle.

CHAPTER 2 - LITERATURE REVIEW

2.1. Problem Research

As it stands, having a well-maintained building, especially the clear glass panes of a high-rise will not only benefit the appearance, but also the tenants that might inhabit the building. Having clear natural light that brightens a room will affect the tenant's mood, productivity, and overall well-being.

To emphasize the significance of replacing manual labour of window cleaning towards an automated system, a number of studies have been conducted to support this idea, one of which concerns the sustainability of tall buildings. The study suggests that, daily activities carried out to repair tall buildings and to clean their windows threatens the lives of workers. People often take the issue of window cleaning of skyscrapers lightly; however, it continues to be a frequent cause of the death of workers. Cleaning crews perform tasks manually by descending the height of the building from the roof to the ground floor while hanging on ropes and carrying water buckets and cleaning tools [1]. Not to mention that the result of the crew or their equipment falling might lead to civilian casualties on the ground. Using a robot that utilises high-gripping strength can also prove to be a safer method of cleaning windows, especially during high-wind speeds.

The main problem that can be stated from this study is that window cleaning is a hazardous task where cleaning mechanical systems have a possibility to fail. A good example of this is that it takes 36 window cleaners four months to clean the 26,000 windows of Burj Khalifa, which is the tallest building in the world. What makes the problem worse is that architects increasingly design complex shapes for skyscrapers, making it harder if an autonomous robot were to do the job. The window-washing problem should be the first not the last to addressed [1].

There is also a demand of window cleaning based on a survey done in Korea, which is driven towards building owners, building managers, and high-rise tenants. This study shows that most respondents (88%) said they should invest to additional window cleaning. As shown in 40% of the respofndents said that the frequency of cleaning should be 'monthly', and 28% of respondents answered 'weekly'. Therefore, the potential demand for a particular window cleaning can be considered sufficient [2]. To fulfil this demand, there needs to be device that is able to provide an efficient method of cleaning that is able to operate at a minimum of a weekly basis. Since manual labour can be considered costly and also dependant on the weather conditions on some high-rise buildings, machines are good alternative to not only reduce costs but also increase the frequency of this solution.

Survey	Response rate	Item
Respondent type	2%	Building owner
	6%	manager
	85%	Tenant
	7%	Others
Building type	15%	Multi-purpose buildings
	62%	Commercial buildings
	9%	Offices
	1%	Apartments
	13%	Other facilities
Floor that the respondent is on. ('5th floor' category includes all floors higher than the 5th floor)	9%	1st Floor
	42%	2nd Floor
	27%	3rd Floor
	6%	4th Floor
	15%	Higher than 5th floor
	1%	No answer
Window	97%	Installed
	0%	Not installed
	3%	No answer
Cleaning	90%	Needed
	6%	Not needed
	4%	No answer
Business type/field	11%	Clinics
	11%	General offices
	48%	Retailers (including clothes shop)
	21%	Service (Restaurants)
	9%	No answer

Figure 2.1. Survey Information
(K. Kim, 2019)

The demand for window cleaning has been on the rise among high-end shops in newly-built buildings. However, not only has the demand not been precisely grasped, but its related technology has also not been defined. For this reason, this study aimed to understand potential clients' demand for a window cleaning device and its related technology [2].

2.2. Mechanism Survey

For developing and creating a device that is able to clean windows on high-rise buildings, references from wall climbing robots (WCR) can be taken. Wall climbing robots can be defined as devices that are able to move around a vertical surface (around 90° angle) reliably through some mechanism to hold it to the vertical surface. WCRs can be adopted in a variety of application such as Non-Destructive Evaluation (NDE), diagnosis in hazardous environments, welding, construction, cleaning and maintenance of high-rise buildings, reconnaissance purpose, visual inspection of manmade structures. They are also used for inspection and maintenance of ground storage tanks and can be used in any type of surveying process including inspection of marine vessels, to detect damaged areas, cracks and corrosion on large cargo hold tanks and other parts of ship [10].

Specifically for window cleaning robots on high-rise buildings, there a few requiring factors that need to be taken into account. This includes,

- The robot should be compact in structure, light in weight, and easy to carry.
- To ensure the safety, there should be no slipping and falling during work.
- It provides high efficiency and quality of cleaning.
- It allows autonomous cleaning and easy manipulation.
- Fast-moving speed and good cleaning performance.
- Automatic identification and overcoming obstacles, such as window borders.
- Good adaptability to wall surfaces.

As the area of the exterior walls is relatively large for the robot to clean, the moving and cleaning speeds of the robot should be taken into account first. Usually, the cleaning robot or high-rise buildings needs to clean the entire surface of the curtain wall. So, the ability of transferring between surfaces is required for the robot [9]. A survey was conducted to record the methods that has been used on wall climbing robots for window cleaning purposes. These methods include,

2.2.1. Movement Systems

There are many methods to move the robot while on a vertical surface. the main specifications of a window-cleaning robot, including its working accuracy, speed, and cost, are heavily affected by the selected locomotion system. The robot locomotion mechanisms can be divided into the following categories:

- Wheel Locomotion

The wheel-driven window-cleaning robot has the advantages of simple structure, flexible movement, and good controllability. Hence, it can be widely used. Wheeled locomotion with differential drives is usually adopted to realize the continuous movement of the robot on glass surface. The is advantages of the wheeled mechanism are poor loading capacity and obstacle-overcoming ability, which are mainly attributed to the small contact area between the wheel and the wall. Due to the small contact area between the surfaces, it is difficult for the robot to adhere firmly on the wall. The adhesion force required by a robot is usually provided by suction cups and magnets. When there are obstacles or gaps on the wall, the adhesion force is unstable and induces the risk of falling off [9].

- Tracked Locomotion

The tracked locomotion mechanism has a similar motion principle and motion ability as the wheeled locomotion mechanism. As compared with the wheeled mechanism, the movement speed of this method is slower, and its turning ability is lower. On the other hand, this method has a larger contact area and better stability to prevent slipping. Its obstacle-overcoming ability and loading capacity are stronger. For example, Cleanbot II robot can overcome the obstacles of 6 mm and carry a weight of 25 kg. The robot GEKKO III uses two planar tracks to overcome 40 mm obstacles and achieves the leaning speed of 240 m2/h [9].

Figure 2.2. Transition Locomotion
(Z. Li, Q. Xu, L. M. Tam, 2021).

The working principle of the translation locomotion mechanism is similar to that of a simple two-leg locomotion mechanism. One leg is attached on the surface, and the other one moves to the target position. Therefore, its movement strategy can be summarized as sticking-moving-sticking mode. As it always maintains a stable adhesion force, the translation locomotion mechanism has a good load capacity. The main disadvantage of this mechanism is that its movement strategy is discontinuous. Thus, the movement speed is relatively slow. To further increase the movement step size, the volume of the translation locomotion mechanism is also increased [9].

Figure 2.3. Transitional Locomotion Example
(Z. Li, Q. Xu, L. M. Tam. 2021).

- Legged Locomotion

As compared with other motion mechanisms, legged wall climbing robots have the obvious advantages of multiple degrees-of-freedom and obstacle-crossing ability. Experiments were carried out in practical scenes, and the results showed that the developed robot can overcome the positive and negative obstacles in the glass panel with significant coverage performance. The safety and load capacity of the robot can be enhanced by increasing the number of legs. For instance, the number of legs can be increased from two to eight. However, as the number of legs increases, the corresponding mechanical structure and the control system will become more complicated. As its movement mode is discontinuous, its energy efficiency is relatively low. To improve the performance of the robot, Hirose et al. reported two types of designs, which are the coupled drive and gravitationally decoupled actuation, respectively. The active coupling drive mechanism was adopted to maximize the output power of the robot system. The gravity decoupling drive mechanism was employed to maximize the energy efficiency of the robot system [9].

- UAV Locomotion

The unmanned aerial vehicle (UAV) is usually adopted for performing non-contact tasks, such as surveillance, reconnaissance, and exploration of hazards. Actually, UAV has a great potential for the use in automated window-cleaning. To further ensure the transmission of information between the UAV and the control station, Miraj et al. proposed several methods to increase the capacity of the information transmission channel and the anti-noise ability. Although using UAV for cleaning work has many advantages, it exhibits a major disadvantage of short endurance [9].

- Cable-driven Locomotion

The cable-driven locomotion mechanism is commonly used in designing a window cleaning robot. This method relies on rope ascenders and a winch mechanism installed on the roof to support and navigate the robot. Such approach ensures a high degree of security, as the robot is always connected to the roof by ropes. To improve the obstacle overcoming ability of the cable driven system, Lee and Chu proposed a three-module cleaning robot. The robot was driven by a winch on the top of the building to achieve vertical movement. Each climbing module was equipped with a variety of sensors to detect obstacles and their states, and each module contained a separate lift drive device [9].

A comprehensive summary can be shown in the table below about the advantages and disadvantages of each method of locomotion.

Table 2.1. Summary Table of Locomotion (Z. Li, Q. Xu, L. M. Tam. 2021.).

Type	**Advantages**	**Disadvantages**
Wheeled Locomotion	Flexible movement and good controllability	Poor Load capacity and obstacle-overcoming ability
Tracked Locomotion	Better stability and slipping prevention	Weak turning ability
Tracked Locomotion	Good load capacity and controllability	Large size and low speed
Legged Locomotion	Multiple DOF and obstacle-overcoming ability	Low movement efficiency and complicated control system
UAV Locomotion	New technology with potential application	Short endurance and complicated control system
Cable-driven locomotion	High degree of safety	Good obstacle-overcoming ability

It is to be noted that one of the disadvantages of a cable-driven system is its bulky and complex application since it needs an anchor unit, as well as the cleaning unit itself on the wall surface.

2.2.2. Adhesion Systems

The adhesion system is the method of maintaining the attachment of the device on the surface of the building, which in this case is the glass panes. All the methods that have been used are listed below,

- Rail Guided Adhesion

The advantages of the rail-guided mechanism are safety and stability, because the robot can be directly installed on the rail of the building surface and performs cleaning tasks along the rail. However, reasonable rails need to be designed for mounting on the exterior walls of the building in advance, which will increase the deployment cost. Rail systems also exhibit some slipping action [11]. Moreover, this method cannot be applied to any existing buildings, which is also the biggest limitation of this method. Experimental results showed that the cleaning efficiency of the robot is up to 234.9 m^2/h [9].

Figure 2.4. Indonesian Rail Guided Cleaning Robot.
(K. Umam, Haryanto, R. Alfita. 2019)

- Propulsion Adhesion

The propulsion adhesion mechanism usually adopts propellers or fans to generate thrust force, which presses the cleaning robot against the wall surface. This method does not consider the air leakage, and the adhesion is more stable. Generally, the adhesion force provided by the propulsion adhesion mechanism is limited and insufficient to compensate for the gravity of the robot. Thus, additional lifting force, e.g., provided by rope pulling, is required. At the same time, the volume of the propulsion adhesion mechanism is generally large, and it is often accompanied by noisy voice [9].

- Negative Pressure Adhesion

The principle of negative-pressure adhesion is to produce an air pressure difference using atmospheric pressure to make the robot be adsorbed on the building surface. Negative pressure adhesion methods mainly include vacuum adhesion, vortex adhesion, and Bernoulli principal adhesion. The vacuum adhesion method creates a vacuum area between the robot and the wall. The area needs to be sealed to prevent the robot from falling off due to the air leak. According to the mechanism of vacuum adhesion, it can be divided into active adhesion mechanism and passive adhesion mechanism which are [9],

- Active Adhesion

Active adhesion is one of the main adhesion technologies for window-cleaning robots. This method has the advantages of reliable adhesion capacity and strong load capacity. It relies on a vacuum pump to provide continuous pressure, which can easily control the attachment and detachment of the vacuum adhesion. However, the vacuum pump will increase the weight and reduce the movement speed of the robot. A negative pressure sensor was used to detect the working condition of the vacuum pump, so as to determine whether to use the standby pump or not, improving the safety of the robot [9].

2.1.1.1.1. Passive Adhesion

Passive adhesion does not require a vacuum pump and other equipment, which has obvious advantages such as low power consumption, compactness, and light weight. The passive suction cups were made of elastic materials, and they can be simply pressed to the surface to exhale air to create a vacuum. Majority of the robot vacuum adhesion systems are concentrated on the application of technology. Whereas the research on its mechanical and dynamic performances is still insufficient [9].

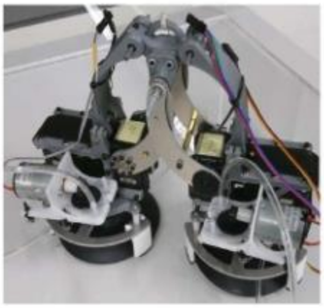


Figure 2.5. Negative Pressure Adhesion Example.
(Z. Li, Q. Xu, L. M. Tam. 2021.)

- Magnetic Adhesion

For the walls with high magnetic permeability, the magnetic adhesion method is usually adopted for the robot. Researchers have applied the magnetic adhesion method for ship welding, oil tank inspection, and other applications. Considering that the glass surface is a non-magnetic permeability surface, the magnetic adhesion mechanism cannot directly generate adhesion force. Hence, it is necessary to devise a special design for window-cleaning robots based on magnetic adhesion. Usually, the magnetic adhesion robot adopts the design of internal and external modules. Hence, it can clean both sides of the window at the same time, which improves the cleaning efficiency. However, for non-magnetic permeability surfaces such as glass, such method exhibits poor barrier-crossing ability. So, it is usually adopted in domestic environments [9].

- Electrostatic Adhesion

A more lucrative method of attaching a wall climbing robot to a vertical surface is by electrostatic adhesion. The electrostatic adhesion force generated by the electrode panel above depends on many parameters, including the voltages exerted on the electrodes, the permittivity of the wall and the insulating film, the thickness of the air layer and the insulating film and the structural parameters of the electrodes. In addition, atmospheric humidity and the electric conductivity of the wall and insulating film can also influence the electrostatic adhesion force to some extent [12]. The resulting adhesion force is quite small, making it inefficient on heavy duty equipment.

2.1.2. Cleaning Mechanism

For the device to be classified as a window cleaning device, it is required to be able to perform a cleaning action. Traditionally, window cleaning has been done with rubber wipers and a wiping cloth, but with the development in technology, things such as pressure washing, polishing, and other techniques can be used. There many mechanisms to achieve a cleaning action on glass panes, particularly for vertical surfaces on building which includes,

- Water Spraying Mechanism

Recently, Lee et al. have proposed a method of spraying high-pressure water using a pump and nozzle to improve the cleaning performance. Whereas this combination way of cleaning has the weakness of excessive use of water and will bring secondary pollution due to scattering and dripping of the contaminated water. To overcome such issue, researchers tried different water recycling technologies to improve the utilization rate of water resources [9].

- Manipulator Mechanism

The manipulator is an important part of the cleaning unit, which is used to connect the cleaning unit and robot body. The manipulator has the important role of changing the position and orientation of the cleaning unit. It is important to maintain a stable pressure between various types of surfaces and the cleaning unit, because it is mainly based on the friction between the cleaning unit and the surface to remove contamination [9].

2.1.3. Electrical Units

As with all electrical robots and devices, it will contain electrical components. The main components that are to be discussed below are the sensors and controllers which helps the machine run. Sensors and controllers are the foundation to develop an intelligent window-cleaning robot. In particular, the sensors are used to detect the robot's working status and to monitor the external working environment. The controllers are adopted to adjust the robot mechanisms to realize the locomotion, adhesion, and cleaning operation [9].

- Internal Sensors

Internal sensors can be classified, as sensors that are used internally in the robot to detect its own condition. This may include devices such as encoders, vacuum sensors, gyro sensors, and many more [9]. These types of sensors are usually used for,

- Rope Tension Detection

The tension detection is usually done by using tension sensors that are designed for metal wires. Due to the nature of this, the rope or tether used for tethered locomotion are limited to only using metal wires. Not only that, these sensors are usually externally attached from the robot, making it rather bulky.

- Adhesion Detection

There are many ways to detect the adhesion of a wall climbing robot. To detect the adhesion of a vacuum system, a pressure sensor (barometer) or vacuum sensor can be utilized on the suction. An alternative way to detect adhesion is through the load characteristic of a DC motor when under load, where it will draw more current to run. This means using an electrical current sensor can also be used to detect adhesion.

- Cleaning Solution Detection

Some robots are equipped with internal reservoirs to hold the cleaning solution on the robot itself, eliminating the need to supply the robot externally using a hose. These reservoirs have liquid level sensor to determine the amount of liquid solution that is present in the reservoir. These sensors can be in the form a buoyant actuated limit-switch, or an analogue liquid level sensor.

- Orientation Detection

To improve the accuracy of a wall climbing robot, orientation detection in the form of an Inertia Measuring Unit (IMU) and gyro sensors are used. These sensors can record the relative angle and orientation of the robot, providing more data to help the robot calculate the movement sequence resulting in more accurate and precise movements.

- Encoder Application

For wall climbing robots that utilizes linear actuators, robotic arms, and tracked locomotion, encoders are used to measure the precise position of the actuated part. Encoders are usually simple digital modules that send two signal pulses, which can be calculated to the amount of rotation. This data can be further processed into variables such as speed, axial position, rotational position, acceleration, etc. There are many types of encoders such as a rotary encoder, an optical encoder, and a magnetic encoder, all with different methods of creating the signal

- External Sensor

For situational awareness, obstacle evasion, and navigation, external sensors are used to measure the surrounding environment. These devices are primarily in the form of proximity sensors and vision sensors [9]. Common application of these devices are as follows,

- Wind Detection

For an added safety feature, wind detection is used to determine whether the current environmental conditions allow the robot to manoeuvre safely across the high-rise building. High force winds can be a detrimental condition that can cause the robot to detach, or become unstable. Wind detection is done using an anemometer attached to a section of the robot.

- Obstacle and Border Detection

For the robot to be able to avoid obstacles that are present on high rise building, it can be fitted with distance sensor to detect object around the area of the robot. The data from the sensors can then be fed into the algorithm to determine the correct moving sequence to avoid the obstacle. These sensors are usually in the form of ultrasonic sensors that are able to detect objects from a relatively large distance. Another method is to use limit switches to detect objects when it collides with the robot.

- Contamination Detection

By detecting the contamination of a surface, the robot can be sequenced to improve both the efficiency and effectiveness of the cleaning process. It can increase the speed of cleaning as it will have the ability to determine clean areas, and the resulting surface can be cleaned much more effectively if it is determined to still be contaminated. The detection of contaminants can be done through the usage of camera systems, where it uses a data set of images to determine contaminated surfaces.

