LEBANESE AMERICAN UNIVERSITY

Mechatronics System Design I
Final Project

Instructor: Dr Bilal Komati

Dustbuddy: The Autonomous Solar Panel Cleaner

Table of Contents

Table of Figures

Table of Tables

Introduction

Solar energy is the most abundant of all energy resources and can be easily employed through the installation of solar panels constructed of photovoltaic cells (PV) that converts the sun radiation to electricity by a process of photon absorption and electrons excitation. The development of industries and orientation toward green energy play a significant role in relying heavily on solar panels which lead eventually to reduce the emission of fuel and gas combustion.

For the mentioned reasons, people opt to install solar panels to procure their daily needs of electricity consumption. This is a fabulous alternative for avoiding skyrocketing electricity bills especially in Lebanon where also we have three hundred days of sunshine during the year.

However solar panels could experience efficiency reduction due to accumulated dust or unclean panels blocking solar radiation from interacting with PV cells. This is a major issue since the light obstruction materials (accumulation of dirt or particles like dust, sand, and moss) pose as external resistances that reduce solar photovoltaic performance. It was found that the external resistance could reduce the photovoltaic performance by up to 85% and the value of short circuit current, power decrease with respect the amount of dust on the solar panel.

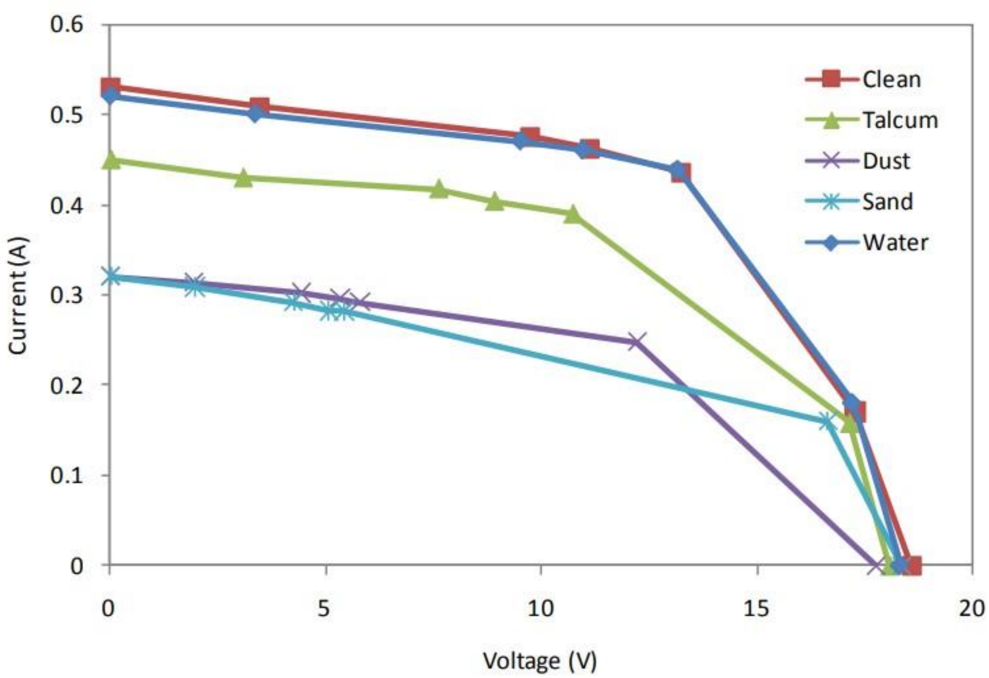


Figure 1 I-V characteristic and external resistance effects

Problem to Solve

From the above discussion we can say that natural dust is the one of the causes to reduce the output of the solar panel. To overcome these problems, a proper maintenance operation for the solar panels would be necessary. Particles like dust and sand can be reduced when washed away by water thus proper cleaning would be required. Therefore, to ensure cleaning maintenance we propose a solar panel cleaning robot that offers regular cleaning for the panels when needed. This could be useful also in saving lives. When it comes to solar panels, it is essential to be aware of the potential risks of cleaning them. Without proper safety precautions, a person can be electrocuted while cleaning photovoltaic cells. This is a severe hazard that you should not take lightly since you may not be grounded in the photovoltaic arrays.

Figure 2 clean versus dusty solar panels

Project Specifications

It is evident that the installation of solar panels follows certain standards especially in large production firms. To collect solar power energy more efficiently, solar panels should be angled to face as close to the sun as possible. Photovoltaics produce power when the angle at which the sun’s rays hit the panel surface (angle of incidence) is small, or when light strikes the panel as close to perpendicular as possible. Therefore, the best angle for your solar panels is the one that allows the panels to get the most direct, perpendicular light. For Lebanon, the solar panels should be installed at an angle around 25~35 degrees from the flat surface and should be facing south in an un-shaded area.

Figure 3 different angle orientation

From the above discussion we can notice that the first obstacle is the issue related to the angle. If we need to build a robot, we need to overcome slope slippage. This is a main challenge since our target is to build modular cleaning mechanism capable of covering different solar panels installation rather than static mechanism such as conventional wipers available for large firms.

The second challenge is to provide a suitable efficient cleaning mechanism that is capable of cleaning panels’ surfaces rather than damaging them.

After making sure that we have a robot capable of withstanding slopes and providing efficient cleaning, we can move to the third challenge which is creating a cleaning algorithm.

Each task be solved alone, taking into consideration time and capacity constraints the project represents new moderated to hard challenges that require design, mechanical, and coding skills along with creative touch.

Design and Proposed Solutions

Chassis and Robot Structure

The first obstacle mentioned was the issue of slope climbing. This cannot be resolved using a standard wheel-based robot since these types of robots have small wheels providing reduced contact area with the surface. Any plastic wheel will fail to provide the necessary grip to overcome slopes. Additionally, when moving from panel to another they will fail due to the spacing between panels. Therefore, we can deduce that we need a non-wheel-based robot. We opted for a caterpillar design and started to search for available models. As first selection we chose the below model:

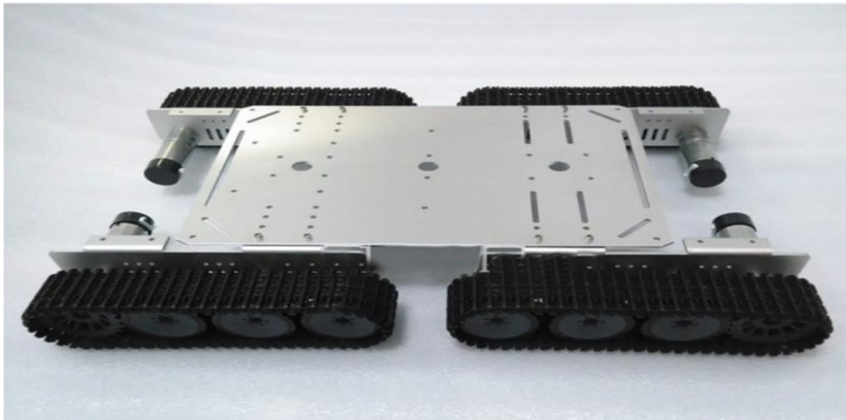


Figure 4 available caterpillar model

As first look one can see it as perfect design however the caterpillars were made of plastic along with a lightweight plastic and metal body and the dimension were small. As such the problem was intensified. All available models were overpriced and will lead eventually to failure. The only way to solve this problem is by creating our own design. Employing the idea of caterpillar, we built a customized powerful model defeating the conventional available models. We started with multiple design iterations and changed the structure based on the performance to finally meet the requirements. In What follows we will see the different model iterations:

Figure 5 first design (car seat chassis)

As we can see the chassis is made of a car seat which was our initial starting point. We wanted to use the back of the seat as our cleaner since it had the ability to move up and down. Moreover, we did not have appropriate belts thus we needed to cut one long belt and connect it. We discovered that the belts were not strongly bonded which will create slippage problems and malfunctioning when moving from one panel to the other. The model was big which implies that we have a load problem. Thus, we reconsidered the design of the chassis.

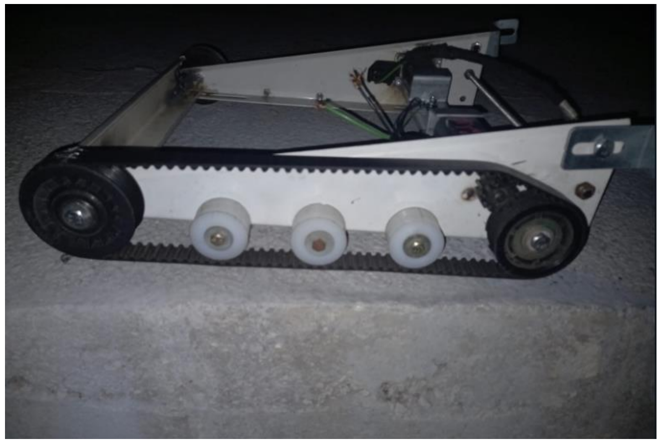


Figure 6 second design (enhanced chassis with polyamide wheels)

As we can notice this time, we designed a new compact chassis that include polyamide wheels that will help us moving from one panel to the other easily. Moreover, we can notice that we added a gear and motors to make the caterpillar move. However, the gears were customized to fit the belt teeth which led to slippage once some teeth were jumped over. To solve this problem, we needed to maintain good contact with our gear. This will be done using pulleys that compress the belt to the gear.

Figure 7 third design (with fixed belt pulleys)

In this design we can see an enhancement of the previous design by adding fixed pulley to press the belt toward the gear. As such we avoid the case of belt's teeth jumping more than one at a time. However, the problem is in the rigidity of the fixed pulley. When stress is coming from the bottom part, the belt will be in excessive tension around the gear which affects motion. To avoid this problem, we must use modular pulley to compress the belt toward the gear.

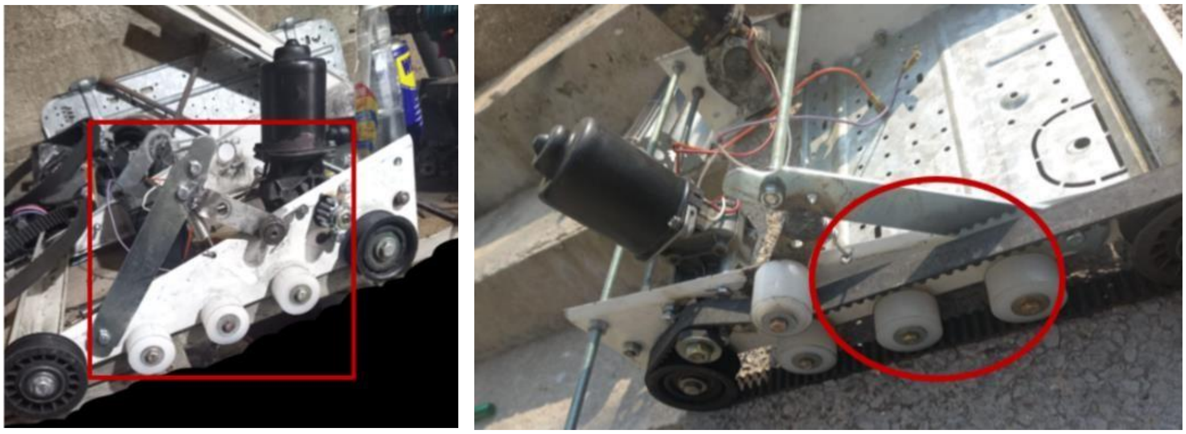


Figure 8 fourth design (modular pulleys with spring loaded "tensioner")

To move from fixed to modular pulley we were inspired by the car mechanism "tensioner." We tried to implement a similar behavior by allowing pulley to move by spring mean. This design allows us to reduce tension on the gear when moving from panel to another and in case the panels are not aligned at the same height this will insure a perfect transition.

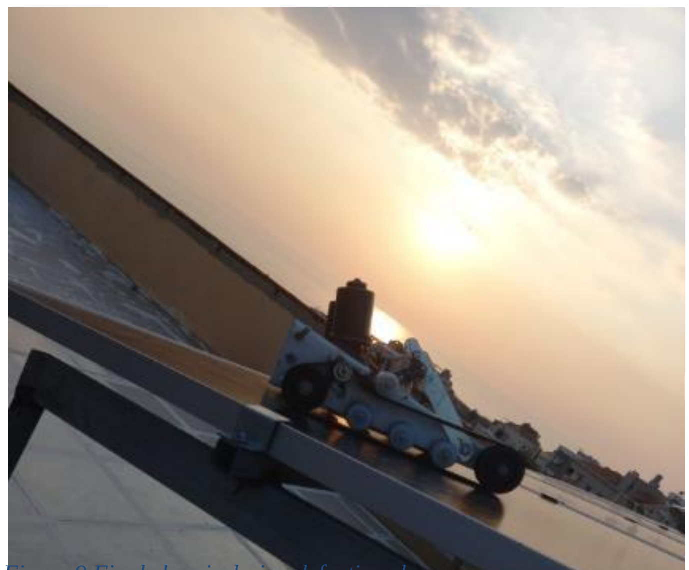


Figure 9 Final chassis design defeating slopes

By following the distinctive design procedure, we successfully designed a differential rover capable of defeating and climbing slops. Our final chassis model is in the adjacent picture. As we can see the robot is standing on the panels without any problem.