- Self-Position Detection

Self-position detection here can be defined as the real-time position relative to a setpoint in an area, different to that of the orientation of the robot. To determine the position of the robot relative to a setpoint, a Global Positioning System module is used, as it can be integrated into many systems.

- Main Control Unit

The controller of a robot can be referred to as the brain of the device, which controls the overall movement, decision making, and actions of the robot. Controllers are designed to receive and transmit electrical signals to components. There are many types of controllers that have been employed on window cleaning robots, such as PLCs for heavy duty machinery and microcontrollers for more smaller and compact devices [9][13]. Recently, a novel control criterion using a fuzzy inference system has been proposed for a wall-cleaning robot to ensure the safety and reliability of operation [9].
The most common controllers for robotics are the single board computers from Raspberry Pi, as well as mini-PCs from Intel (NUC Mini PC) and NVIDIA (NVIDIA Jetson). Microcontrollers are also usually utilised for small and simpler design of a wall climbing robot. These controllers have robotic algorithms to calculate and execute the control system of the robots, one of which is the Robot Operating System (ROS). ROS Itself is not an operating system, but rather an open-source software that runs inside the controllers to help with things such as inverse kinematics.

2.2. Actuators Utilized

There are various actuators that are used based on the survey by 9. Z. Li, Q. Xu, L. M. Tam. 2021. “A Survey on Techniques and Applications of Window Cleaning Robots”, such as motors and pneumatic actuators. Such actuators can be listed below,

2.2.1. DC Motors

These motors are widely used due to their simplicity and ease of control. They convert electrical energy into mechanical motion and are ideal for providing precise speed and position control. Various types of gearboxes are also used to support the application of DC motors on wall climbing robots. One such types is a worm-gear transmission which only allows rotation from the motor side, which prevents the back-driving.

2.2.2. AC Motors

AC Motors are typically used for applications requiring high power and efficiency. AC motors are robust and durable, making them suitable for continuous operation in window cleaning robots, particularly for lowering the machines using a tether. These are mostly used for cable-driven locomotion.

2.2.3. BLDC Motors

These motors are known for high efficiency and reliability, and are commonly used in modern robotic applications, but require complex circuitry to allow them to be controlled. They offer precise control and have a longer lifespan compared to brushed DC motors. a BLDC machine is best suited for high speeds, while the overload capacity is best for Brushed DC machines [21].

2.2.4. Vacuum Pumps

These pumps are essential for creating suction that allows the robot to adhere to and move along vertical surfaces like windows. Vacuum pumps ensure the robot can maintain a stable position while cleaning the surface area. There various types of vacuum pump that utilizes different methods of moving the air, with the most common method being a diaphragm type pump.

2.2.5. Liquid Pumps

Liquid pumps are used to dispense cleaning solutions or water onto the window surface. These pumps control the flow of liquid, ensuring even distribution or high-pressure water for effective cleaning. Like the vacuum pump, there many types that uses different methods to displace the liquid, such as centrifugal, gear, peristaltic, diaphragm, etc. The most common method that is used for window cleaning robots is a diaphragm pump.

2.2.6. Motorized Linear Actuators

These actuators convert rotational motion into linear motion, allowing the robot to extend or retract parts of its cleaning mechanism. They provide precise and easy control over linear movements, but are slower when compared to other types of mechanism. These are usually used on robot that have two degrees of movement, which won't allow it to traverse between different angled surfaces.

2.2.7. Pneumatic Linear Actuators

These actuators use compressed air to produce linear motion. They are known for their high force output and are often used in applications where powerful, quick movements are required such as the ones used for the *Sky Cleaner 3*. Pneumatic actuators are usually used in heavy equipment, due to the high-power output but also has a higher weight. These actuators also require a compressor to be driven, which is not place on the robot itself, but attached with a long hose, unless the robot is designed to handle the weight.

2.3. Previous and Existing Research

2.3.1. Sky Cleaner 3.

A promising attempt was made in China to support the maintenance of the Shanghai Science and Technology Museum by creating a product name Sky Cleaner 3. The robotic system consists of three parts: 1) a following unit; 2) a supporting vehicle; 3) the cleaning robot. The robot is supported from above by cables from the following unit mounted on the top of the building. All following movements of the unit which protects against falling due to any type of failure are synchronized by the robot itself [3].

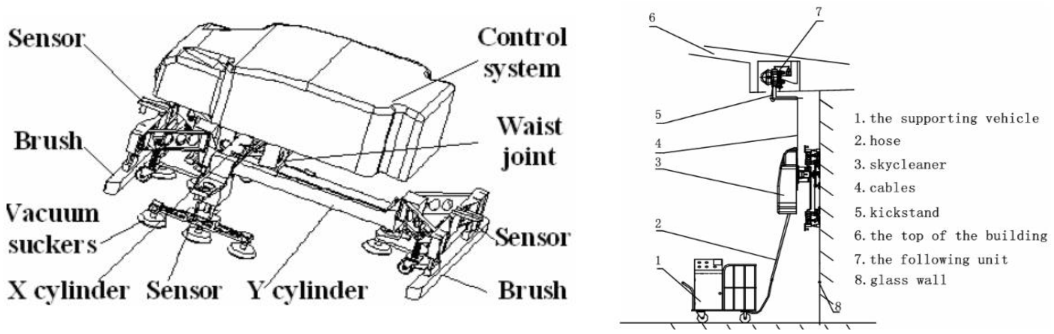


Figure 2.6. The Robotics System of *Sky Cleaner 3 (*H. Zhang, J. Zhang, G. Zong. 2004).

The *Sky Cleaner 3* itself supports a turning waist joint actuated by a pendulum cylinder connects the X and Y cylinders. On opposite ends in the Y direction there are also four brush cylinders, which actuate the brushes up and down. An adaptive cleaning head is designed especially for effective, efficient and safe cleaning, equipped with a drainage collecting device. When the glass is being cleaned, the water is not allowed to drip down; it is firstly drawn off the glass wall through a vacuum pump on the robot. Then the water will flow down because of the gravity and be collected on the supporting vehicle on the ground. At last, the drainage will be filtered, and then reused for cleaning. Some sensors that can detect the window obstacles are mounted on each end of the X and Y cylinders. The robot can both clean and walk on the glass walls automatically in the up-down direction as well as the right-left direction. A PLC is used for the robot control system, which can directly count the pulse signals from the encoder and directly drive the solenoid valves, relays and vacuum ejectors. FX2N-4AD which is added to the system can identify the sensor analogue signals. The control and monitoring of the robot is achieved through the GUI to allow an effective and user-friendly operation of the robot. The communication interface between the PLC and the controller of the following unit is designed to synchronizing the following movement of the cables [3].

In present day, this type of machinery would prove to be inefficient due to its bulky size and complex usage. The need for a supporting vehicle can be eliminated by onboard storage. This might however increase the total weight of the machine, but with advancements in material and current technologies, the overall weight of the machine can be reduced by replacing heavy parts with more light weight material. The control unit of the robot in the form of a PLC can be replaced with another type of MCU which is not only smaller in size, but also can handle more computing power thus allowing better control of the robot. The movement of the robot is also only limited to only one side of the building per cycle.

2.3.2. Adhesion-Aware Facade Cleaning Robot (*Mantis*).

A more recent addition of the window cleaning robot utilising a vacuum adhesion method was made by a team from M. A. Viraj. The proposed method analyses the current flow of the motors attached to the impellers to establish the adhesion-awareness, which is essential for identifying the deficiencies in sealing between the robot and a glass surface that may hinder the safe operation. The decision of the robot such as the initiation of the reconfiguration can be made by accounting the adhesion-awareness for improving the operation of the robot [4].

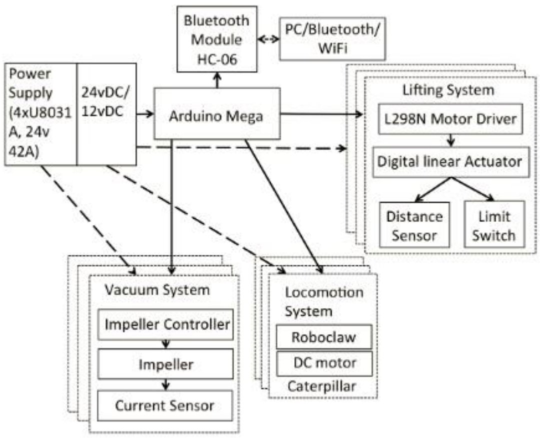


Figure 2.7. *Mantis* Robot Architecture
(M. A. Viraj, J. Muthugala, M. Vega-Heredia, A. Vengadesh, G. Sriharsha, and Mohan Rajesh Elara. 2019).

The proposed method for establishing the adhesion-awareness was implemented on Mantis robot, and experiments have been conducted inside a laboratory environment by exposing the Mantis robot into different conditions. Mantis is a window climbing robot composed of 3 modules interconnected by parallel longitudinal bars of carbon fibre. Each of the modules is composed of a vacuum system using an impeller-based blower. Each of the blowers have a capacity of 8 kPa, which generates vacuum and consequently friction between the microfiber towel adhered to the module and the glass. Mantis can make the transition between one window and another by the avoidance of the frame of the window where the linear actuator separates the pad from the glass for each of the modules respectively, as the modules approach the frame [4].

The locomotion is developed using caterpillar wheels, actuated with DC motors, controlled with Roboclaw velocity controller. The system is controlled using Arduino Mega, and a Bluetooth link is used for teleoperation. The module can lift from the glass using a lead-screw linear actuator, using a stepper motor. During the transition of the robot from one panel to another, the modules are separated from the glass of the window. To know the distance between the module and the glass, Time of Flight (TOF) distance sensors are used. For the heading angle estimation, an Inertial Measurement Unit (IMU) is used. A current sensor is connected to measure the current flow of each of the blower. ACS715 current sensor from Pololu brand, which has an accuracy level of 0.01 A and a maximum rating of 30 A, is used for this purpose. The sampling frequency is 1.2 milliseconds. The power supply to the impeller motors is kept at 24 V constant [4].

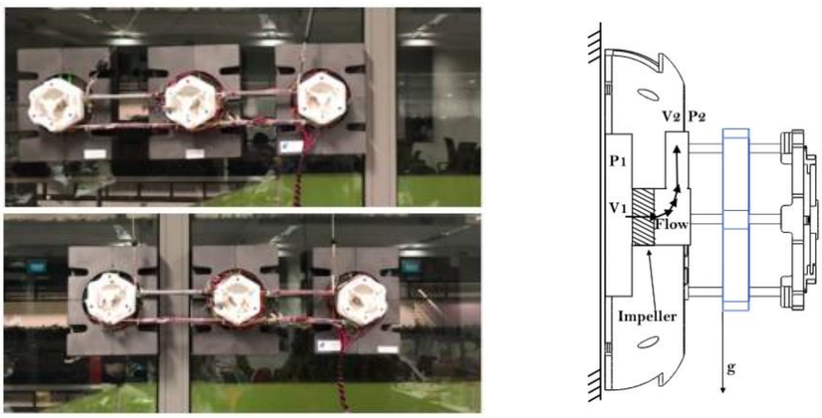


Figure 2.8. *Mantis* Robot Structure (M. A. Viraj, J. Muthugala, M. Vega-Heredia, A. Vengadesh, G. Sriharsha, and Mohan Rajesh Elara. 2019).

The overall robot research explores the transition and movement method on a glass surface, along with power requirements to move the impeller vacuum that keeps the robot attached to the surface. The results states that the current drawn from the impeller of the vacuum can vary depending on the adhesion condition of the suction cup. High current drawn expects the vacuum to be not fully attached to the surface, while a “Fine” area where the current is expected to be the normal rated consumed expects the suction to be fully attached to the glass surface. This adhesion awareness method can be used to further develop window cleaning robots, especially one that is equipped with machine learning to make movement decisions based on the conditions of the robot.

2.3.3. Other Attempts

There have been many attempts in trying to make a device to automate the process of window cleaning. Recently the usage of propellor drones have proven to be quite popular in experimentation. This method uses a four propellor quadcopter modified and equipped with a high-pressure sprayer which will clean the glass [5][9]. However, this type of method is inefficient in many environmental conditions. Being an aerial device, there needs to be some distance between the surface area of the building and the drone itself. This results in inefficient cleaning of the surface since there will be a drop in pressure power of the cleaning mechanism itself. Not to mention the force of the spray can interfere with the stability of the drone. Though using drones for cleaning work has many advantages, it exhibits a major disadvantage being the short endurance. For

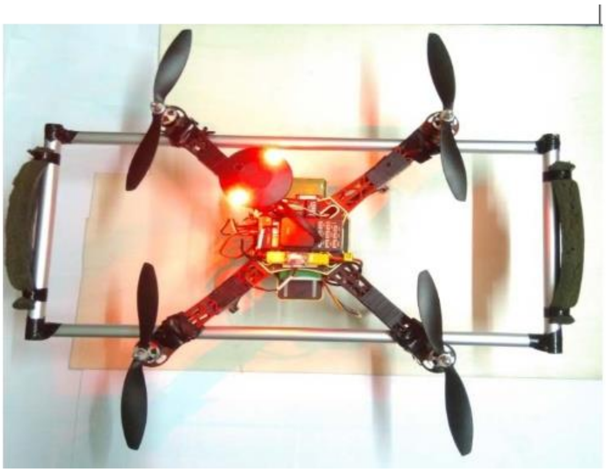


Figure 2.9. UAV Cleaning Device
(Dr. John Dhanaseely.A. 2021).

example, Dhanaseely and Srinivasan reported a drone with the endurance of only 8 minutes [9].

A large number of systems utilises a rooftop hoisting system, much like the traditional method of window cleaning on high-rises. A commercialized product of an automatic cleaning system exists named *IPC Eagle HighRise*. Created by the IPC of Tenant Company, this product consists of two units, which is the Cleaning Unit and the Rooftop Unit. The Rooftop unit serves as an anchor that supports and moves the Cleaning Unit along the surface of a building. The overall system is a type of rigging method that utilises ropes to move a cleaning device. Unfortunately results showed that even when working in full speed, the device is relatively slow as compared to what a machine can do with no major speed advantage over what a human can do (actually in some cases they are even slower than humans). The main reason for the relatively low speed of those robots in performing cleaning services is due to the fact that most of the robots available in the market depend (with some exceptions) on circular or zigzag motion to clean different parts of the windows [7]. A similar device was developed in Egypt with the sole purpose of improving on this system. The results that the system reduced the weight by applying parts of lightweight material. However, the weight such as control panel including the power supply controller and circuit interface devices and actuators that are carried within the cleaning device should be additionally considered [7]. Both these devices utilise a method can be found in most automated window cleaning system in the market. Such approach ensures a high degree of security, as the robot is always connected to the roof by ropes [9]. However, the devices are very bulky, and needs to be move from one side of a building to another. They also require large amount of power to be drawn as they employ a heavy-duty method of cleaning.

Figure 2.10. Robotic Façade Cleaning System (H. Fawzy, H. El Sherif, A. Khamis. 2019).

Another attempt is by creating a device named *Windoro* using a combination of magnetic adhesion and suction. This attempt was created to develop two-sided cleaning device that is portable, lightweight, and efficient. The robot, is moved on the X and Y axis by motor driven wheels. The main advantage of this device is its high mobility and failsafe mechanism from its magnets. However, for outdoor applications such as a high-rise building, it cannot with stand the winds that are present in such environment. Moreover, the mobility of the device in a larger scale is limited due to reliance of the magnetic adhesion, which requires there to be an indoor and outdoor unit to properly attach to the surface [8].

Figure 2.11. *Windoro (*Y. Choi, K. Jung. 2011*).*

2.4. Methods of Calculation

Designing a mechanical system will require calculation, which in this case would be the vacuum system which hold the robot onto the vertical surface, along with the actuators that maneuvers the robot itself. This ranges from basic formulas that factors pressure and the sizing of the motor.

2.4.1. Vacuum Calculation

To calculate the amount of holding force produced from the vacuum pump, Pascal's law can be used to convert the vacuum pressure into the holding force,

$$\boldsymbol{P} = \frac{\boldsymbol{F}}{\boldsymbol{A}} \tag{1}$$

Where,

P = Vacuum Pressure (Pa)

F = Holding Force (N)

A = Vacuum Area (m^2)

How well a given suction cup grips and seals against a workpiece surface is another consideration. Friction coefficient μ approximates the relationship between friction force and normal force. These factors need to take it into consideration when sizing vacuum cups. Typical estimated values for μ from several manufacturers are [16]:

Table 2.2. Friction Coefficient Table.

Surface Type	**Friction Coefficient**
Oily Surface	0.1
Moist or Wet Surface	0.2-0.4
Glass, Stone, Dry Plastic	0.5
Rough Surfaces	0.6

The formula used to describe a vertical suction cup with in conjunction with a vertical load is as follows,

$$\boldsymbol{F_{holding}} = \left(\frac{\boldsymbol{m}}{\boldsymbol{\mu}}\right) \cdot \boldsymbol{g} \tag{2}$$

Where,

$F_{holding}$ = Holding Force (N)

m = Mass of the Load (kg)

μ = Coefficient of Friction.

g = Gravity (ms^{-2})

2.4.2. Actuator Sizing

To find the appropriate specifications of the actuating motors, the required torque needs to be calculated. This can be done mainly through a free-body diagram. Motors that are actuating on the vertical axis won’t require as much torque as the forces acting on it are only bending motions. However, for this development, leadscrews are used to move on the vertical axis. Sizing these lead screws can be done using the following formula [17],

$$\boldsymbol{T} = \frac{\boldsymbol{F_{load}} \cdot \boldsymbol{L}}{\boldsymbol{2} \cdot \boldsymbol{\mu_{thread}} \cdot \boldsymbol{e}} \tag{3}$$

Where,

F_{load} = Loaded Force (N)

L = Lead Length (mm)

μ_{thread} = Thread Interface Friction.

e = Lead Screw Efficiency.

The efficiency of lead screws, on the other hand, typically ranges between 20% and 80%. The efficiency of a lead screw is highly dependent upon its helix angle. As a general rule, higher helix angles mean higher efficiency. A higher helix angle is more efficient because less of the energy used to drive the lead screw goes into overcoming friction. This is because the number of times the screw must rotate to achieve the same linear displacement is reduced on a high helix screw [18].