The effort put in the mechanical design is a considerable effort since it is not a conventional design available in the market. The rover is a customized and assembled part by part from different components that include customized belts, car pulleys used as wheels, polyamide wheels, and copper tubes fixing the wheels to the chassis, springs to support the moving pulley. We can say that as first step which is the base of the project is achieved by producing a successful model able to climb slopes.

Cleaning Structure

Figure 10 Cleaning Mechanism

The cleaning mechanism can be considered as our second challenge. The main difficulty consists in ensuring a brush able to clean the distance traveled by the robot. This can be done using two separate disks rotating each having a diameter equal to half the width of the robot. This is technically realizable however it implies the need of two motors each for one brush or the use of gear assembly like the differential gear where two brushes will be rotated by one motor. The second solution is to use one a single brush able to cover the width of the robot and a motor attached at one end. The problem faced in this approach was the motor installation which was solved by adding chain and gears between the motor and brush. The model is shown in the below picture:

Further explanation of the implemented functionalities and control mecanism will be provided in later sections in the report.

Hardware Used

In this section we will present the list of different hardware used along with the calculation needed to justify our choice.

Microcontroller

Every project contains a microcontroller responsible of executing programmable task. The selection of microcontroller is crucial since it will be the brain of the project delegating tasks and ensuring functionalities. In our case we chose the “Arduino Mega ADK” which is like the “Arduino Mega 2560” with one additional feature that it can be connected to android phones offering additional functionalities. This microcontroller was selected because we needed additional pins than those present in “Arduino UNO,” and it offers more memory and processing power.

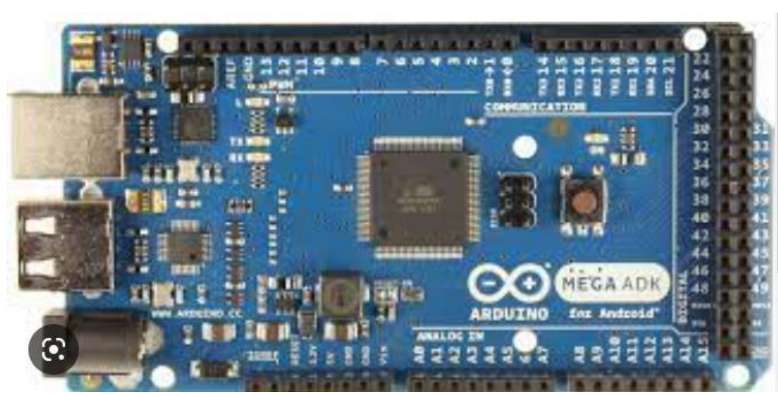


Figure 11 Arduino Mega ADK

Motor selection

The selection of our motors is a particularly major step in the design since we have slopes to climb along with the weight of the robot and other components such as power source water tank and electric components (we could design external power and water sources however we wanted these features to be embedded in the robot). As mentioned previously, our designed is based on differential meaning it is driven by two motors. The calculation done to approximate the size of needed motors is as follows:

- Step 1: determine the design constraints: speed, load, weight, climbing angle… ○ Weight of the robot: w = 10kg (including battery and water tank) ○ Maximum speed of the robot: v = 0.2m/s (belt speed driven by motor gear) ○ Maximum incline to climbs: $\theta = 40°$
  ○ Reach maximum speed in two seconds (acceleration) ○ Belt length from center of the rear pulley to the front pulley:
- Step 2: Calculate the rotational speed of wheels:
  (We can remove π since we are not converting to circumference and assuming the diameter of the pulley is added to the length of the belt since we are measuring from center to center)
  ○ Number of turns: $N_T = \frac{60v}{2L} = \frac{60*0.2}{0.46} \approx 27$ rpm
- Step 3: Calculate the motor force (thrust force) ○ If we omit friction, we can estimate the motor torque needed to climb the maximum slope previously specified. The force needed to overcome weight and gravitational acceleration can be calculated using Newton's second law
  $\sum F = ma$ ○ Assuming constant maximum speed of 0.2m/s and no initial speed we can calculate the acceleration using the equation of motion $v - v_0 = at$. Solving for a: $a = \frac{v}{t} = 0.1m/s^2$
  ○ $F_T - mg * \sin(\theta) = ma$ ○ $F_T = mg * \sin(\theta) + ma$ ○ $F_T = 10 * 9.81 * \sin(40) + 10 * 0.1 \approx x\ N$
- Step 4: calculate power ○ Since the nominal speed is not specified, we will assume that the robot will be driven at constant maximum speed once starting to move. To calculate the power required: $P_m = F_T * V = x * 0.2 = y\ W$
- Step 5: calculate torque ○ Mechanical power for the entire robot was found, so we have y/2 W per one traction motor. The corresponding torque per motor: $T_T = \frac{1}{2} * R * F_T \approx z\ Nm$
- Step 6: summarize the values calculated ○ Rotational speed: $N_T = 27\ rpm$ ○ Torque: $T_T = z\ Nm$ ○ Power: $P_T = y/2\ W$

Since we need a considerable torque, we need to consider using geared DC motors. After searching for available motor and comparing performance versus cost we selected motors used for car wipers. Below a brief description from datasheet:

As we can see, the wiper motors perfectly match our needed characteristics. The most important two features to extract from the provided description are operating voltage and current. We will operate the motor at 12 V where the current will vary ideally between 4.5 and 5 Ampere. For safety reasons we will consider the maximum output current to be 7~8 A.

Motor Driver Selection

GERIA Basic technical parameter (WM6061) 30W

Voltage:	12V		24V	
Test Voltage:	13.5V		27V	
Braking Torque:	28N.m		28N.m	
Working Torque:	6N.m		6N.m	
	Low	High	Low	High
No-load Speed:	35 rpm	52 rpm	35 rpm	52 rpm
No-load Current:	1.8A	2.5A	1A	1.5A
Working Speed:	30 rpm	45 rpm	30 rpm	45 rpm
Working Current:	4.5A	5.5A	2.5A	3.5A
Noise	50dB	55dB	50dB	55dB

Figure 12 wiper motor specifications

After finding the max output current of the two motors we must find a suitable motor driver

The L298N motor driver has a maximum current of 4A which is less than our maximum current, so we cannot use this module.

We then researched to find 2 types of motor drivers DBH-01C Dual channel motor driver which has a maximum current of 20A which is suitable for our design however, it cost 35$ which is out of our price range, so we settled for two BTS7960 with a peak current of 43 Amps which cost 14$ each (28$ total).

We used the dual L298N motor driver for both the pump and the brush motor

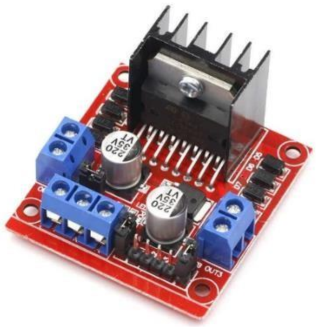


Figure 14: L298N

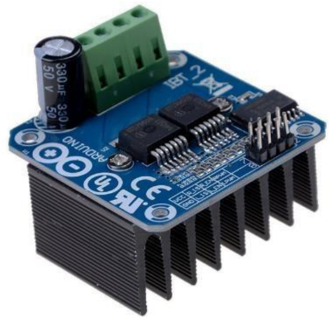


Figure 13:BTS7960

Other Sensors

IR sensors

Our design uses 4 IR sensors (infrared sensors). An infrared sensor includes two parts namely the emitter & the receiver. The sensors send a digital either 0 or 1 to the Arduino. There are two sensors on each of the back and front side of the rover. These sensors are used to detect when the rover comes to the edge and determine how will the rover move based on the values of these IR sensors.

Accelerometer

Figure 16: IR sensor module

Our design uses the MPU6050 which consists of three axis accelerometer and three-axis gyroscope. We can measure the velocity, orientation, acceleration, displacement. To connect it to the Arduino we must install the MPU6050 libraries to use the features. The accelerometer was used to collect data of the X, Y, Z positions of the rover and to determine the path of movement depending on the position of the rover in these planes.

Figure 15: MPU6050

Bluetooth Module

As an extra feature, we designed an android Bluetooth app to give access to the user to navigate the rover and override the autonomous motion. This feature was attained by linking the Bluetooth

app by the Arduino using HC05 Bluetooth module. The HC05 module is operated by connecting the Rx pin of the module to the Tx of the Arduino and the Tx pin to the Rx of the Arduino.

Battery

To power the motors attached to the motor drivers a DC 12V battery with initial current 2.7A MAX.

To power the Arduino a 9V battery was used with its cable

Figure 17: HC05

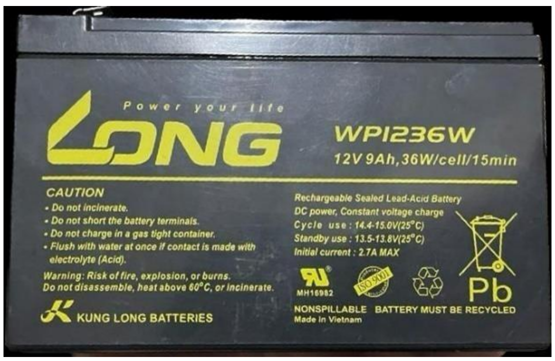


Figure 20: 12V battery

Figure 18: 9V battery

F

Pump

To transfer water from the tank to the brush we used the R385 Mini Aquarium Water Pump DC 6V- 12V

Figure 21: R385 water pump

Bluetooth Application

The Bluetooth application designed using MIT app inventor which contains:

- Devices to connect to the Bluetooth module (HC05)
- Four buttons to navigate robot (Forward, Backward, Turn Left, Turn Right)
- Pump button to flow water to the brush to clean
- A brush switch to turn the brush's motor on and off
- Two text boxes for the user to input the height and the width of the panel (this is used to store data)
- A stop button to stop the rover