2.4.3. PID Equation

To create a smooth movement for the robot, a PID function can be utilized to help the sequence during movement. A PID function is advantages as it compares the error of the system with the target that it is trying to achieve. PID uses three constants to calculate the needed control variable that is sent to the actuator. The equation for the PID can be seen below,

$$\boldsymbol{u(t) = K_p e(t) + K_i \int e(t)dt + K_p \frac{de}{dt}} \tag{4}$$

Where,

$u(t)$ = PID Control Variable

K_p = Proportional Gain

$e(t)$ = Error Value

K_i = Integral Gain

de = Change in Error Value

dt = Change in Time

CHAPTER 3 – RESEARCH METHODS

This study is conducted by developing the mechanical, electrical, and programing aspects of the robot, with the main focus being on the base development of the prototype. Due to this being the first study of the topic to be conducted from the institution, many of the methods are considered to be approximations. The general plan of this study can be seen in the flow chart below,

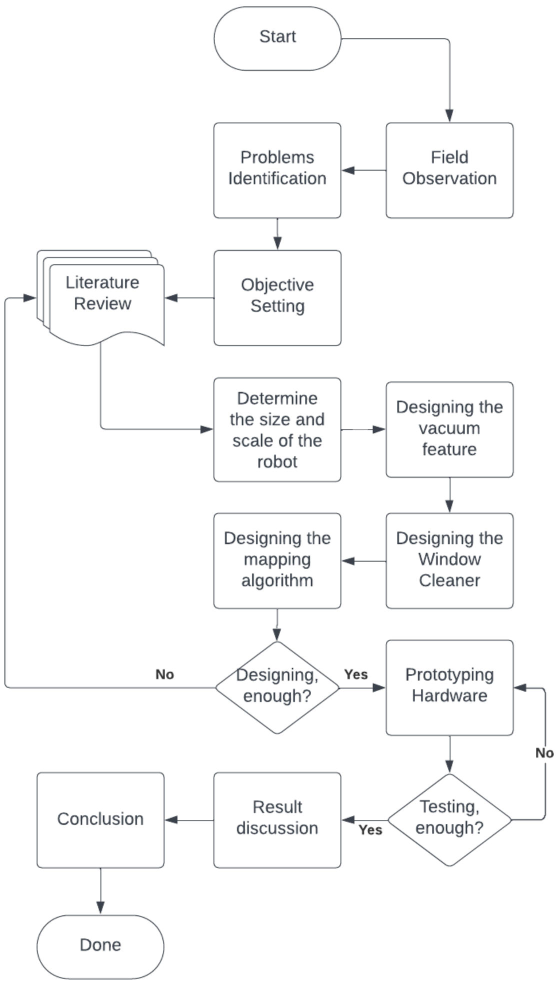


Figure 2.1. Flow Process of Study.

The process of prototyping the hardware will start with the overall mechanical design through CADs and sketches. Revisions of the mechanical design are done after the study has been conducted, or if it hinders the objectives of the study. Electrical design and prototyping are done in conjunction with the mechanical design, with it being integrated after the mechanical design has been completed. The control algorithm is created after the electrical components have been fully, or somewhat integrated with the mechanical system to allow physical testing of the program itself.

This study will focus on four systems of the robot, which is the vacuum adhesion system, the modular limb system, the main body frame, and overall control algorithm. Each system is to be tested separately first to record and measure the performance of each of them, and conclude if a redesign is required in the future development.

3.1. Vacuum Adhesion System

The vacuum adhesion system is used to attach the robot onto the surface of a building, which in typical application, would be glass surfaces. The vacuum adhesion system was chosen because it is more reliable than other methods of adhesion, due to the higher holding force produced by vacuum pumps. This system comprises of a custom-made suction cup, a small sized vacuum pump, and a controller to control the pump itself.

3.1.1 Suction Cup

The suction cup is designed to withstand the weight of the whole robot. The material for the suction cup would ideally be a polycarbonate plastic which is machined per the model, however as this is a developmental prototype, PLA would prove sufficient in handling the loads of the system. The suction cup is fitted with a joint made from aluminum sheet metal which holds a bearing house. The bearing house holds a shaft in place, allowing the suction cup to move freely around a 260° angle. This is done to accommodate different angled surfaces which are present on high-rise buildings. The sheet metal is held onto the suction cup using fasteners and seals to prevent any leakages that might affect the performance of the holding force. The outer rim uses a silicone seal to achieve adhesion on the vertical surface. Suction force is transferred from the vacuum pump to the suction cup through a 6mm silicone tubing attached to the pneumatic fitting of the suction cup.

Figure 2.3. Silicone Seal and Pneumatic Fitting.

To produce a suction force for the suction cup, a 12VDC suction pump is used. The pump itself is a diaphragm style pump rotated by a 12VDC motor which is rated to achieve a vacuum pressure of 75kPa.

Figure 2.2. Vacuum Pump.

3.1.2 Vacuum Sizing

The custom suction cup designed has a suction chamber with a surface area of around 15713.92mm^2, which is evaluated using the SolidWorks Evaluation tool. The rated vacuum pressure that is able to be produced by the vacuum pump is written to be 75kPa. By using formula (1), the theoretical holding force produced by the suction cup can be calculated as follows,

$$F_{holding} = P.A_{surface} \tag{1}$$

$$F_{holding} = 75000\,.\,0.0157\,.\,Nm^{-2}.m^2$$

$$F_{holding} = 1177.5\ N$$

The resulting theoretical force is around 1177.5 N. From that, the theoretical maximum weight that a single limb can handle can be calculated based on formula (2) while factoring the coefficient of friction of a glass surface as well as the general safety factor,

$$m_{max} = \frac{F_{holding} \cdot \mu_{glass}}{g \cdot S_{safety}} \qquad (2)$$

$$m_{max} = \frac{1177.5 \,.\, 0.5}{9.81 \,.\; 1.5} \, \frac{N}{ms^{-2}}$$

$$m_{max} = 40.010 \text{ kg}$$

3.1.3 Electrical Control

The adhesion system is fitted with various electrical components which controls the mechanism. The two main components of this system are the 5VDC Relay which controls the activation of the suction, and an INA226 current sensor to detect the current of the robot. The usage of the INA226 is to replace that of a pressure or vacuum sensor which detects whether successful adhesion was achieved. Pressure sensors are costly in the market, moreover, it would be difficult to integrate it into the system. An alternative approach to this is by using the load characteristics of a motor under load, where it will increase the amount of current which is feed into the motor. The sensor itself is able to be integrated using an I2C communication protocol, allowing quick and easy data reading from the sensor.

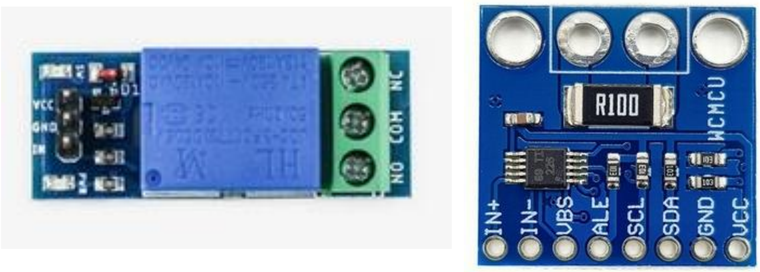


Figure 2.4. Relay and Current Sensor.

The 5VDC relay is a simple solenoid gate, where in this application is activated when a LOW signal is given into the input pin of the relay. It is used due to it simplicity, and

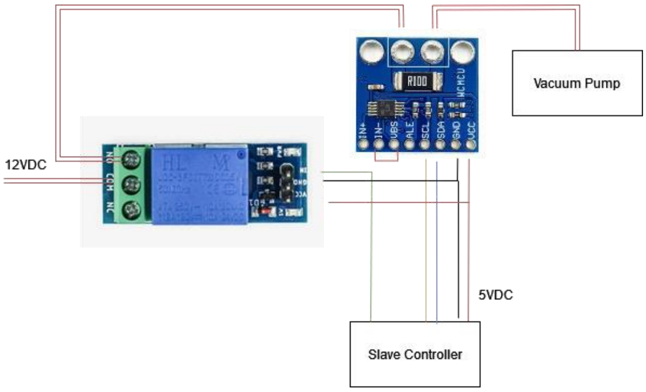


Figure 2.5. Suction Electrical Control Wiring Diagram.

because it can handle higher ratings which in this case, up to 24VDC and 10A of current. Both the INA226 and relay are linked with a slave controller, which is an Arduino Nano 328p (which will be explained later on.). The electrical power will first run into the relay to avoid misreading the current, and the transferred to the INA226 to allow it to read the driving current.

3.2 Modular Limbs

The modular limbs are comprised of three segments, each one able to control the angle around 90°. The rotating action between these segments are done using motors which is controlled by the slave controller through a motor driver. To read the position of the limbs itself, an electric encoder is used.

3.2.1 Motor Sizing

The motors used to rotate the segments of the limb can be calculated based on the load torque applied. Since the application of the motor is on the limbs, where the shaft is positioned vertically, the load torque that affects the motor will be due to the bending moment which will be assumed to be very small. However, the motor used to rotate the frame segment of the system can be calculated by first analysing the free body diagram below,

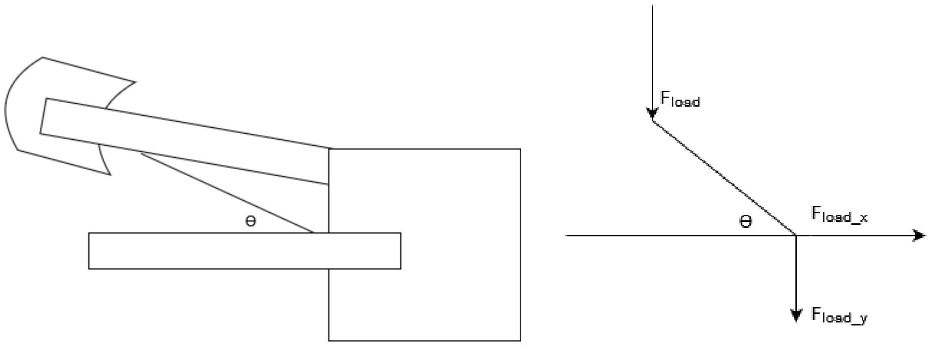


Figure 2.6. Free Body Diagram Analysis.

By using formula (3) the load torque can be calculated by applying a safety factor of 1.5, a minimum angle of 33.1 degrees (from SolidWorks Evaluation) for θ, and limiting that the maximum weight of each limb to be 1,5kg,

$$T_{min} = \frac{F_{load\_x}.L}{2.\mu_{thread}.e}.n_{safety} \tag{2}$$

$$T_{min} = \frac{1.5 \times 9.81 .\times \cos 33.1 \times 2}{2 \times 0.15 \times .0.52} \times 1.5$$

$$T_{min} = 2.371\ Ncm^2$$

$$T_{min} = 0.242\ kgcm^2$$

3.2.2 Mechanical System

The limb segments are driven using a JGY370 (10RPM) which is a 12VDC worm gear motor able to produce 15.9 $kgcm^2$ of continuous torque. Worm gear motors are advantageous in this application as it can stop the passive movements of the segments when it is currently on idle. In addition to its small size, it can also handle a larger amount of torque to move the segments. The shafts of these motors are attached to the segments using a rigid flange made out of stainless steel, which is able to transfer the torque quite effectively.

To move the limb modules to up and down, a drive screw is used to move two modules simultaneously. The driving screw is connected to the limbs using custom made tie rods. The driving screw itself is driven using another JGY370 (30RPM) with 5.3kgcm^2 of continuous torque, however at a different rotational speed.

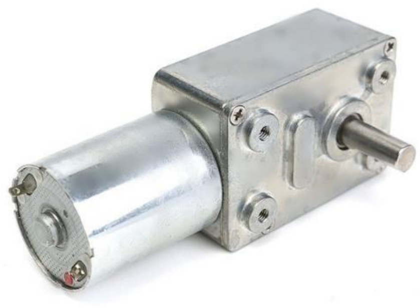


Figure 2.7. JGY370 DC Motor.

3.2.3 Electrical Control

The limb module is mainly controlled using a slave microcontroller, in this case is the Arduino Nano which uses an Atmega328p processor. The processing power of the Arduino Nano is sufficient for a slave application of the limb module, however might be limiting for future developments. The Arduino Nano is also suitable for low power application, where if the robot uses batteries, is a more suitable controller.

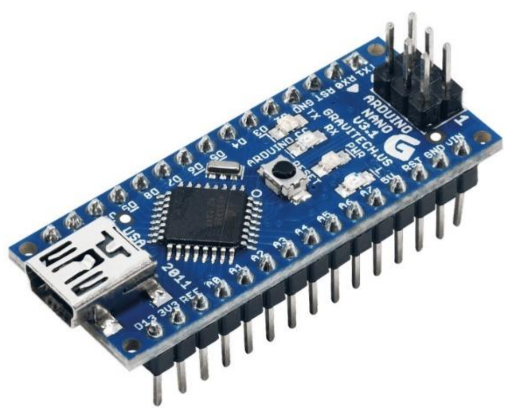


Figure 2.8. Arduino Nano.

To drive the motors, the L298N motor driver module is used to control the speed and direction. The driver is a dual channel driver which means it is able to control two motors simultaneously, which means only one driver is need to move the segments. It is capable of an input voltage of 12VDC with the maximum input current being at around 3A. The L298N is connected to the PWM pins of the Arduino Nano to control the speed, while the source voltage is supplied with 12VDC.

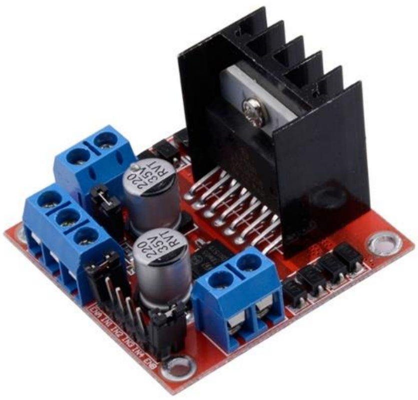


Figure 2.9. Motor Driver L298N.

Traditionally, a rotary encoder is used to measure the position of a shaft. This application does not use a rotary encoder in a traditional way, but rather uses a magnetic encoder that is able to measure the position of a shaft directly instead of using pulses. The magnetic encoder is the AS5600, which uses a magnet on the end of the shaft to read the position in degrees. By using this instead of the traditional encoder, it can negate the need to zero all the segments when starting any sequence. The encoder housing itself is custom made to ensure easy integration and so that the shaft properly aligned with the encoder. The encoder is integrated using I2C, however, the address of the module is fixed and cannot be changed. To circumvent this problem, an I2C multiplexer is used to connect two AS5600 modules into one microcontroller. It is capable of up to 4 channels, and is able to change its address. The multiplexer used in this application is the TCA9548, which is connected to the Arduino Nano.

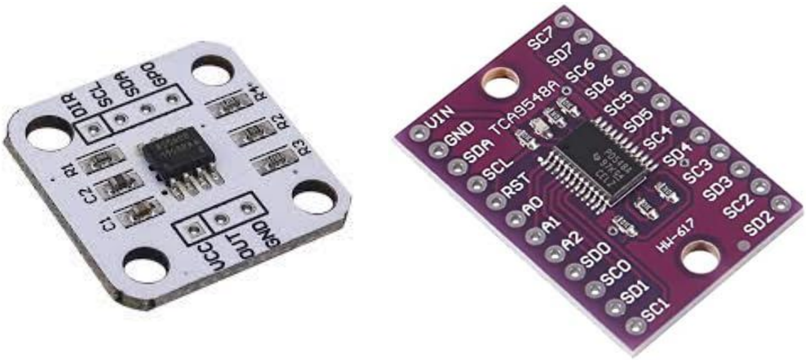


Figure 2.10. Magnetic Encoder and I2C Multiplexer.

Each electrical module except the AS5600 is soldered on to a PCB matrix board, with jumper wires connecting the required pins of the modules. A custom printed 2-layer PCB would be more ideal to minimize electrical noise from the signal transmission. However, for an early development of the system, a matrix board will be sufficient. The wiring of the controls is as below,

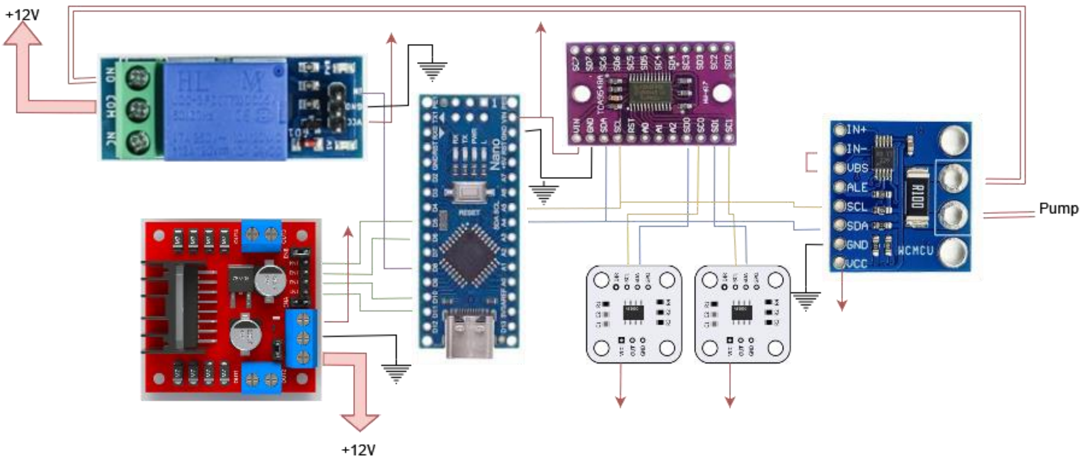


Figure 2.11. Slave Control Wiring.

3.3 Cleaning Module

The cleaning module of the robot is a simple system using a spraying nozzle, a servo, and a liquid pump. The spraying nozzle is a quick connect nozzle that is capable of spraying at a 40° angle. The nozzle itself is attached to a servo motor to control the direction of the water spray. The servo, which is a MG995, is connected to the main controller of the robot, mounted on the main frame.

The pump used is a diaphragm pump powered by a VDC Motor. It can be fed 6-12VDC, and capable of outputting a flow rate of up to 2L/min. Its small size makes it easy to integrate, and the flow rate is sufficient to output a spraying action on the nozzle. The pump itself is activated using a 5VDC Relay which is connected to the MCU.

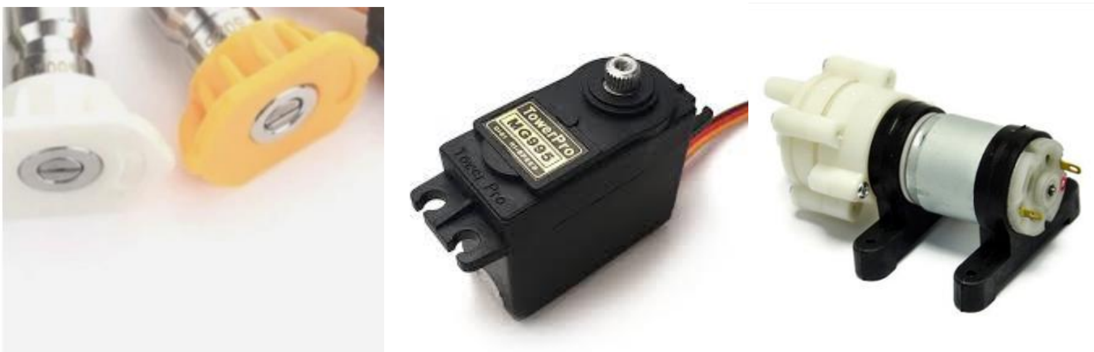


Figure 2.12. Nozzle, Servo, and Pump.

Each module is connected to the main controller of the robot, with the input voltage being separate (only sharing a common ground). This is due to the power of the modules needing around 5V, while the main controller can only handle 3.3V. The connections between these modules are as the diagram below,

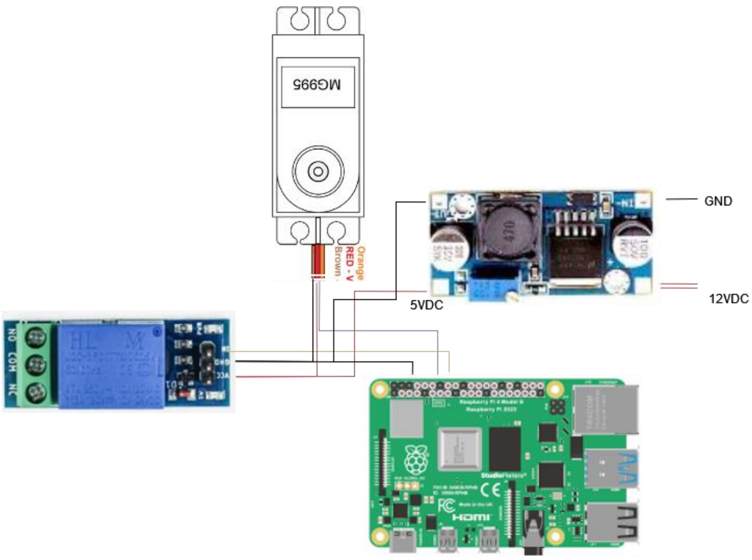


Figure 2.13. Cleaning Module Wiring Diagram.

3.4 Main Body Frame

The main body frame is made out of 3mm sheet aluminum to ensure rigidity and strength since it will handle most of the weight of the robot. KFL08 bearings are used to mount the shaft connecting the limb modules, allowing a free moving motion. The limb modules are also attached to tie rods connected to a T8 leadscrew rotate them about the shaft. The leadscrews are mounted using KP08 pillow blocks which is further connected through a shaft coupling to the previously mentioned JGY370. The cleaning module is attached to the top part of the frame. The main body also contains a terminal block to tidy the connections of between the system, mainly power lines. It also contains DC-DC buck converter (L2596) which converts 12VDC from the source line to power 3.3-5V electrical components.

3.5 Control Algorithm

There are two separate controllers that are used in this system. The slave controller which uses an Arduino Nano 328p and the Main Controlling Unit (MCU) which is a Raspberry Pi 4B with 4GB of RAM. The way that the system works is that the Raspberry Pi will send a command signal that the slave controller will actuate. Since the limbs are modular, the slave controllers can be flashed with the same program as each other.

3.5.1 Slave Controller (Arduino Nano 328p)

The slave controller is connected to other modules such as a relay, encoders, and current sensor. There are libraries to allow easier integration such as the INA226 library, AS5600 library, and the PID library which supports a PID function. The relay is controlled using a digital output, with a LOW signal activating the relay of the pump.

The way the program works is that it will start off by rotating all the segments to the beginning position. After that, it will wait for an input command from the MCU. After an input has been sent and successfully received on the slave, it will translate the command into a readable variable, which in this case is the angle need to be rotated on a specific segment and the command to turn the adhesion on or off. The angle variable is then inputted into a PID function which will drive the motor while accelerating it accordingly. After it has been inputted into the function, the vacuum pump will be turned off if deemed by the command by deactivating the relay making the suction cup release. The motors will then move the limbs based on the PID function. If the limbs haven't reached their position in a certain amount of time, it will send back a warning to the MCU to change the position. After the limbs have reached their target position, the suction will begin if it was previously off. If the adhesion is fails, it will also send a warning message to the MCU. If every function succeeds, the slave will send a success command to the MCU, and wait for a new command to be inputted.

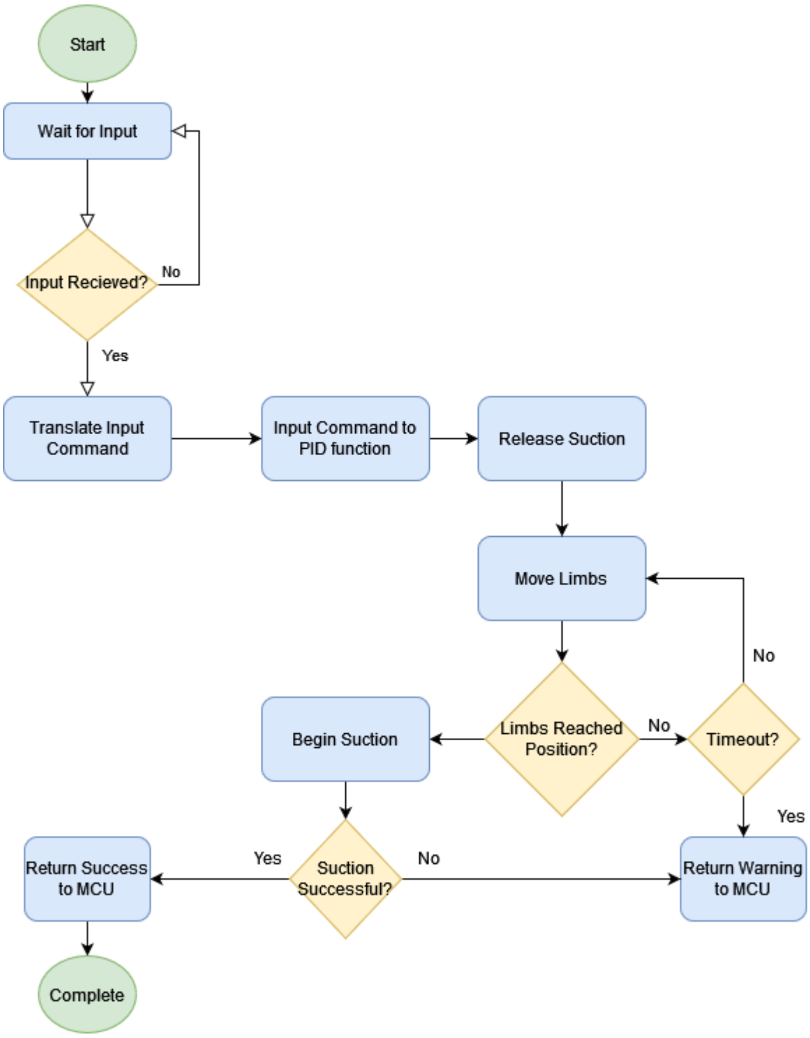


Figure 2.14. Slave Controller Program Flow.

In this design, the PID function is tuned manually where the constants K_p and K_i is the main constants to be tuned. This is done through the plotting of the graph based on the encoder value and determined to sufficient if the system is stable. This can be further optimized to increase the speed of the cycle time, but for the development phase, a stable system is the main goal.

3.5.2 Main Controller (Raspberry Pi 4 B)

The Main Controller Unit (MCU) of the robot utilises a Raspberry Pi 4 which is a single board computer which is typically used in robotic controls. The benefit of using the Raspberry Pi 4 is due to it being capable to run the Robot Operating System (ROS) which is capable of controlling the four-limb system of this robot. Unlike mini-PCs, this single board computer is also more power efficient, making it ideal for mobile robots.

The board runs Ubuntu Server x64 which enables it to run ROS. However, for prototyping and communication testing, Raspbian can be used to develop an initial control algorithm. The Raspberry Pi 4 communicates to the slave Arduino Nano through serial communication. The board is fitted with 4 serial ports, but a USB hub is added for additional ports, which is usually needed for external controls such as a mouse or keyboard to configure the software on the board. Scripts on the board can be executed

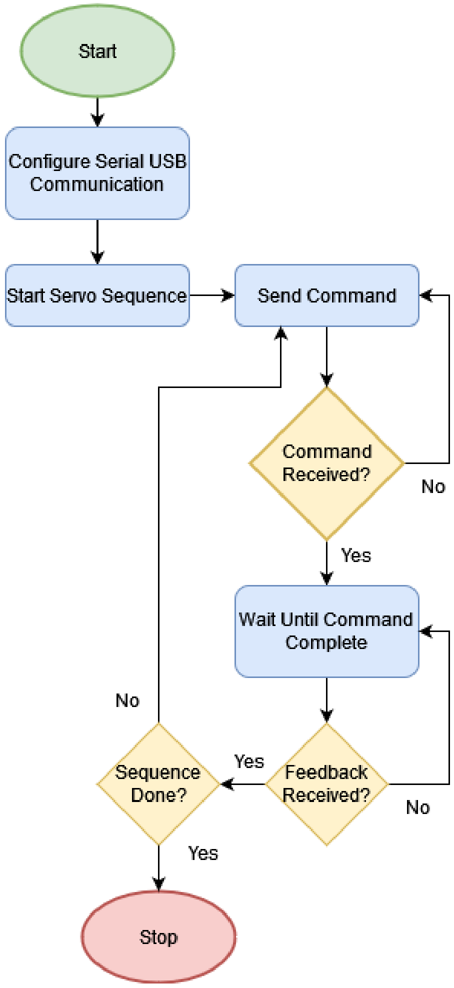


Figure 2.15. Raspberry Pi 4 Algorithm Framework.

to sequence the limb systems simultaneously. Wireless control is also done by enabling SSH on the board, which is further connected to a laptop or other computer through RealVNC.

A function is created to move the limb systems from the MCU to the Arduino Nano. First, a serial communication is opened on MCU. After all initialization is complete, a command is sent with the format “[XX] [YY] [ZZ]” where “XX” is the target angle for the middle segment motor, “YY” is the target for the motor of the frame, and “ZZ” is the command to activate or deactivate the pump. If the command is successfully received by the slave, it will send feedback to the Raspberry Pi 4. If there is no feedback, then it will continue to attempt to send the command. If feedback is received in the Raspberry Pi 4, it will wait until another feedback from the slave is received telling it that the command was successfully executed, and the limbs have reached the target angle.

3.6 Analytical Testing

The various techniques of testing the performance of the robot will be done through tests such as the limb adhesion, current sensor, movement, and cleaning. Since this is a developmental prototype, most of these tests are done in a safe simulated environment (indoors) instead of a real environment (vertical building surface) to ensure safety to all parties involved.

3.6.1 Vacuum Adhesion Test

The vacuum adhesion test is a test conducted to find and ensure that the breaking point or detaching point of the suction is high enough so that it does not fail under the weight of the robot. The overall test will be conducted using a crane scale with the suction cup being attached to vertical surface similar to that of a glass surface. A weight will also be attached to the bottom of the weight scale which will be increased slowly over time. The weight in which the suction fails will be the maximum amount of weight that the designed suction cup can handle.

3.6.2 Current Sensor Test

The current sensor is tested to find the point in which a successful adhesion can read. Since the method of detection is by using the load characteristic of a DC motor, the current sensor should be able to read the rise in current which is being fed into the vacuum pump. The test will be conducted with the vacuum system, where the current reads the data continuously every 20ms. The suction cup will be placed on a surface which will create rise in current. The maximum current that the INA226 detects will be the setpoint that is used to determine successful adhesion.

3.6.3 Movement Test

To measure the performance of the limb system, a test is done by sequencing the limb systems individually and simultaneously. Before combining the systems into one, each limb is tested to measure the limitations, cycle time, and travel that it can achieve. The consistency can be defined as the average angle that the limb can move. The cycle time is the time it takes to achieve a target angle. Travel is the distance that the limb is able to move the robot in a single sequence.

After the individual assessment on each of the limbs has been conducted, all of the limb systems can be joined together into a single system. All of the systems will be connected to the MCU which will run a simple sequence algorithm to move the robot. The resulting sequence that is executed by the whole system will be measure by the travel speed that it can achieve, as well as the cycle time to complete a singular sequence. Moreover, the overall current consumption is also measured to assess the performance and power rating of the robot.

3.6.4 Cleaning Test

The cleaning module is to be tested on how effective it is on removing contaminants on a glass surface. This test is done with the robot being attached on a vertical surface, preferably to that similar to that of a glass surface of a high-rise building. The surface area above the robot is to be contaminated with dust, dirt, or other contaminants that might be present on high-rise buildings. The cleaning module is then sequenced to clean the contaminants, and after the sequencing is done, the robot is detached from the surface. The surface is examined and compared between before the cleaning and after the cleaning. The effectiveness is determined by the surface area that it is able to clean and how clean it is.

CHAPTER 4 – RESULTS AND DISCUSSIONS

In this chapter, the result of our tests on the prototype for the autonomous skyscraper cleaning robot will be discussed. We'll take a closer look at how each of the four key systems performed: the vacuum adhesion system, the modular limb system, the cleaning module, and the control algorithm. The goal is to see how well the prototype meets our expectations and assess whether it's ready for real-world use.

4.1. Initial Physical Evaluation

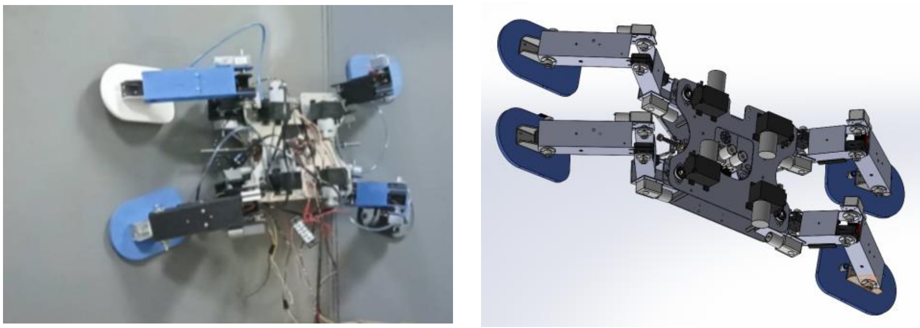


Figure 4.1. Physical Robot.

The resulting robot is equipped with all the planned systems (vacuum system, limb system, cleaning module). Due to time constrains the cable and MCU is not attached properly on the board. However, the overall system is able to execute commands from the MCU, individually and simultaneously. The overall weight of the robot is measured to be around 7.694kg.

Each segment of the limb systems is made from PLA, which proves to be able to handle the weight of the overall robot. However, it needs to be noted that the initial segment length of the arms (particularly the second segment) needed to be shortened, due to the material not being able to handle the bending and twisting moment. The segments are further reinforced using PLA supports, which proves to negate most of the bending and twisting moment on the segments. Each segment is weighted to be around 907g.

4.2. Vacuum Adhesion Test

Examining the outcomes of the vacuum adhesion system tests will determine whether the theoretical calculation is consistent with that of the real-world application. The main focus of the adhesion test is on its holding capacity and reliability, mainly by overloading the system and seeing the releasing point. The reliability is seen through the rate of success in which the vacuum can adhere to a surface.

4.2.1. Releasing Point

The releasing point of the suction cup is determined when the adhesion fails. This is recorded using a crane scale to measure the weight of a simulated load being placed on the suction cup. The resulting points of failure are as follows,

Table 4.1. Releasing Point Data.

Attempt	Weight (kg)
1	11.54
2	11.27
3	11.67
4	11.39
5	11.28
6	11.70
7	11.74
8	11.29
9	11.68
10	11.54
11	11.83
12	11.39
13	11.82
14	11.68
15	11.71

The data from the table above can be translated into the following graph below to determine the safe area in which the suction cup is able to operate. The average releasing point of the 11.569kg, and by applying a safety factor of 1.5, the recommended maximum weight that a single suction cup can handle is 7.731kg.

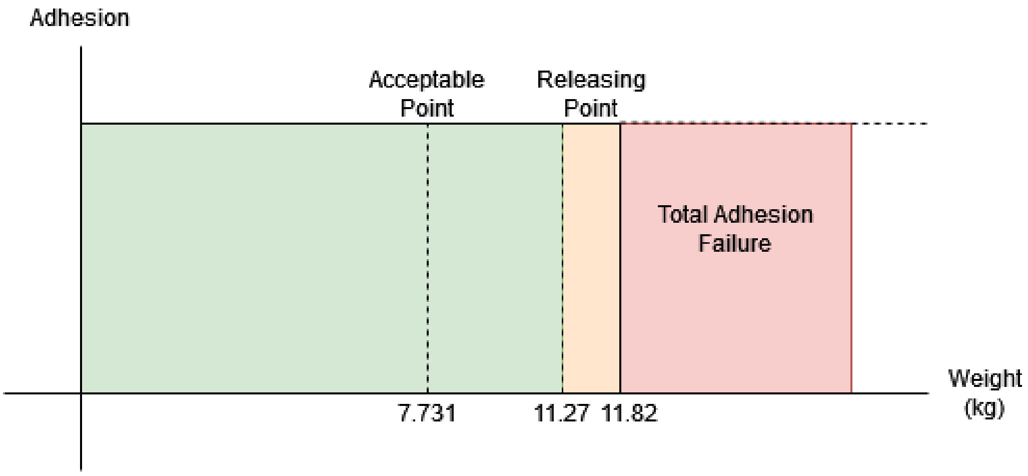


Figure 4.2. Releasing Point Graph.

This graph highlights the critical weight threshold, allowing us to ensure the robot's stability and adhesion during operation.

Upon comparing the theoretical releasing point value with the actual experimental data, we observe a significant discrepancy. The theoretical calculations predict a certain releasing point, yet our experiments show a 71.017% error from this predicted value. This discrepancy emphasizes the importance of empirical testing, as real-world conditions often introduce variables that theoretical models may not fully capture.

There are many factors that might affect the performance of the suction cup adhesion, the main one factor being air leakages. Leakages can come from various factors, with the primary suspect being the material used for the suction cup. Since the suction cup itself is made from 3D printing PLA, it may result in various leaking spots. The material itself is already porous, but in conjunction with the fabrication process, this may result in the performance having a severe drop in the suctions adhesion.