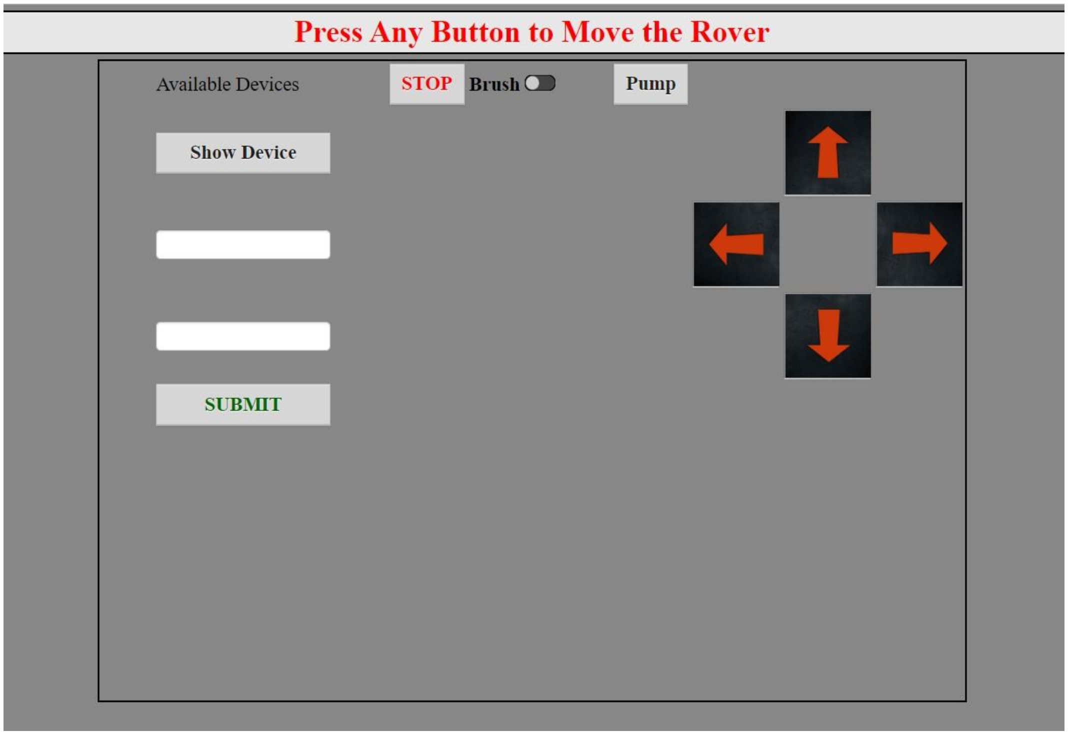


Figure 22: Bluetooth app screen

Fine State Machines

Algorithm Flowchart

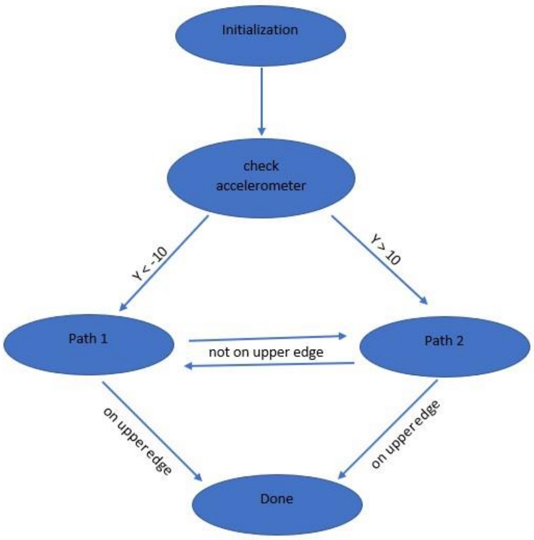


Figure 23: Algorithm Flowchart

This flow chart summarized the algorithm used according to the value of the accelerometer