Figure 4.3. Adhesion Test.

Another factor that was spotted during this experiment is the release of the adhesion when the joint between the sheet metal and suction cup is bending. Due to the suction cup being attached with fasteners and seals, the effect of bending when the joint is under load causes a tiny gap to open between the fasteners and suction cup. This was able to be mitigated by tightening the fasteners more.

4.2.2. Success Rate

The success rate of the adhesion is calculated by placing the robot vertically, then vacuum systems are moved into a position that allows it to adhere to a surface. The success of the adhesion is determined when the suction cup sinks into place and is fixed to the position. The amount of success is calculated out of 50 attempts and the success percentage can be seen below,

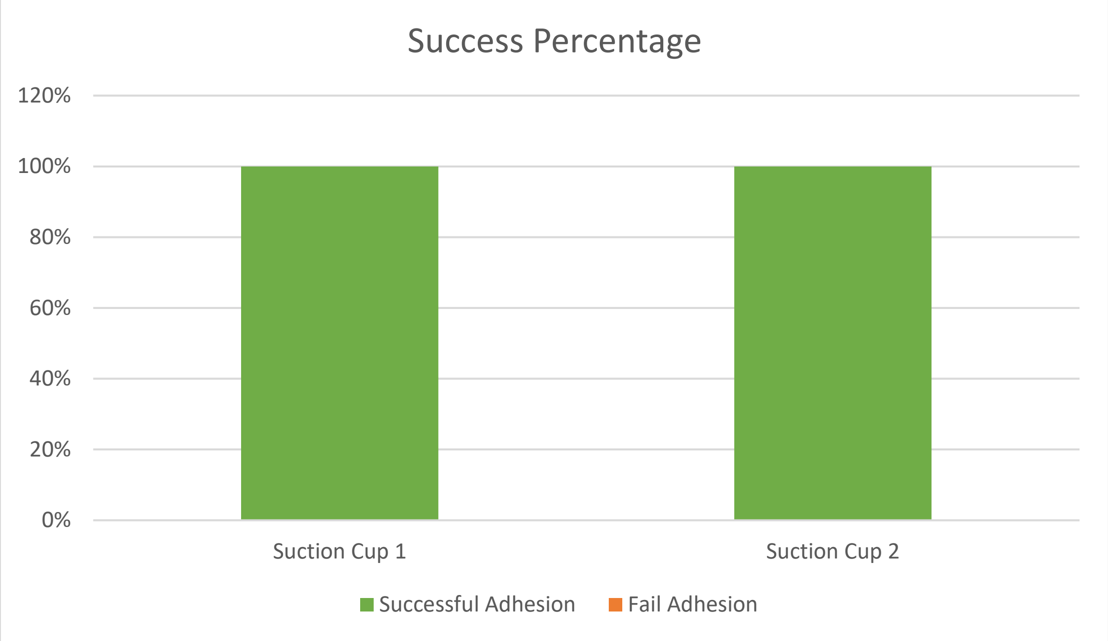


Figure 4.4. Success Percentage Graph.

From the graph, each suction cup has a 100% success rate, which means that the suction system of the robot is very reliable and can stick to a surface. It needs to be noted this success rate is done on a sheet metal surface with holes or pores. The angle of the surface can vary horizontally since the built of the joint is freely moving helping it to align with the surface.

4.2.3. Adhesion Point on Sensor

To determine the adhesion of the current sensor which is attached to the vacuum pump, a simple program is used on the slave controller. The program used will simply read the value that is given from the sensor, which is the converted into a readable value. The readable value is then printed on the serial monitor of the computer, which can be automatically plotted into a graph using the serial plotter.

There are three states that is measured for determining the adhesion point. The first state is the no load state where the suction cup is freely floating with no obstruction. The second state is the under-load state where the suction cup is fully adhered to a surface, resulting in the motor drawing more current from the source. The third state is adhering state where the suction is starting to attach itself to a surface. It needs to be noted that the value produced by the sensor is very different with the real current measured. This is due to the nature of the sensor itself, which needs to be properly tuned to match the real value. Ideally, the measured current has to be the same with the real current, but since the value is still proportional with the real drawn current, it can be disregarded to find the adhesion point. The adhesion point from the scaled value of the measured variable proves to be sufficient on determining the adhesion of the suction cup.

On the first state, the measured real current drawn by the vacuum is on average 642mA. The resulting values are plotted in the graph below,

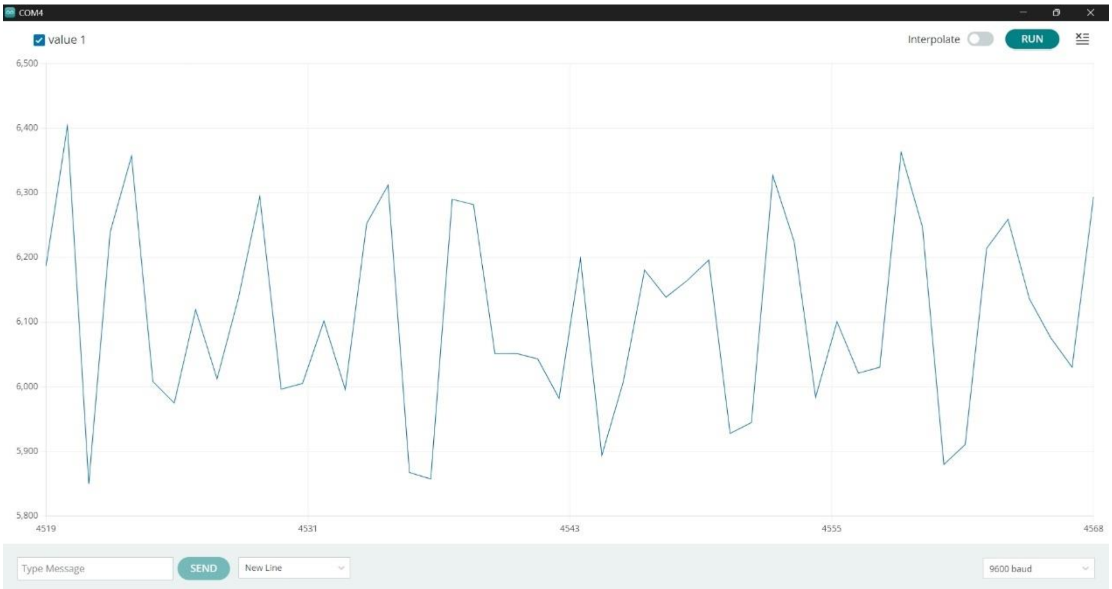


Figure 4.5. No-adhesion Sensor Data.

From the graph, it can be seen that the highest recorded value is 6400, and the lowest recorded value being 5700. The average value that is produced in the graph is approximately 6050.

The second state shows the value of the suction cup adhering to a surface which can be seen below,

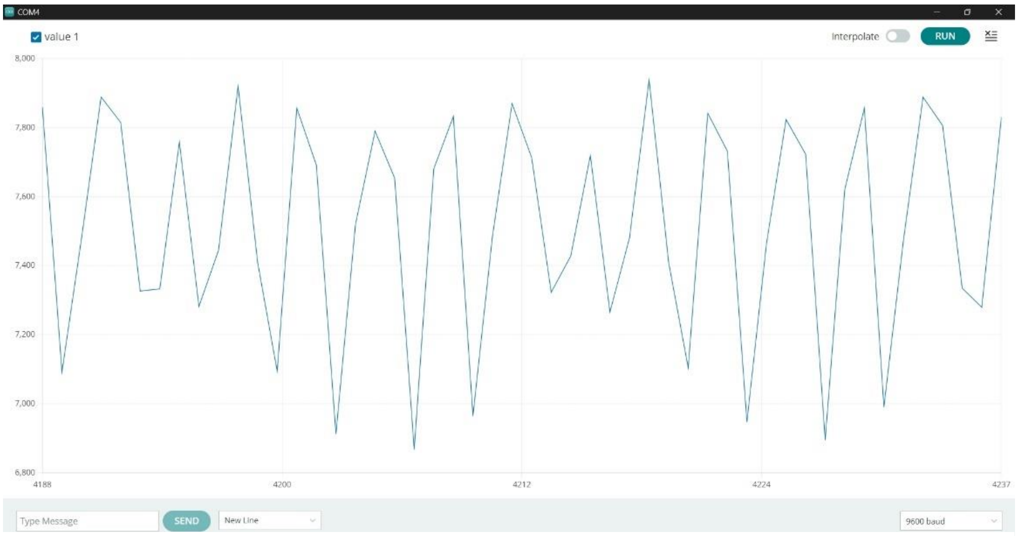


Figure 4.6. Adhesion Sensor Data.

The data shows that the highest recorded value is around 8900, with the lowest being 6850. The average point taken from the graph is around 7875. To determine the point in which the suction cup has adhered, the highest no-load value and lowest under-load value is average which gives the value of 6625. The produced value is inputted into the slave algorithm, which determines the success of the adhesion on each limb.

To further see the point of adhesion, the data from the third state can be seen below,

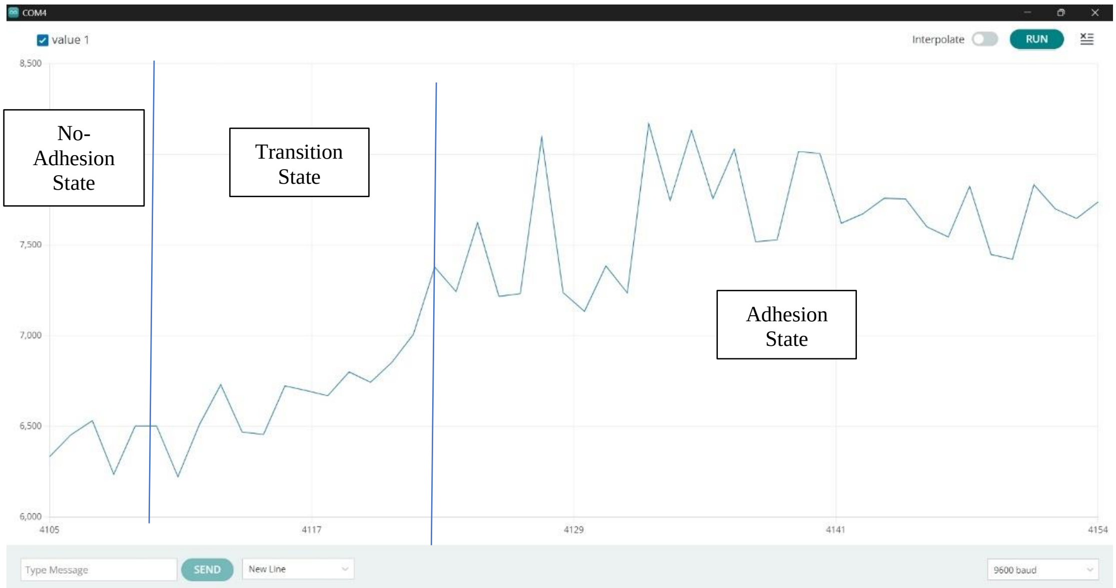


Figure 4.7. Transition Sensor Data.

4.3. Limb System Test

The limb system that is fabricated is created based on a first successful prototype to ensure that the duplicated limbs are easily integrated. The first successful prototype was

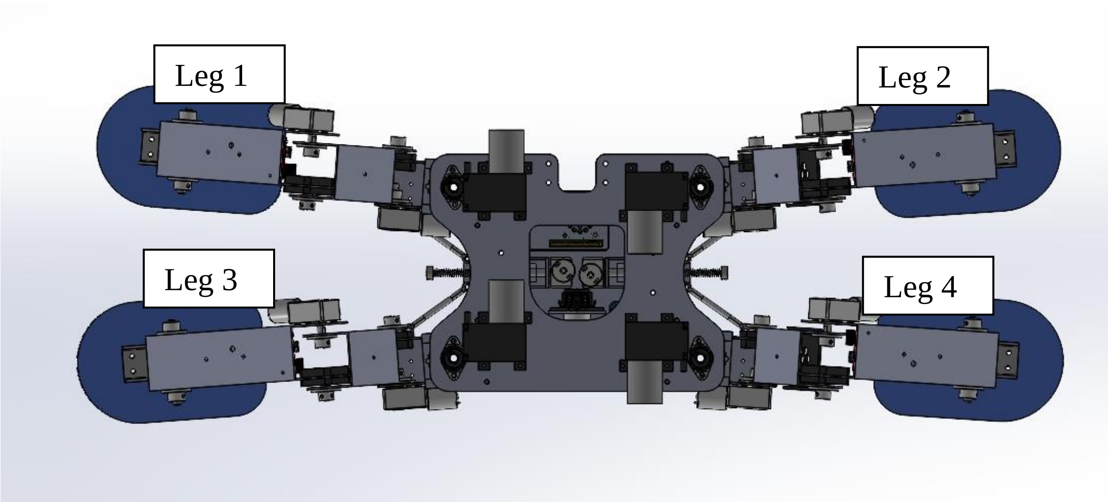


Figure 4.8. Limb System Numbering.

built, and was determined to be successful by using a debugging code created to ensure all of the modules on the system is operational. This debugging process firstly involves ensuring all connection between the modules has been established. The modules are then tested by reading or writing to them, such as the sensors, actuators, and relay. The following test are conducted on the limb systems that are duplicated from said successful prototype. The individual and simultaneous tests are done with the limb systems attached to the main body frame. The robot itself is placed on a small cart on the floor to allow it freely move. This is done due to limbs not being able to handle the weight of the main body frame when it is on the horizontal position (laid down). The robot itself can handle the weight when vertically attached to the wall. Since there are four limb systems, the numbering of each system can be seen below,

4.3.1. Individual System Testing

The individual system testing is done on the four limbs fully attached to the main body frame. This testing mainly involves measuring the angle and movement of the limb systems. The rotation angle is measured by the encoder of the robot, to see whether all of the systems give a consistent value. The travel distance of the robot is the distance on which a singular arm can give based on a particular cycle. The current consumption of the limb system is the measured drawn current of a singular system when every component is activated.

4.3.1.1. Rotation Angle

To see the angle that is rotated by the limb modules, the limits of each segment need to be determined. From some testing, the resulting measured angles that is achievable to enable the robot to move is on the following table,

Table 4.2. Middle Encoder Limits.

	Closest (Degrees)	**Farthest (Degrees)**	**Rotation (Degrees)**
Leg 1	335	315	20
Leg 2	283	312	-29
Leg 3	276	312	-36
Leg 4	319	281	38

The angle data above is the limits for the middle segment of the limbs. The closest and farthest angle is defined as the distance relative to the robot itself to the edge of the segment. The negative value from the *Leg 2* and *Leg 3* is due to it being mirrored from *Leg 1* and *Leg 2,* making the values of the degrees flipped. Overall, there is some deviation on the rotation range of each leg, with the largest deviation around 18 degrees between *Leg 1* and *Leg 4*.

Table 4.3. Frame Encoder Limits.

	Closest (Degrees)	**Farthest (Degrees)**	**Rotation (Degrees)**
Leg 1	295	308	-13
Leg 2	263	245	18
Leg 3	265	247	18
Leg 4	268	289	-21

The angle data above is the limits for the frame segment of the limbs. Again, the negative value from the *Leg 1* and *Leg 4* is due to it being mirrored from *Leg 2* and *Leg 3*, making the values of the degrees flipped. Unlike the middle segment of the limb, the frame segment of the limb results in better values due to the deviation being smaller. The largest deviation is around 8 degrees between *Leg 1* and *Leg 4*, with the values being *Leg 2* and *Leg 3* being consistent.

The main factors that result in the deviation in the test above is the testing environment and method. Due to the fact that the testing is done horizontally on a small trolley, instead of vertically on a wall, the tilt affects the angle for the robot to complete a moving sequence. Other than the testing method, the build of the robot also affects the rotation of the segments. The build of the joint between segments utilizes rigid flanges which are fastened on the shaft using set screws. Due to this, there is a large backlash present in system when rotating each segment. Along with that, the robot is vertically attached to a wall, the only holding force that holds the segments in place is the set screws between the motor shaft and the rigid flange.

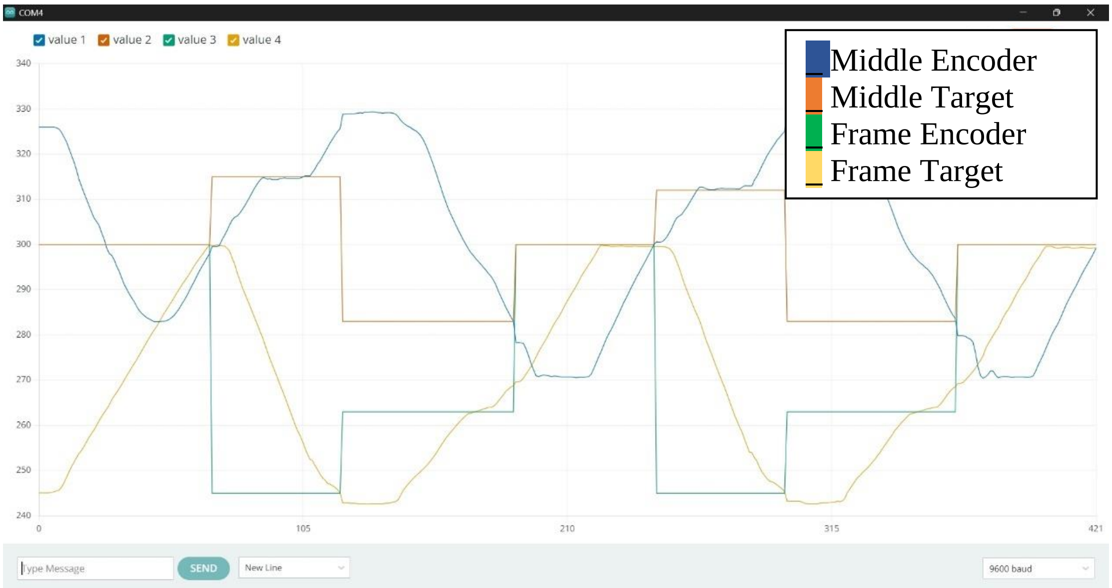


Figure 4.10. Leg 2 Cycle Graph.

The rotation can be further analyzed through the graph above which is the response of the motor (recorded by the encoder) after being run through the PID function. It can be

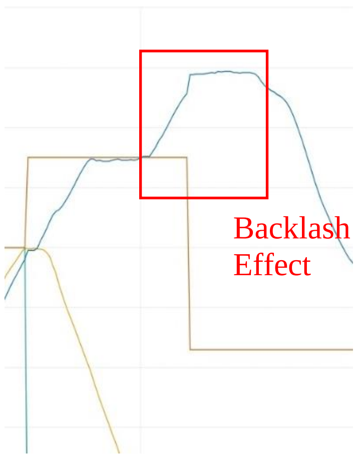


Figure 4.9. Backlash Graph.

seen that the motors rotate accordingly with the target angle being changed. However, there is a segment on the graph that indicates the backlash that was previously mentioned. The backlash is shown below when the frame segment reaches a certain angle,

4.3.1.2. Travel Distance

The achievable travel distance that a singular limb can travel is measured using the closest and farthest point of the rotation angle. The motion that is done by moving between these two points can be defines as pulling and pushing, with pulling being the motion of going from the farthest to the closest point, and pushing being the motion of going from the closest point to the farthest point. This test is conducted by sequencing the pulling motion of each leg. The resulting data can be seen in the table below,

Table 4.4. Travel Distance Data.

	Total Travel (mm)	**Travel (mm)**	**Average (mm)**
Leg 1			
1st Cycle	67	67	68.2
2nd Cycle	145	78	
3rd Cycle	205	60	
4th Cycle	274	69	
5th Cycle	341	67	
Leg 2			
1st Cycle	85	85	74
2nd Cycle	146	61	
3rd Cycle	225	79	
4th Cycle	297	72	
5th Cycle	370	73	
Leg 3			
1st Cycle	87	87	86.2
2nd Cycle	177	90	
3rd Cycle	264	87	
4th Cycle	345	81	
5th Cycle	431	86	
Leg 4			
1st Cycle	78	78	80.2
2nd Cycle	170	92	
3rd Cycle	245	75	
4th Cycle	325	80	
5th Cycle	401	76	

The results of the test shows that the travel of each limb is inconsistent with each other. The average deviation between each limb is around 7.435mm, with the most significant deviation is between the *Leg 1* and *Leg 3*. The smallest deviation is between *Leg 3* and *Leg 4* which is considered proportional with the previous data of the rotation angle, where the smallest deviation is also between *Leg 3* and *Leg 4*.

There are several factors that effects the inconsistency of the travel distance between each limb. The main reason being the previous deviation of the angle in each segment of the limb. Another reason is due to the data taking method, where the robot is placed on top of a free moving trolley. Some cycles produce noise in the data in the form of the trolley moving out of position. Moreover, the high performance of the suction hinders the process of moving the segment when trying to reach out, thus making the trolley move out of place when trying to complete the sequence.

4.3.1.3. Current Consumption

The current consumption is measured by activating all the components of a single limb system, which are mainly the motors and pumps. Each components drawn current can be seen in the table below,

Table 4.5. Measured Individual Current.

Leg	**Components**	**Average Current Measured (mA)**
1	Middle Motor	0.051
	Frame Motor	0.059
	Pump	0.841
	Overall	1.061
2	Middle Motor	0.057
	Frame Motor	0.049
	Pump	0.859
	Overall	1.055
3	Middle Motor	0.049
	Frame Motor	0.051
	Pump	0.838
	Overall	1.062
4	Middle Motor	0.053
	Frame Motor	0.051
	Pump	0.829
	Overall	1.057

4.3.2. Simultaneous System Testing

The simultaneous system testing is done by controlling the entire robot from the MCU, instead of a laptop. From the MCU, all of the systems can be controlled at the same time. This allows to measure the performance of the integrated systems, and the overall performance of the robot.

4.3.2.1.Vertical Robot Adhesion Test

The vertical robot adhesion test is done two separate times. The first being a half-assembled robot with two limbs attached adhering to a vertical surface. This test is performed directly by powering the vacuum pumps and placing the robot on a vertical surface. The robot held onto the vertical surface for 2 minutes without any changes, such as slipping, movement, or failures. The measured current during this test was 1.601A, with only two pumps running.

The second time this test was conducted, all four of the limbs' systems were attached to the main body frame and fully functional. When the robot was adhering to the surface of the vertical surface, material failure was determined to be imminent while being slowly released to adhere without any support. The material failure was avoided, and mitigated by attaching supports on the limb segments, which changes the structure characteristic of the segment making it able to handle more bending and twisting motion.

The robot was able to adhere on a vertical surface with all four limbs being active. A duration of 5 minutes was given to see if any changes occur. After the 5-minute duration, there was no changes in the state of the robot. The current drawn during this test was measured to be 3.592A.

4.3.2.2. Sequence Cycle Time

To measure the cycle time of the limb system completing a single sequence, the algorithm on the Raspberry Pi 4 is executed to run the continuously run the sequence of the limb system to the left or to the right. The cycle time that is measured can be seen below,

Table 4.6. Cycle Time Table.

Number of Sequence	Total Time	Time to Complete	Average Cycle Time
1	00:38:12	00:38:12	00:41:24
2	01:02:51	00:24:39	
3	01:50:14	00:47:23	
4	02:47:42	00:57:28	
5	03:25:51	00:38:09	
6	03:59:13	00:33:22	
7	04:53:58	00:54:45	
8	05:31:11	00:37:13	

Based on the data above, there are some inconsistencies on the time it takes to complete one sequence of movement. The longest time take around 57 seconds, while the quickest time takes about 24 seconds.

The high variation between the results is due to the framework of the program, where if there are two conditions that affect the completion of a cycle. The first condition being the successful adhesion of the suction cup measured by the current sensor. If the current sensor does not detect the current going over the adhesion point threshold, then it will keep trying to adhere to the surface until the condition has been met. The second condition is in the PID movement function where it will try to reach the target if it has not. Due to overshoots or the tuning of the constants, the process of this function may vary slightly which affects the acceleration of the limbs. Another factor which was stated before is the high performance of the suction causing it to stick to the surface, even though the motors are running. This is second most impactful factor on cause the higher duration of the cycle time.

4.3.2.3. Speed Test

The speed test is conducted by measuring the amount of time it takes the robot to travel a 2m distance. This is done using a stopwatch while allowing the robot to move on its own in a certain direction. This test held 5 attempts to measure the speed of the robot while sequencing the movement. The measure time to complete these attempts are show below,

Table 4.7. Speed Test Table

Attempt	Distance (mm)	Time
1	2000	06:49:12
2		06:59:53
3		08:56:74
4		06:54:13
5		07:24:14

Based on the data above, the average speed that the robot is able to achieve is approximately 1cm.s^{-1} or 60 cm. Considering the application of this robot, the speed is rather slower than the traditional method of using manual labor. However, as this is an unmanned autonomous robot in a hazardous environment, the results are considered acceptable. Somethings to factor from this test is that the trolley method affects the time that might decrease or increase it due to it rolling.

4.3.2.4. Overall Current Rating

During the speed test, the current drawn by all the components was also measured to observe the power consumption of the whole system. The highest recorded current being drawn, while in the condition where a moving sequence is being done was 3.863 A. This means that the current was measured while 4 motors of 2 limb systems are running, along with all 4 vacuum pumps being active. The lowest measured current was rated to be 0.061A, where all components that consume the highest energy are deactivated (pumps and motor).

4.4. Cleaning Module Test

Due to time constrains, the cleaning module test is not able to be done and completed. This test should be done in future development, as the cleaning module has already been designed and fabricated. The pump has been tested to transfer water from a reservoir into the nozzle, which provides sufficient spraying pressure. Moreover, the servo has sequence has already been integrated into the Raspberry Pi 4 board. Only the assembly is left to be completed, with some fixes to the hose connection due to leaking issues.

4.5. Troubleshooting

During the development of the robot, there has been various numbers of troubleshooting that was done. The most significant being the slave controller of the limb system. Due to unknown circumstances, the controller kept producing error in the form of a system hang. It was traced down to the I2C communication protocol that is used between the sensor modules, where if the controller attempts to communicate to these modules, it will hang the program. This problem was prominent during the development of *Leg 1* and *Leg 4*, but continues to persist in *Leg 1*. The problem seems to happen during initialization, but disappears after a certain amount of time has passed. Based on these symptoms and some research, it was speculated that the modules had improper discharge, so in accordance to that, external decoupling capacitors were added to try to eliminate this issue. The results proved no results as the effect still persists. More research was conducted into the matter, and another speculation is made where the state of the module is in “bus stuck” where the signals during the I2C communication were unsynchronized causing the IC chip that’s responsible for the I2C registers to have improper signals.

Another problem that seems to persist is the serial communication between the Raspberry Pi 4 and the Arduino Nano. The problem is that the USB port number for serial communication sometimes change by themselves causing the wrong commands to be inputted to the wrong limb module. Has not been researched into due to time constrains, but should be able to be prevented in the code of the Raspberry Pi 4. Temporarily to fix this issue, the USB connections are disconnected and reconnected in the order of the number of each limb system (e.g. first connection is *Leg 1*).

CHAPTER 5 – CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

The development and testing of the automated wall climbing robot for glass cleaning prototype have yielded valuable insights into its performance and its functionality. This study delved into the design and implementation phases, highlighting key aspects such as the structural design, locomotion mechanism, sensor integration, and control system architecture. These elements created the base in which the prototype was built, by creating a systematic approach to creating the solution for cleaning tall buildings.

In after the prototype fabrication was complete, the focus shifted towards evaluating the prototype's performance through various tests and analyses. The initial physical evaluation revealed a well-equipped robot capable of executing commands and handling its weigh. However, the assembly such as the cabling and MCU attachment were unable to be properly completed, highlighting the importance of meticulous assembly and integration processes in future iterations.

The vacuum adhesion system was tested to assess its holding capacity and reliability. While the system demonstrated a high success rate in adhering to surfaces, discrepancies between theoretical calculations and experimental data emphasises the complexities involved in real-world applications. Factors such as air leakages and joint bending were identified as potential areas for improvement to enhance adhesion performance.

The limb system tests proved to be the most challenging, where it involves the individual and simultaneous operations, rotation angles, travel distances, current consumption, and system integration. Despite some inconsistencies in travel distances and cycle times, the overall performance of the limb systems shows that by further developing this prototype, the capabilities in executing cleaning sequences and adhering to vertical surfaces can be very promising.

Although the cleaning module test was not fully completed due to time constraints, preliminary tests on pump functionality and servo integration can be used to ease the future development of this robot.

Throughout the development process, several troubleshooting efforts were undertaken to address several issues, the primary one being the I2C communication protocol errors and serial communication instability between components. These challenges highlighted the importance of robust system design, thorough testing protocols, and continuous refinement to ensure optimal performance and reliability in real-world scenarios.

In conclusion, while the autonomous skyscraper cleaning robot prototype has shown significant progress and potential, ongoing refinement, testing, and iterative improvements are essential before it can be a solution for efficient and safe cleaning operations in tall buildings.

5.2. Recommendations

The main recommendation for further developing this robot is by changing the material used during the prototyping phase. The material for the limb segments, which is currently 3D printed PLA, can be replaced with another light-weight material. The structure itself proves that it can handle the load weight of the robot on a vertical surface, however material failure can still occur if a wrong angle command is inputted.

The joints attaching the segments should be revised to connect both shafts (encoder shaft and motor shaft) to the segments. Currently the single set-screw holding the motor and segment together is sufficient, however in some cases it releases the shaft. The revision should also take into account the backlash effect to minimize the deviation between the angles.

The tie rods that connect the frame segments of the limb system needs to be refabricated to according to the dimensions given. The current tie rods were hastily made to save time, causing the limbs to have some sagging. Also consider adding a sliding slot at the edge of the frame segment which attaching it with the main body frame to minimize the sagging.

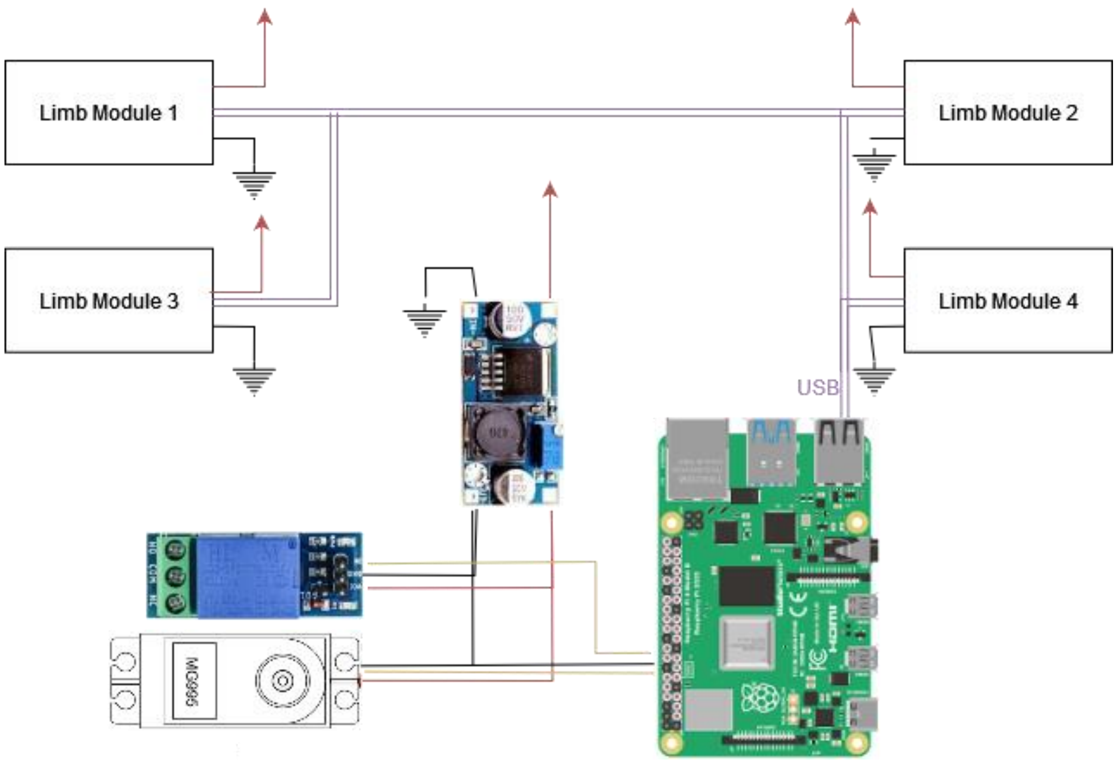
The material of the custom suction cups of the adhesion system can replaced with a more solid material (recommended: Polycarbonate), with the fabrication process being machining. The seals prove to be sufficient, however can also be replaced to further increase the holding force of the suction cup. Replace the temporary shafts that is on the ankle joint of the suction cup to aluminum.

Try to revise the mechanism to move the robot up and down (when attached vertically), due to it having a long cycle time to complete the whole sequence. To allow better design, consider replacing the joint of the suction cup into a ball joint. Moreover, consider adding a quick exhaust mechanism on the vacuum system to allow the suction cup to release quicker, which in turn will create a quicker cycle time.

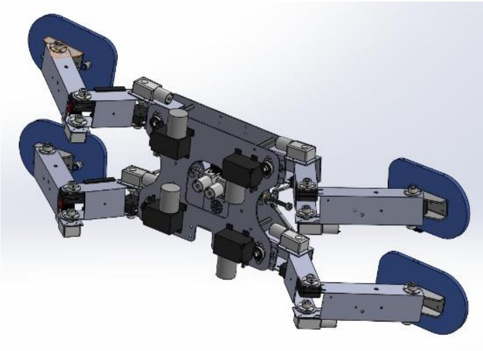
Since the I2C communication still proves to be problematic, consider switching the communication between module to an analog system if a matrix board is still used. Another alternative is to create an equivalent printed PCB to minimize the signal noise. Try to integrate a limit switch on the suction cup to allow further redundancy. Also, consider a shield to protect the electrical components that are present in the system from the water that is sprayed for cleaning.

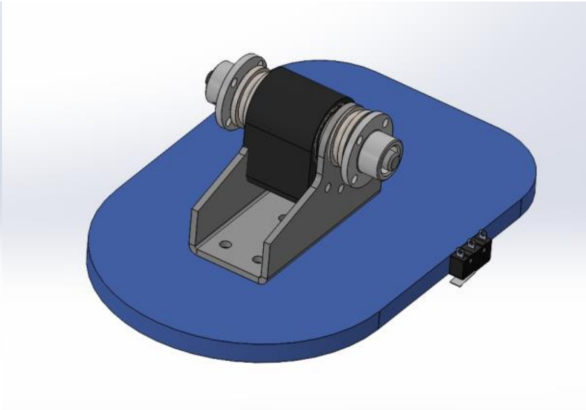
To eliminate or minimize the backlash issue of the system, the joints of the robot need to be rigid. The current design of using flanges with set screws creates a lot of backlashes. A custom-made flange can be made with a tight fit on to the motor shaft directly, which will minimize the backlash.