Initialization FSM

Describes how the rover moves to its initial state

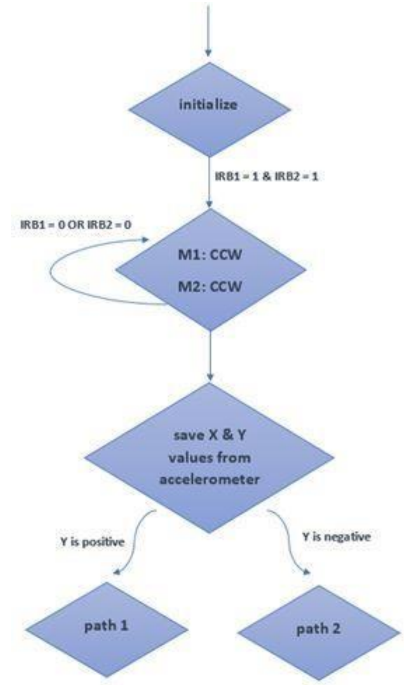


Figure 24 initialization FSM

Paths FSM

Descries the trajectory of the rover depending on the IR sensors

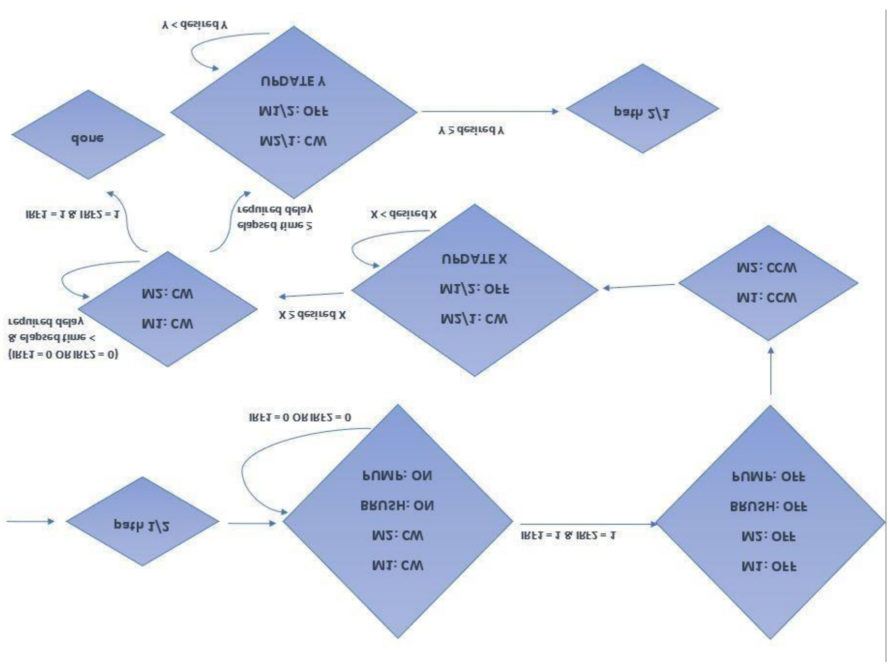


Figure 25 paths FSM

Done FSM

Describes the path when the rover reaches the edge

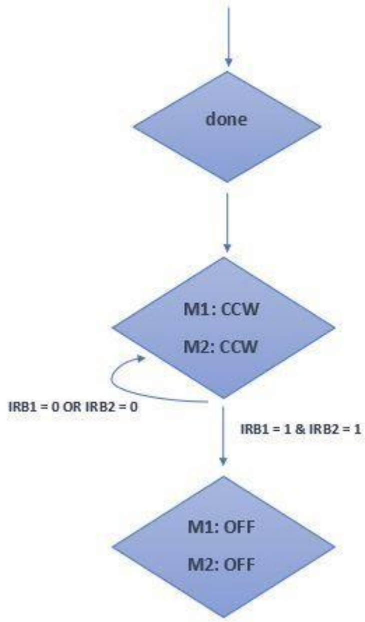


Figure 26 done FSM

Pump FSM

Describes how the pump sprinkles water and stops on different time intervals

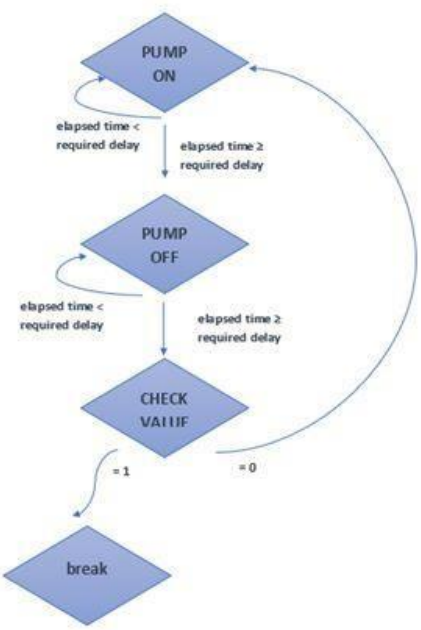


Figure 27: pump FSM

Bluetooth App

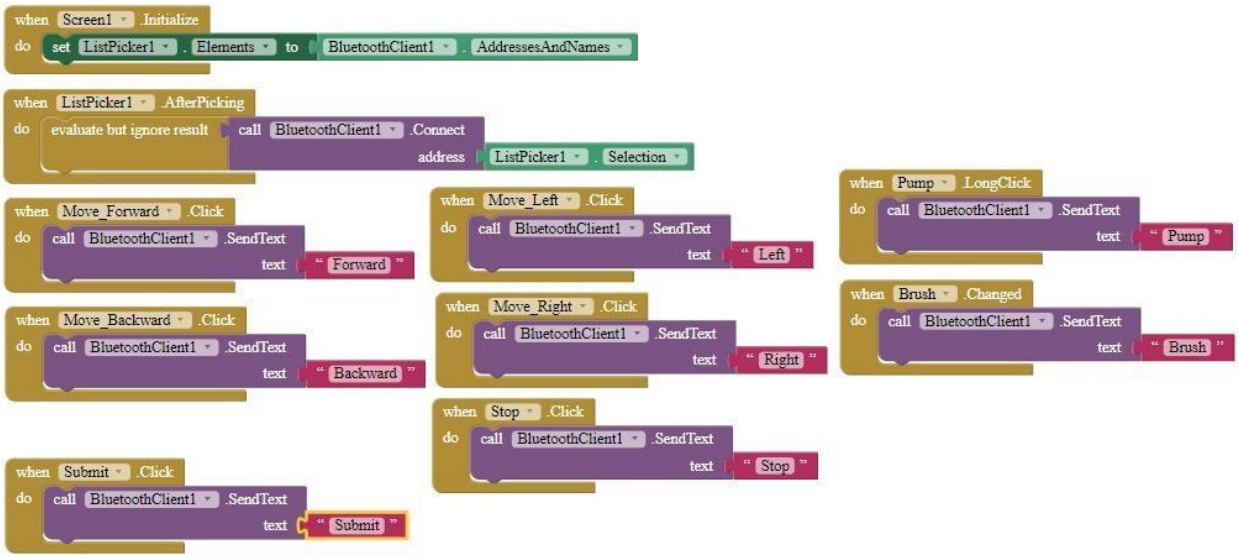


Figure 28: Bluetooth app FSM

Method to Connect the System

Wiring Diagram

The below picture describes how the sensors and actuators were connected to the Arduino Mega

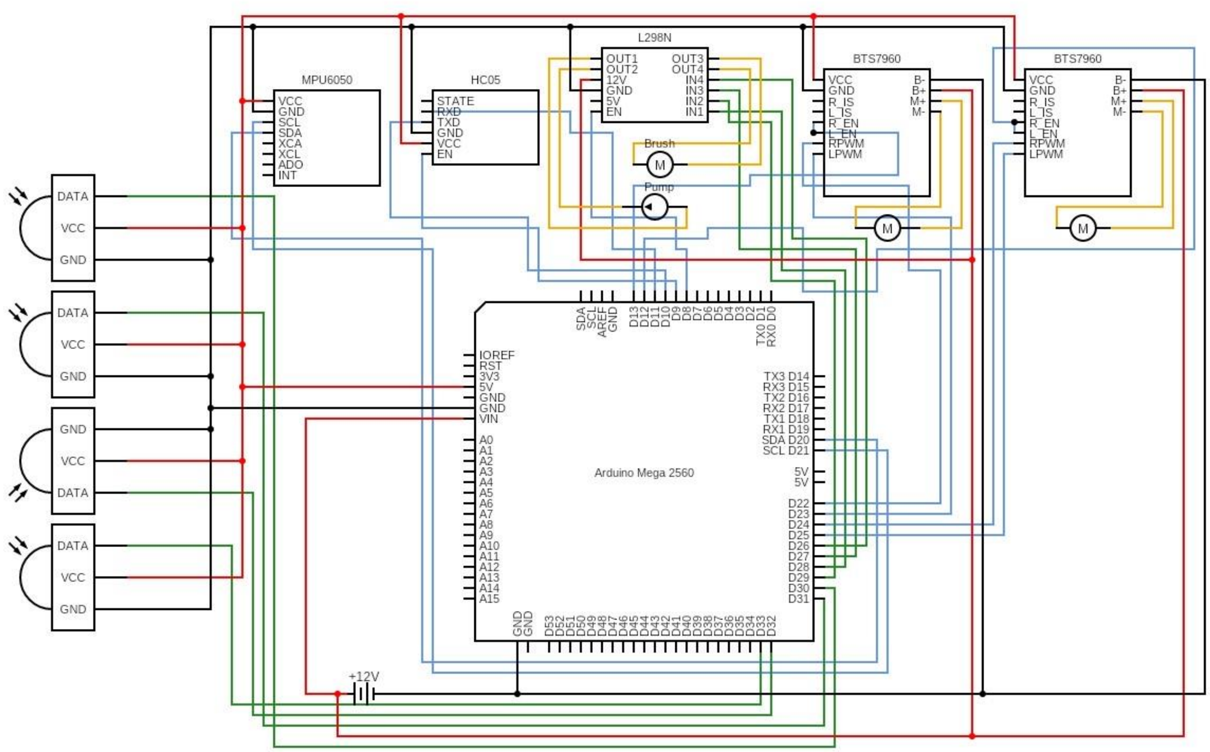


Figure 29: Circuit Diagram

Cost

Item	Price $	Quantity	Total $
Arduino Mega	38	1	38
Battery 12V	8	1	8
Battery 9V	2	1	2
Chassis	65	1	65
Wiper Motor 1	17	2	34
Drill Motor 2	25	1	25
BTS7960	14	2	28
L298N	3	1	3
HC04	5	1	5
MPU6050	4	1	4
Brush	5	1	5
IR sensor	1	4	4
Pump	5	1	5
			226

Table 1: Cost summary

Procedure Followed for Testing

First, after designing the chassis the model was tested against slippage and transition from panel to the other before proceeding with other tasks (Refer to figure 9). Then, we placed the chassis on a table to test the codes related to the IR sensors; if the IR detected '1' it will proceed with the following command (stop or turn back), else it will continue moving. We then tested the speed of the motors and tried to synchronize them to have the desired rotation. Next, we tested each method (Initiation, Path 1, Path 2) alone to simplify the debugging procedure. Then, the same methods were tested among the inclined plane to ensure that the rover does not slip and functions normally. Then, the brush and pump codes were tested and were functioning at the programmed time. After that, the Bluetooth application was tested to ensure proper functionality. Finally, we proceeded to join all pieces of code together to finalize the Arduino code and did all necessary final tests to finalize the project.