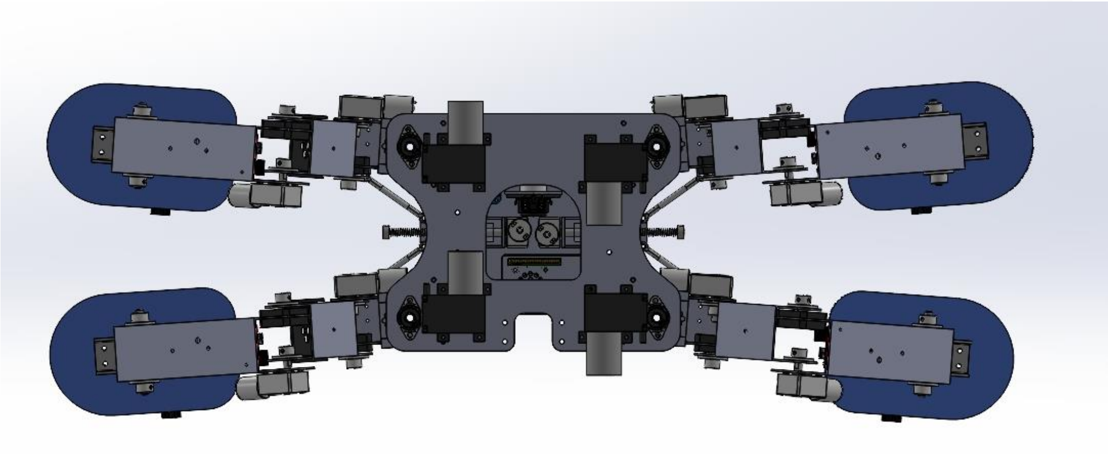
APPENDIX – Overall Wiring Diagram

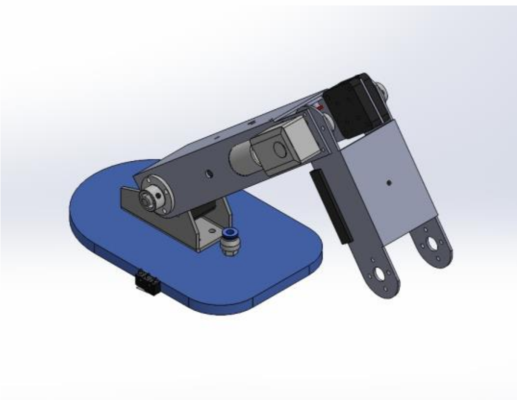


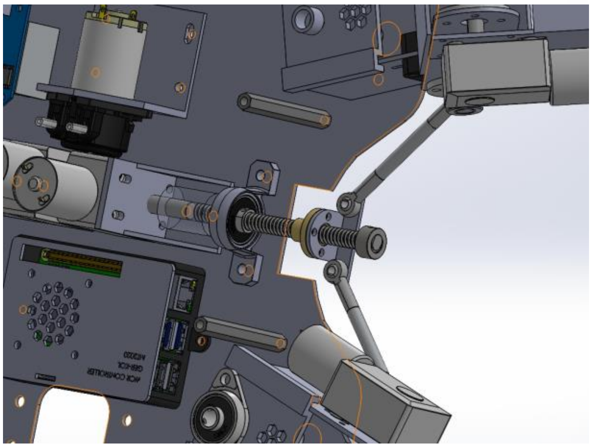
APPENDIX – CAD Design

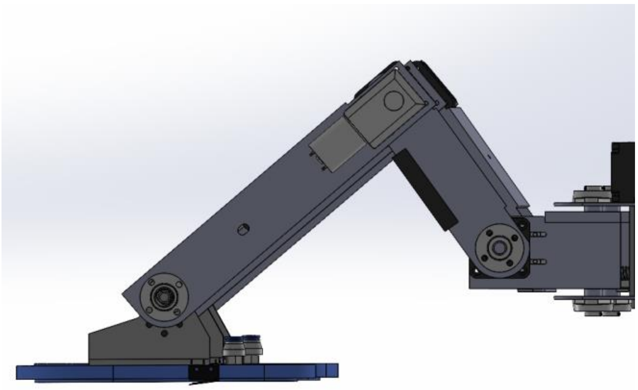


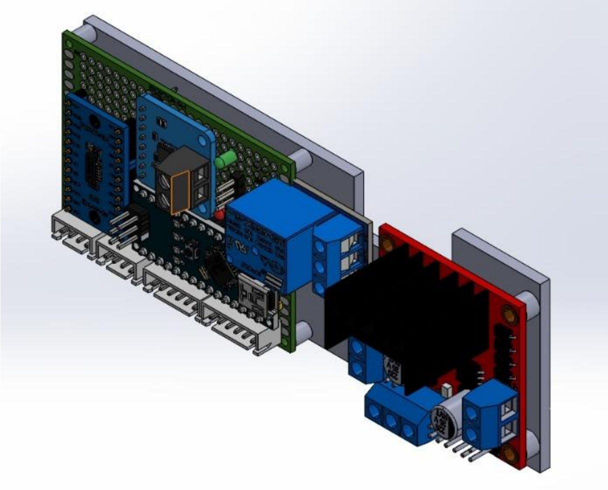


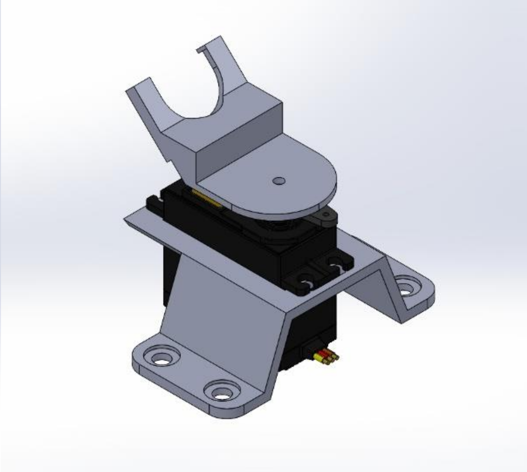


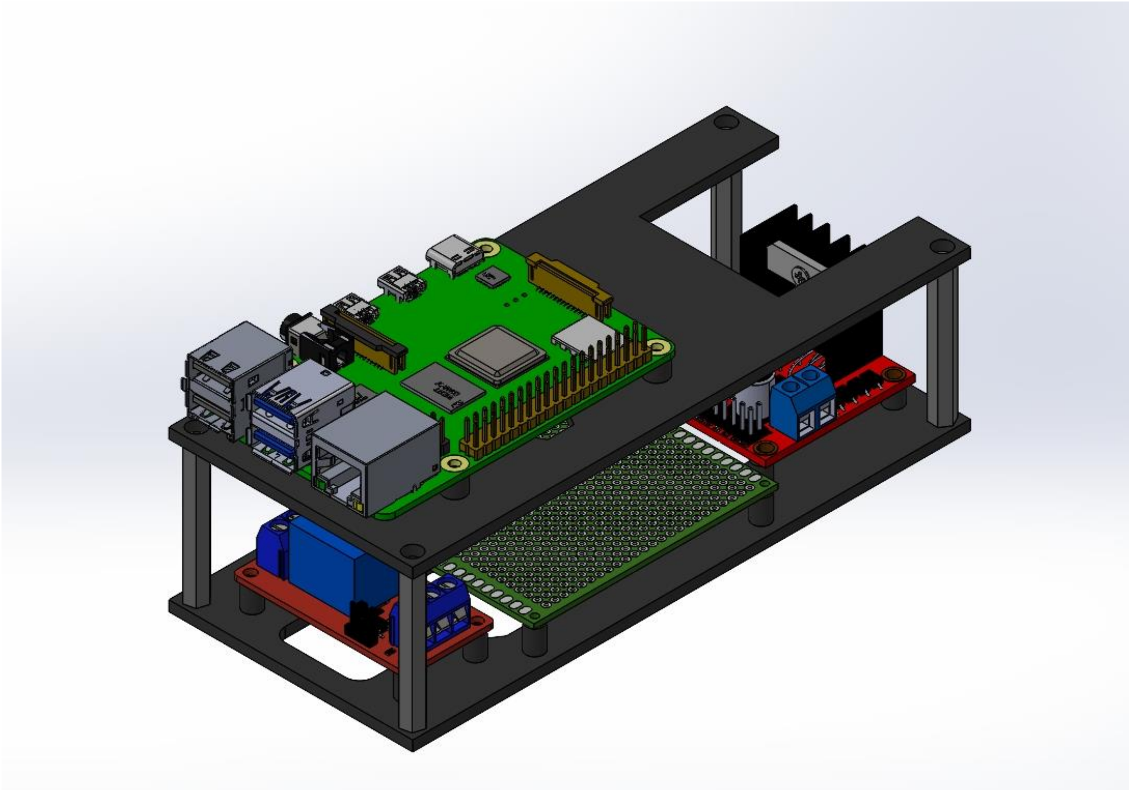












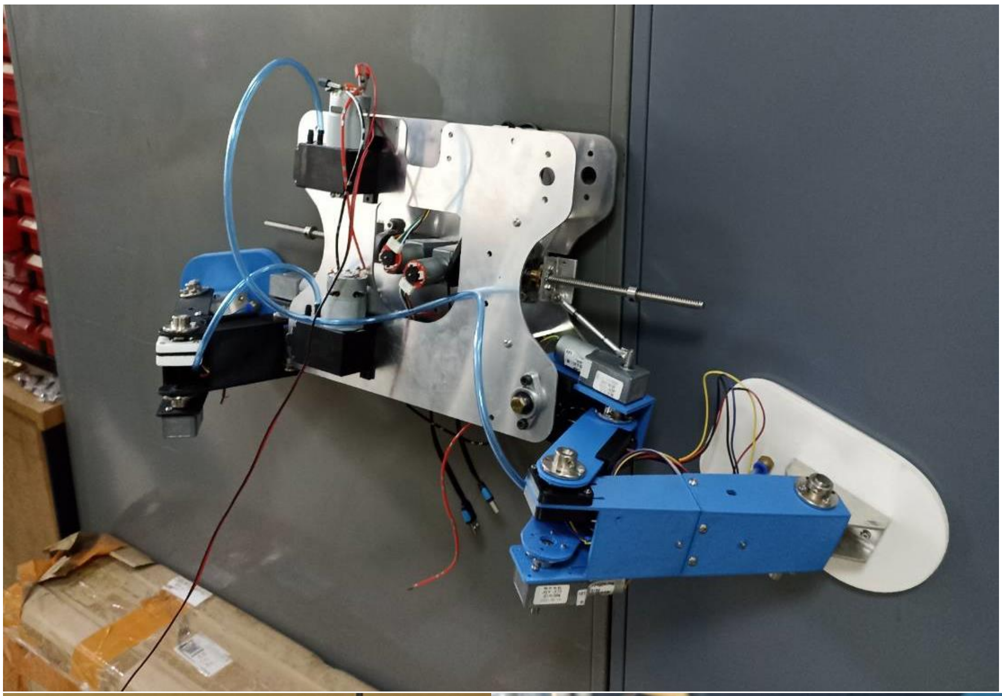
APPENDIX – Bill of Materials

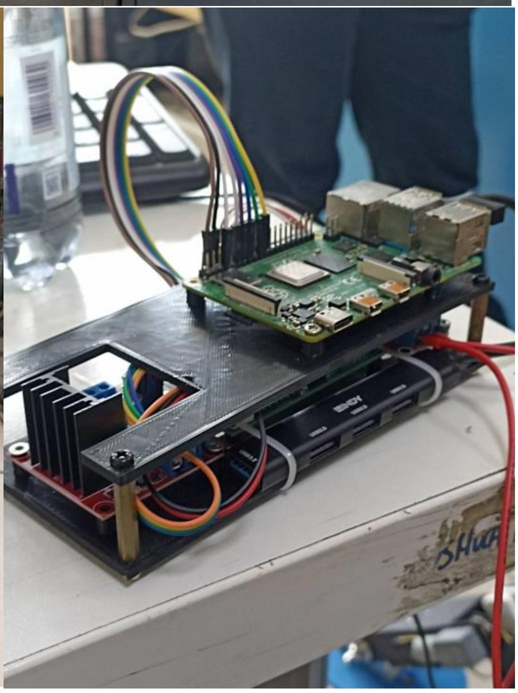
Limbs System BoM							
No.	**Part Name**	**Qty**	**Price**		**Total Price**		**Material**
1	Limb Ankle Segment	1	Rp	45,650	Rp	45,650	PLA
2	Limb Elbow Segment	1	Rp	25,300	Rp	25,300	PLA
3	Limb Frame Segment	1	Rp	22,000	Rp	22,000	PLA
4	Suction Cup	1	Rp	40,700	Rp	40,700	PLA
5	Ankle Bearing Housing	1	Rp	8,250	Rp	8,250	PLA
6	Encoder Housing	2	Rp	3,300	Rp	6,600	PLA
7	Encoder Bearing Housing	2	Rp	2,750	Rp	5,500	PLA
8	Ankle Support	1	Rp	8,800	Rp	8,800	PLA
9	Elbow Support	1	Rp	8,250	Rp	8,250	PLA
10	8mm Rigid Flange	4	Rp	28,000	Rp	112,000	Stainless Steel
11	6mm Rigid Flange	2	Rp	17,500	Rp	35,000	Stainless Steel
12	Ankle Joint	1	Rp	26,000	Rp	26,000	Aluminium
13	8x80mm Shaft	1	Rp	5,000	Rp	5,000	Aluminium
14	Encoder Shaft	2	Rp	1,000	Rp	2,000	Aluminium
15	End Ball Joint M4	2	Rp	11,300	Rp	22,600	Stainless Steel
16	Tie Rods	1	Rp	1,000	Rp	1,000	Aluminium
17	60cm Silicone Seals	1	Rp	43,499	Rp	43,499	Silicone
18	6mm Fitting M5	1	Rp	2,300	Rp	2,300	Brass
19	6mm Pneumatic Hose	1	Rp	11,500	Rp	11,500	Polyurethane
20	NBR70 O-Ring Seals	5	Rp	300	Rp	1,500	Rubber
21	NKN51101	2	Rp	3,700	Rp	7,400	Stainless Steel
22	608Z Bearing	4	Rp	3,430	Rp	13,720	Stainless Steel
23	JGY370 Motor (10RPM)	2	Rp	90,000	Rp	180,000	-
24	AS5600 Encoder	2	Rp	22,500	Rp	45,000	-
25	M4x12 Hex Cap	5	Rp	500	Rp	2,500	Stainless Steel
26	M4x10 Hex Cap	14	Rp	500	Rp	7,000	Stainless Steel
27	M3x30 Hex Socket	2	Rp	1,000	Rp	2,000	Stainless Steel
28	M3x8 Hex Socket	14	Rp	300	Rp	4,200	Stainless Steel
29	M3x12 Hex Socket	1	Rp	400	Rp	400	Stainless Steel
Total		78	Rp			695,669	

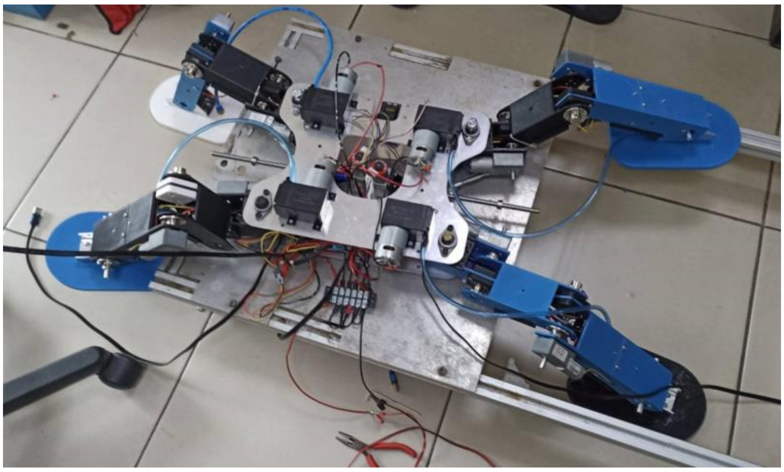
Limb Controller BoM							
No.	**Part Name**	**Qty**	**Price**		**Total Price**		**Material**
1	Arduino Nano 328p	1	Rp	39,900	Rp	39,900	-
2	INA226 Current Sensor	1	Rp	59,000	Rp	59,000	-
3	TCA9548 Multiplexer	1	Rp	10,500	Rp	10,500	-
4	5V Relay Module	1	Rp	5,300	Rp	5,300	-
5	L298N Motor Driver	1	Rp	15,500	Rp	15,500	-
6	Mini USB B to USB A	1	Rp	18,300	Rp	18,300	-
7	JST XH 2.54 6 pin	2	Rp	1,000	Rp	2,000	-
8	JST XH 2.54 4 pin	2	Rp	1,000	Rp	2,000	-
9	Jumper Cable	30	Rp	700	Rp	21,000	Dupont
10	JST SM 2.54 3 pin	1	Rp	700	Rp	700	-
11	M3x8 Hex Socket	2	Rp	300	Rp	600	Stainless Steel
12	M2x8 Screws	2	Rp	500	Rp	1,000	Stainless Steel
Total		45	Rp			175,800	

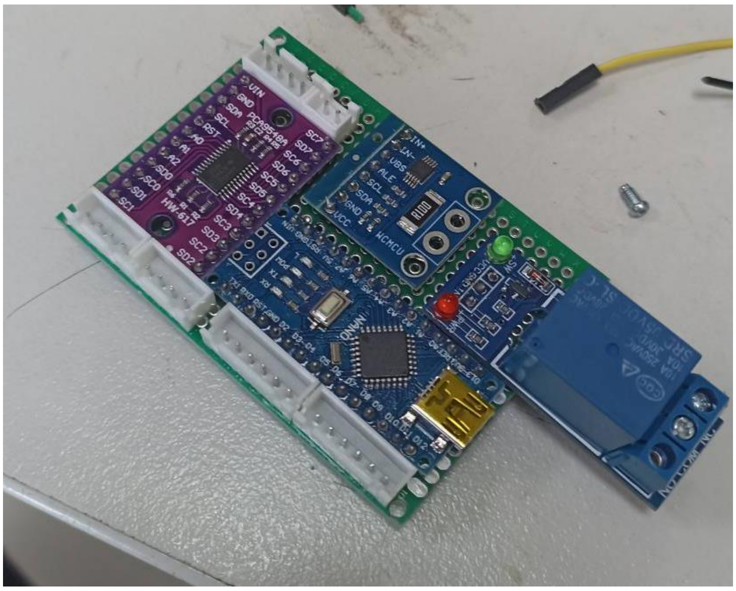
Main Body Frame BoM							
No.	**Part Name**	**Qty**	**Price**		**Total Price**		**Material**
1	Bottom Frame	1	Rp	255,000	Rp	255,000	Aluminium
2	Top Frame	1	Rp	255,000	Rp	255,000	Aluminium
3	KP08 Pillow Block	2	Rp	11,500	Rp	23,000	Aluminium
4	KFL08 Pillow Block	8	Rp	11,000	Rp	88,000	Aluminium
5	T8x150 Leadscrew & Nut	2	Rp	82,000	Rp	164,000	Stainless Steel
6	6x8mm Shaft Coupler	2	Rp	11,500	Rp	23,000	Aluminium Alloy
7	T8 Lock Collar	2	Rp	9,000	Rp	18,000	Stainless Steel
8	1/4 Inch Female Quickconnect	1	Rp	40,000	Rp	40,000	Brass
9	M4x60 Spacer	4	Rp	4,600	Rp	18,400	Brass
10	JGY370 Mounting	2	Rp	5,500	Rp	11,000	PLA
11	Servo Mounting	1	Rp	6,050	Rp	6,050	PLA
12	Shaft Spacer	8	Rp	550	Rp	4,400	PLA
13	Nozzel Mount	1	Rp	2,750	Rp	2,750	PLA
14	Raspberry Pi 4 Mount	1	Rp	12,100	Rp	12,100	PLA
15	Base Electronic Mount	1	Rp	12,100	Rp	12,100	PLA
16	Tie Rod Mount	1	Rp	5,000	Rp	5,000	Aluminium
17	6 Pin Terminal Block	1	Rp	11,500	Rp	11,500	-
18	Nozzle Quickconnect	1	Rp	46,000	Rp	46,000	-
19	5V Relay Module	1	Rp	7,000	Rp	7,000	-
20	L298N Motor Driver	1	Rp	15,500	Rp	15,500	-
21	Vacuum Pump	4	Rp	189,000	Rp	756,000	-
22	JGY370 (10RPM) & Encoder	2	Rp	277,000	Rp	554,000	-
23	Raspberry Pi 4B				Rp	-	-
24	LINDY 4 Port USB Hub	1	Rp	150,555	Rp	150,555	-
25	JST XH 2.54 3 Pin	1	Rp	1,000	Rp	1,000	-
26	JST XH 2.54 6 Pin	1	Rp	1,000	Rp	1,000	-
27	LM2596 Buck Converter	1	Rp	6,200	Rp	6,200	-
28	12VDC Pump R385	1	Rp	20,000	Rp	20,000	-
29	M8x40 Bolts	8	Rp	500	Rp	4,000	Mild Steel
30	M8 Nut	8	Rp	300	Rp	2,400	Mild Steel
31	M8 Washer	8	Rp	200	Rp	1,600	Mild Steel
32	M5x15 Hex Socket	24	Rp	800	Rp	19,200	Stainless Steel
33	M5 Nut	24	Rp	300	Rp	7,200	Stainless Steel
34	M4x8 Hex Cap	4	Rp	500	Rp	2,000	Stainless Steel
35	M4 Nut	4	Rp	200	Rp	800	Stainless Steel
36	M3x15 Hex Socket	4	Rp	800	Rp	3,200	Stainless Steel
37	M3x12 Hex Socket	12	Rp	700	Rp	8,400	Stainless Steel
38	M3 Nut	16	Rp	300	Rp	4,800	Stainless Steel
Total		165	Rp			2,560,155	