Results and Analysis

We collected almost one thousand samples from the MPU6050 the following results are described in the graphs below:

As shown aside the accelerometer Vs sample graph (Figure 28) the Z component was ranging between 8g and 10g, X between 4g and 5g, and Y between 0g and 2g

As shown aside the gyroscope VS sample graph (Figure 29) the X component was ranging between 0.17dps and -0.05dps, Z between -0.05dps and 0dps, and Y between -0.05dps and 0.1dps.

The high pass filtered linear position VS sample (Figure 31) and the high pass filtered position VS sample (Figure 30) graphs are shown below

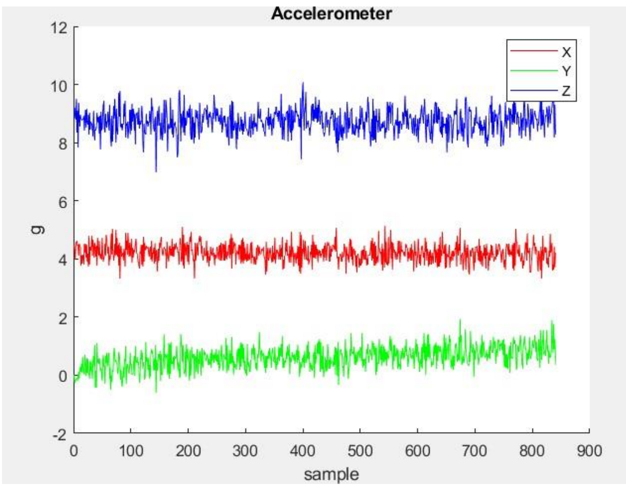


Figure 30 accelerometer vs sample graph

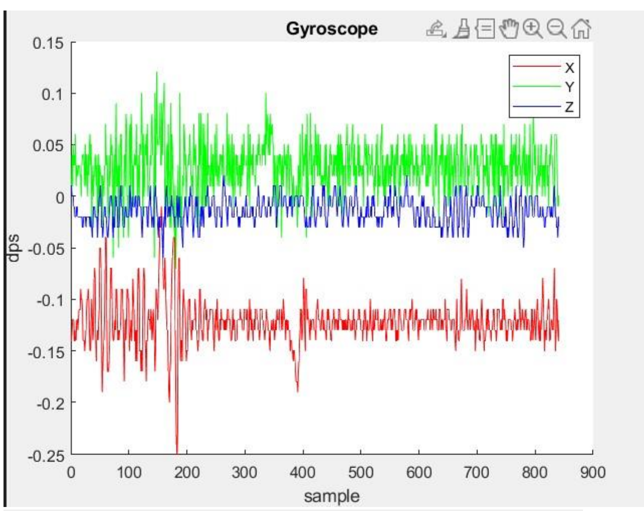


Figure 31:gyroscope vs sample graph

As shown aside the Linear acceleration VS sample graph (Figure 32) the X component decreased exponentially from 40g then was ranging between -10g and 10g, Z between 70g and 90g, and Y between -10g and 10g.

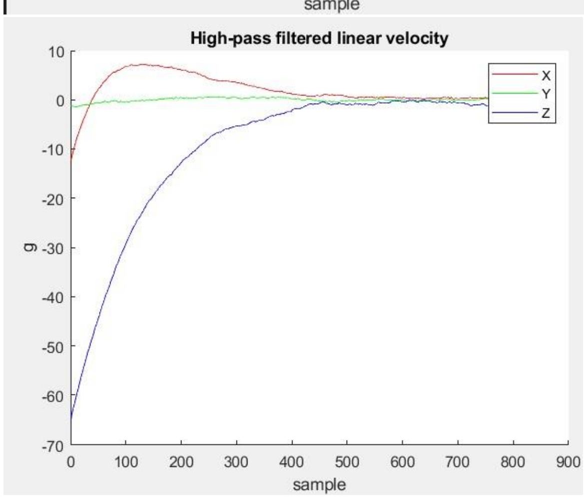


Figure 33: high pass filtered linear velocity vs sample graph

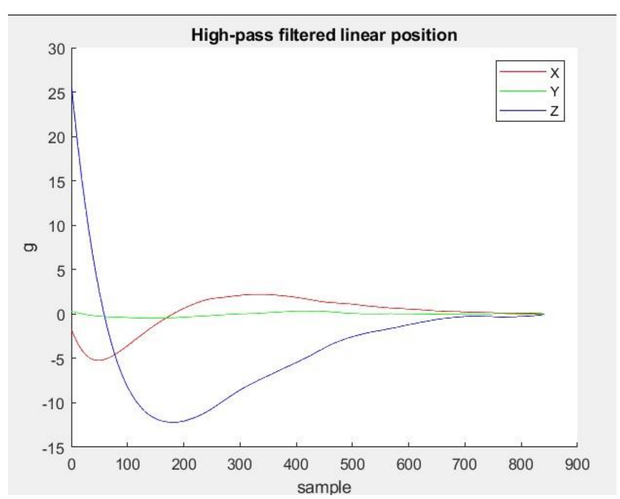


Figure 32:: high pass filtered linear position vs sample graph

As shown aside the Linear position VS sample graph (Figure 33) the X component decreased a bit and then increased to remain constant at 20g, Z decreased from 0g -100g and remained constant at that value, and Y remained constant at 0g.

As shown aside the Linear velocity VS sample graph (Figure 34) the X component increased to a constant value of 50g, Z increased linearly from 0g to 1000g, and Y remained constant at 0g.

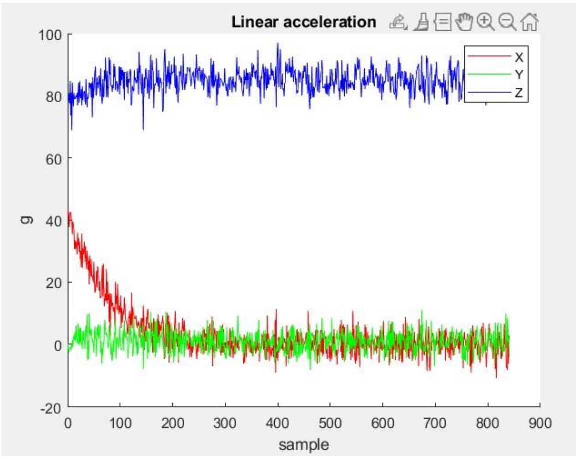


Figure 34: Linear acceleration VS sample graph

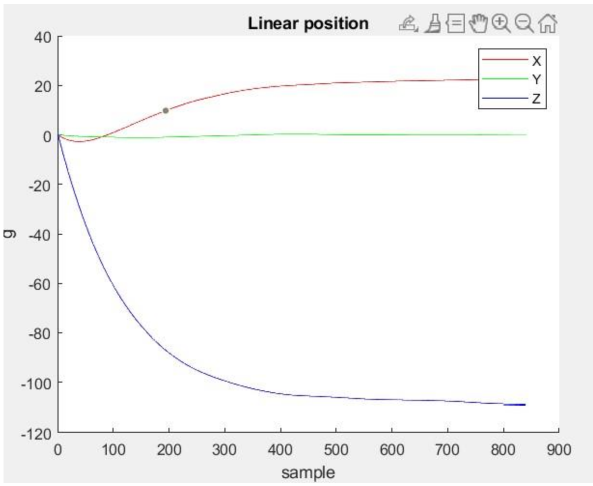
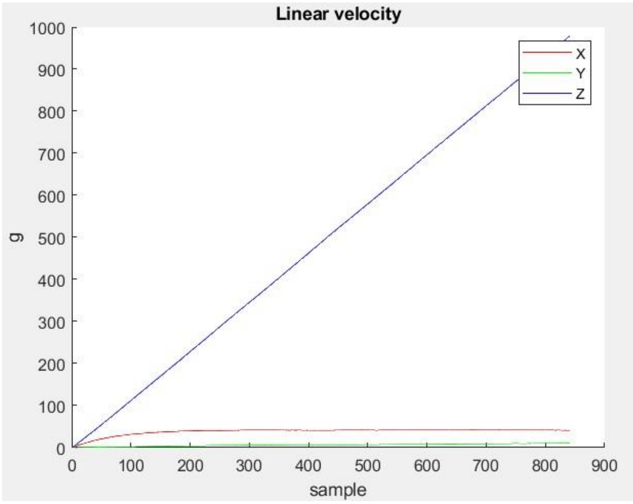


Figure 35: Linear position VS sample graph

Figure 36: Linear velocity VS sample graph



As shown aside the tilt compensated accelerometer VS sample graph (Figure 35) the X component decreased exponentially from 4g then was ranging between 0g and 1g, Z between 8g and 11g, and Y between -1g and 1g. (like the Linear acceleration VS sample graph (Figure 32)

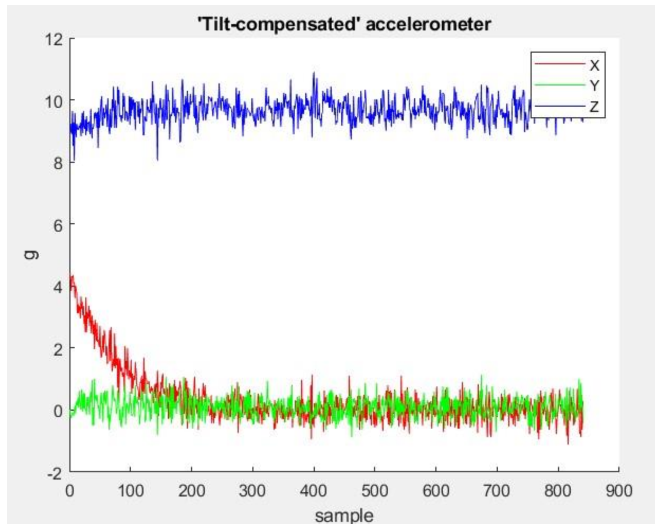


Figure 37: tilt compensated accelerometer VS sample graph

The unfiltered XYZ curve shows the position calibration by offsets as X, Y, Z components and as U, V, W vectors as shown on the curve (figure 36)

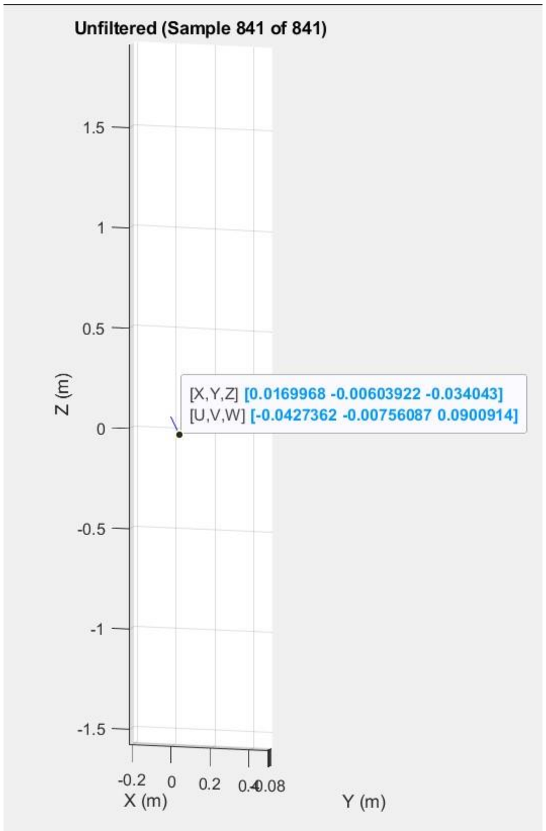


Figure 38: unfiltered XYZ curve

Possible Amelioration

Possible Amelioration for the future could be:

- Reduction in rover's mass; the rover is almost 15kg and is heavy for the user to lift considering the transfer from the ground to the solar panel. Such improvement could occur by using a lightweight battery and drone like material (light in weight), but such improvements are a bit expensive.
- Making the wheels faster; to stop slippage, the rover was moving along a certain speed, thus, to finish cleaning the panels, it will take enough time to clean.
- Adding sensors to the tank to indicate when it is empty, such feature will have to override the rover's motion to stop the action and return to the initial position for the user to fill, but it requires a GPS to return to the position it was and continue from where it was stopped.
- Using a camera to enhance cleaning. Consider a spot which was not cleaned by the rover by just passing above it once, a camera enables the rover to make sure the panel is clean and will clean the dirty spots left.

- Adding grip so that the rover can navigate along inclined planes with a higher angle without slipping.

References

Circuit Diagram - A Circuit Diagram Maker (circuit-diagram.org)

MIT App Inventor

ElectroSLab – Electroslab

Home | Electronics Katrangi Trading (katranji.com)