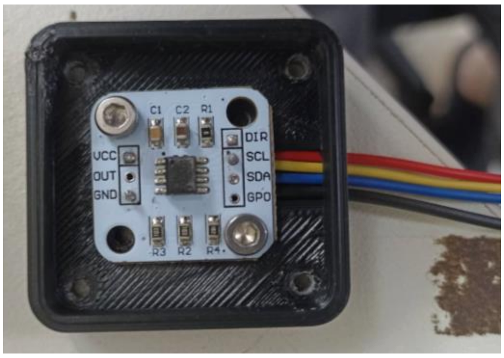
APPENDIX - Documentation










C1
C2
R1
VCC
OUT
GND
DIR
SCL
SDA
GPO
R3
R2
R4

APPENDIX – Slave Arduino Code

```
#include "defines.h"
#include "motor_control.h"
#include "globals.h"

#include <TCA9548.h>
#include <AS5600.h>

#ifdef SENSOR226
#include <INA226.h>
INA226 pump_monitor(0x40);
#endif

#ifdef SENSOR219
#include <Adafruit_INA219.h>
Adafruit_INA219 pump_monitor(0x40);
#endif

#include <Wire.h>
#include <PID_v2.h>

#include "string.h"

#define RP4
//#define PLOT //uncomment and comment PRINT to allow serial plotting
//#define PRINT //comment to disable printing
#define CONNECT_INA //comment to bypass

PCA9546 multiPlex(0x70);
AS5600 middle_encoder;
AS5600 frame_encoder;
PID_v2 middleMotorControl(Kp, Ki, Kd, PID::Direct);
PID_v2 frameMotorControl(Kp, Ki, Kd, PID::Direct);

double mValue;
double fValue;
int pValue;
int prevValue;
```

```
void setup()
{
  pinMode(PIN_RELAY     , OUTPUT        );
  pinMode(PIN_DRIVER_1A , OUTPUT        );
  pinMode(PIN_DRIVER_1B , OUTPUT        );
  pinMode(PIN_DRIVER_2A , OUTPUT        );
  pinMode(PIN_DRIVER_2B , OUTPUT        );
  pinMode(PIN_LS        , INPUT         );

  digitalWrite(PIN_RELAY, RELAY_OFF);
  Serial.begin(9600);

  //I2C Init
  #ifdef RP4
  Serial.println("Starting Controller");
  #endif
  #ifdef PRINT
  Serial.println("Initializing I2C Network");
  #endif
  Serial.flush();
  //WRCT = 0;
  Wire.flush();
  Wire.end();
  delay(1000);
  Wire.setClock(10000);
  Wire.begin();

  if(multiPlex.begin())
  {
    #ifdef PRINT
    Serial.println("Connected to TCA9546.");
    #endif
  }
  delay(200);

  multiPlex.selectChannel(AS5600_MIDDLE);
  if(middle_encoder.begin(4))
  {
    #ifdef PRINT
    Serial.println("Connected to AS5600 1.");
    #endif
  }
  //Wire.flush();
  delay(500);
  middle_encoder.setDirection(AS5600_CLOCK_WISE);

  multiPlex.selectChannel(AS5600_FRAME);
```

```
  if(frame_encoder.begin(4))
  {
    #ifdef PRINT
    Serial.println("Connected to AS5600 2.");
    #endif
  }
  //Wire.flush();
  delay(500);
  frame_encoder.setDirection(AS5600_CLOCK_WISE);


  if (!pump_monitor.begin() )
  {
    #ifdef PRINT
    Serial.println("Could not connect to INA226.");
    #endif
  }

  #ifdef SENSOR229
  pump_monitor.setMaxCurrentShunt(8, 0.01);
  pump_monitor.setAverage(INA226_16_SAMPLES);
  #endif

#ifdef PRINT
  Serial.println("I2C Devices Connected");
  #endif

  middleMotorControl.SetOutputLimits(-240, 240);
  middleMotorControl.SetSampleTime(10);
  // middleMotorControl.Start(middleEncoderAngle(), 0, 10);
  frameMotorControl.SetOutputLimits(-240, 240);
  frameMotorControl.SetSampleTime(10);
  // frameMotorControl.Start(frameEncoderAngle(), 0, 10);
  // while(middleEncoderAngle() >= 10)
  // {
  //   setMotorSpeedD1(middleMotorControl.Run(middleEncoderAngle()));
  // }
  // while(frameEncoderAngle() >= 10)
  // {
  //   setMotorSpeedD2(frameMotorControl.Run(frameEncoderAngle()));
  // }

  #ifdef PRINT
  Serial.println(" Current Encoder Level : ");
        Serial.println(middleEncoderAngle());
        Serial.println(frameEncoderAngle());

  Serial.println("Initialization Complete");
  #endif

  #ifdef RP4
  Serial.println("Init Complete");
  #endif

}
```

```
void loop()
{
  Serial.flush();
  if (Serial.available() > 0) {
    // Read the incoming data
    char message = Serial.read();

    // Buffer to hold the command string
    char command[20];
    int index = 0;
    delay(500);
    // Read until a newline character is received
    while (message != '\n' && index <= sizeof(command) - 1) {
      command[index] = message;
      index++;
      if (Serial.available() > 0) {
        message = Serial.read();
      } else {
        break;
      }
    }
    // Null-terminate the command string
    command[index] = '\0';
    #ifdef PRINT
    Serial.println("New Command :");
    Serial.println(command);
    #endif
    #ifdef RP4
    Serial.println("Command Recieved");
    #endif

    char* secCom;
    char* thirdCom;

    mValue = strtod(command, &secCom);
    fValue = strtod(secCom, &thirdCom);
    pValue = strtod(thirdCom, NULL);


    if(pValue != prevValue)
    {
      switch(pValue)
      {
        case 0:
        digitalWrite(PIN_RELAY, RELAY_OFF);
        delay(3000);
        //while(pump_monitor.getCurrent_mA() >= ADHESION_POINT);
        break;
        case 1:
        digitalWrite(PIN_RELAY, RELAY_ON);
        //while(pump_monitor.getCurrent_mA() <= ADHESION_POINT);
        break;
      }
      prevValue = pValue;
    }
    Wire.flush();
```

```
    if (mValue >= 270 && mValue <= 340 && fValue >= 245 && fValue <= 310)
    {
      #ifdef PRINT
      Serial.println(mValue); // Move motor M to the specified angle
      Serial.println(fValue); // Move motor F to the specified angle
      #endif

      bool middleRotate = true;
      bool frameRotate = true;
      middleMotorControl.Start(middleEncoderAngle(), 0, mValue);
      frameMotorControl.Start(frameEncoderAngle(), 0, fValue);
      delay(20);
      #ifdef PRINT
      Serial.println(" Current Encoder Level : ");
      Serial.println(middleEncoderAngle());
      Serial.println(frameEncoderAngle());
      #endif
      while(middleRotate || frameRotate )
      {
        if(middleEncoderAngle() <= (mValue - ROT_THRES) || middleEncoderAngle() >= (mValue + ROT_THRES))
        {
          setMotorSpeedD1(-(middleMotorControl.Run(middleEncoderAngle())));
        }
        else
        {
          middleRotate = false;
          setMotorSpeedD1(0);
        }
        if(frameEncoderAngle() <= (fValue - ROT_THRES) || frameEncoderAngle() >= (fValue + ROT_THRES))
        {
          setMotorSpeedD2(frameMotorControl.Run(frameEncoderAngle()));
        }
        else
        {
          frameRotate = false;
          setMotorSpeedD2(0);
        }
        #ifdef PLOT
          Serial.print(middleEncoderAngle());
          Serial.print(", ");
          Serial.print(mValue);
          Serial.print(", ");
          Serial.print(fValue);
          Serial.print(", ");
          Serial.println(frameEncoderAngle());
          #endif
      }
      #ifdef RP4
      Serial.println("Command Complete");
      #endif
    }
    else
    {
      #ifdef PRINT
      Serial.println("Error : Angle out of Bounds");
      #endif
    }
  }
```

```
        }
    }
    setMotorSpeedD1(0);
    setMotorSpeedD2(0);
}

double middleEncoderAngle()
{
  Wire.flush();
  multiPlex.selectChannel(AS5600_MIDDLE);
  return (middle_encoder.readAngle() * AS5600_RAW_TO_DEGREES);
}

double frameEncoderAngle()
{
  Wire.flush();
  multiPlex.selectChannel(AS5600_FRAME);
  return (frame_encoder.readAngle() * AS5600_RAW_TO_DEGREES);
}
```

APPENDIX – Raspberry Pi 4 Code

```
            line = limb_2.readline().decode('utf-8').rstrip()
            #print(line)
            if line == "Command Recieved":
                print("Run Leg 2")
                processL2 = True
            else:
                limb_2.write(angle2.encode('utf-8'))
        if processL2 == True and processL1 == True:
            break
        time.sleep(0.1)

def inputCommand2(angle1, angle2, suction1, suction2):
    global processL3
    global processL4
    limb_3.flush()
    limb_4.flush()
    angle1 = angle1 + " "  + str(suction1) + "\n"
    angle2 = angle2 + " "  + str(suction2) + "\n"
    limb_3.write(angle1.encode('utf-8'))
    limb_4.write(angle2.encode('utf-8'))
    while True:
        if limb_3.in_waiting > 0 and processL3 == False:
            line = limb_3.readline().decode('utf-8').rstrip()
            #print(line)
            if line == "Command Recieved":
                print("Run Leg 3")
                processL3 = True
            else:
                limb_3.write(angle1.encode('utf-8'))
        if limb_4.in_waiting > 0 and processL2 == False:
            line = limb_4.readline().decode('utf-8').rstrip()
            #print(line)
            if line == "Command Recieved":
                print("Run Leg 4")
                processL4 = True
            else:
                limb_4.write(angle2.encode('utf-8'))
        if processL3 == True and processL4 == True:
            break
        time.sleep(0.1)

def waitForCommand1():
    global processL1
    global processL2
    while True:
        processL1 = False
        if limb_1.in_waiting > 0:
            line = limb_1.readline().decode('utf-8').rstrip()
            #rint(line)
            if line == "Command Complete":
                print("L1 Done")
                processL1 = False
        if limb_2.in_waiting > 0:
            line = limb_2.readline().decode('utf-8').rstrip()
            #print(line)
            if line == "Command Complete":
                print("L2 Done")
                processL2 = False
        if processL2 == False and processL1 == False:
            print("Movement Done")
            break
def waitForCommand2():
    global processL3
    global processL4
        if limb_2.in_waiting > 0 and processL2 == False:
```

```
    while True:
        if limb_3.in_waiting > 0:
            line = limb_3.readline().decode('utf-8').rstrip()
            #rint(line)
            if line == "Command Complete":
                print("L3 Done")
                processL3 = False
        if limb_4.in_waiting > 0:
            line = limb_4.readline().decode('utf-8').rstrip()
            #print(line)
            if line == "Command Complete":
                print("L4 Done")
                processL4 = False
        if processL3 == False and processL4 == False:
            print("Movement Done")
            break
```

```
if __name__ == '__main__' :
#     servo = threading.Thread(target=runServo, args=(0))
#     limb_1 = serial.Serial('/dev/ttyUSB2', 9600, timeout=1)
    limb_2 = serial.Serial('/dev/ttyUSB0', 9600, timeout=1)
    limb_3 = serial.Serial('/dev/ttyUSB1', 9600, timeout=1)
    limb_4 = serial.Serial('/dev/ttyUSB2', 9600, timeout=1)
#     limb_1.flush()
    limb_2.flush()
    limb_3.flush()
    limb_4.flush()
    startCycle = True
#     servo.start()
    time.sleep(3)
    while startCycle :
        print("Move Left")
        time.sleep(1)
        inputCommand1(limb_1_reach, limb_2_reach, 0, 0)
        waitForCommand1()
        time.sleep(1)
        inputCommand2(limb_3_reach, limb_4_reach, 0, 0)
        waitForCommand2()
        time.sleep(1)
        inputCommand1(limb_1_far, limb_2_close, 1, 1)
        waitForCommand1()
        time.sleep(1)
        inputCommand1(limb_1_close, limb_2_far, 1, 1)
        waitForCommand1()
        time.sleep(2)
        #repetition
        while True:
            inputCommand2(limb_3_far, limb_4_close, 1, 1)
            waitForCommand2()
            time.sleep(2)
            inputCommand1(limb_1_reach, limb_2_reach, 0, 0)
            waitForCommand1()
            time.sleep(3)
            inputCommand2(limb_3_close, limb_4_far, 1, 1)
            waitForCommand2()
            time.sleep(2)
            inputCommand1(limb_1_far, limb_2_close, 1, 1)
            waitForCommand1()
            time.sleep(2)
            inputCommand2(limb_3_reach, limb_4_reach, 0, 0)
            waitForCommand2()
            time.sleep(3)
            inputCommand1(limb_1_close, limb_2_far, 1, 1)
            waitForCommand1()
            time.sleep(2)
            print("Sequance Done")
        startCycle = False
        break
```

REFERENCES

1. K. Al-Kodmany. 2018. “The Sustainability of Tall Building Developments: A Conceptual Framework”.
2. K. Kim. 2019. “Study on the Potential Demand of Windows Cleaning by Survey”
3. H. Zhang, J. Zhang, G. Zong. 2004. “Requirements of Glass Cleaning and Development of Climbing Robot Systems”.
4. M. A. Viraj, J. Muthugala, M. Vega-Heredia, A. Vengadesh, G. Sriharsha, and Mohan Rajesh Elara. 2019. “Design of an Adhesion-Aware Facade Cleaning Robot”.
5. Dr. John Dhanaseely.A. 2021. “Remote Controlled Painter Combined Window Washer using Quadopter”.
6. Uddin, S.M. Miraj, Md. Reyad Hossain, Md. Shahriar Rabbi. 2019. “Unmanned Aerial Vehicle for Cleaning the High-Rise Buildings”.
7. H. Fawzy, H. El Sherif, A. Khamis. 2019. “Robotic Façade Cleaning System for High-Rise Building”.
8. Y. Choi, K. Jung. 2011. “WINDORO : The World’s First Commercialized Window Cleaning Robot for Domestic Use”.
9. Z. Li, Q. Xu, L. M. Tam. 2021. “A Survey on Techniques and Applications of Window Cleaning Robots”.
10. R. D. Dethe, Dr. S.B. Jaju. 2014. “Developments in Wall Climbing Robots: A Review”.
11. K. Umam, Haryanto, R. Alfita. 2019. “Rancang Bangun Robot Pembersih Kaca Otomatis Berbasis Mikrokontroler ARM STM32 Dengan Sensor Proximity”.
12. R. Liu, R. Chen, H. Shen, and R. Zhang. 2013. “Wall Climbing Robot Using Electrostatic Adhesion Force Generated by Flexible Interdigital Electrodes”
13. N. Mir-Nasiri, H. Siswoyo J. and Md. H. Ali. 2018. “Portable Autonomous Window Cleaning Robot”
14. Raspberry Pi (Trading) Ltd. 2019. “Raspberry Pi 4 Model B”.
15. InvenSense Inc. 2016. “MPU-9250 Product Specification”.
16. Schmalz. n.d. “Theoretical Holding Force of a Suction Cup”.
17. WM Berg Inc. n.d. “Lead Screw Calculations“.
18. PBCLinear. 2018. “What is Lead Screw Efficiency?”.
19. Nadine Oosthuizen. 2020. “BOATSWAIN’S CHAIR [BOSUN’S CHAIR]”.
20. Hengxiang. 2022. “What is a suspended platform”.
21. S. Derammelaere, M. Haemers, J. De Viaene, F. Verbelen, K. Stockman. 2016. “A quantitative comparison between BLDC, PMSM, Brushed DC and Stepping Motor Technologies “

CURRICULUM VITAE

GERALDI KOLIN

Mechatronics Engineer

Contact

+6281212774948

grld.kln.cl@gmail.com

Geraldi Kolin

@geraldikolin

A young engineer that adapts to various environments, proficient in handling computer based tasks. Intrigued by the world of machinery and robotics.

Languages

Fluent English
Fluent Indonesian
A1 Deutch

EDUCATION

Swiss German University, Indonesia | 2020-2024
Mechatronics Engineering

Fachhochschule Südwestfalen, Germany | 2023
Electrical Engineering

SMA Don Bosco II Pulomas, Indonesia | 2017-2020

SKILLS

C++ Programming
Visual Basic
Arduino Programming
Ladder Programming
KiCAD Design
Solidworks CAD
Machining

EXPERIENCES

Teaching Assistant | 2024
Swiss German University

Production Intern | 2023
SEM Schalt- und Energietechnik München GmbH

Engineering Intern | 2022
PT. Astra Honda Motor

Machining Training | 2022
Politeknik Industri ATMI

ACHIEVEMENTS

2nd Runner Up - Universitas Kristen Maranatha
Maranatha Automation Championship | 2024

AIJAM Japan Gold Award - Tokyo Innovation Hub
Advanced Innovation Creative Olympic Japan | 2019

Indonesia Inventors Day Gold Award - INNOPA
The 6th International Young Inventors Award | 2019