Miami University

College of Liberal Arts and Applied Sciences

Department of Engineering Technology

ENT 497/498 - Senior Design Project

GOFR (Guided Object Fetching Robot) /ˈgōfər/

Final Report

Prepared for:

Dr. Mert Bal

Prepared by:

Francis Nkrumah

Isaiah Storey

Eli Westbay

May 8th, 2020

Statement of Purpose

This is a reflection on what the GOFR group has accomplished in the pursuit of making an Autonomously Guided Vehicle (AGV). This document will lay out the steps and research performed in order to design and build a functional AGV that improved upon a previous group's attempt. The importance of this project is that the GOFR will be able to receive and deliver pallets autonomously. The expectation is that the GOFR will yield higher efficiency than current processes. Using Robot Operating System (ROS), Lidar technology for Localization, linear actuators,a Pixy2 Object-tracking camera for pallet locating and docking, and mecanum wheels for omnidirectional movement, an AGV capable of delivering parts to and from different robotic cells is created for the Phelp's Hall Robotics Lab at Miami University, Hamilton.

Table of Contents

Scope and Methodology

The lab in Phelps hall has many robotic cells, and because they are stationary robots, they are all isolated from each other. In order for these robots to work together, a robot that can move from station to station is required. A robot that can autonomously move to where it is needed without intervention (usually to pick up and deliver parts) is called an Automated Guided Vehicle or AGV. Although AGVs are already very prevalent in many industries, this AGV is custom designed to work with the Phelps Lab. In order to fulfill the design requirements of this project, a robot needed to be designed so that it could move to a location, pick up a pallet, and then deliver it to a designated location autonomously. The robot would need to be programmed so that it can receive data about its surroundings and a goal so that it could create a path to follow to its goal, It would need a way to handle the pallets, and it would have to be able to maneuver in tight spaces.

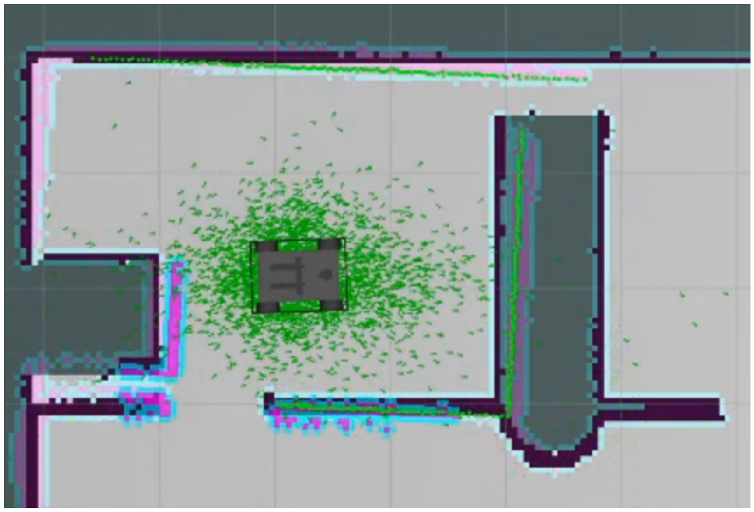
Software Development

To solve the automation requirement, ROS (Robot Operating System) was implemented instead of recreating the code for the robot's automation. ROS is an open source operating system that includes software components and libraries specifically designed for controlling robots.[3] ROS oversees all tasks and operations that are required by the AGV by communicating with the drive control systems, Sensing system, Localization systems, and Material Handling systems. Each of these systems are composed of nodes within a package. A node is an executable program that can gather information and send messages to other nodes.[3] The programs can be either C++ or Python. This adds to the compatibility of the operating system. Most of these systems will use multiple nodes to operate making it unwieldy to execute the nodes individually. The launch file takes care of this by batch loading these nodes along with any parameters required. It will even launch other launch files, allowing for a complex system that is maintained by ROS. An example of a launch file can be seen below. The various systems and their corresponding ROS packages will be explained next.

```
<launch>
  <arg name="multi_robot_name" default=""/>
  <arg name="set_lidar_frame_id" default="base_scan"/> <!-- base_laser -->

  <include file="$(find gofr_bringup)/launch/gofr_core.launch">
    <arg name="multi_robot_name" value="$(arg multi_robot_name)"/>
  </include>
  <include file="$(find gofr_bringup)/launch/gofr_lidar.launch">
    <arg name="set_frame_id" value="$(arg set_lidar_frame_id)"/>
  </include>
  <node pkg="gofr_bringup" type="odometry_publisher" name="odometry_publisher"/>
  <!-- <node pkg="gofr_bringup" type="gofr_diagnostics" name="gofr_diagnostics" output="screen"/> -->
</launch>
```

The drive control system controls the speed and direction of the AGV based on the information it receives from other systems. It runs a node called gofr_core that uses the arduino serial port to subscribe to the cmd_vel topic which publishes linear and rotational speeds. It then converts the speed data into motor signals and then publishes encoder readings to the encoder topic. A flow chart of the signals that are sent to the motors can be seen in Appendix A2. From there, a node converts the encoder data into odometry data. This can be seen in Appendix A3. This system is a part of the GOFR_bringup package in ROS. This package handles all the initializations for the GOFR and the LIDAR as well.

The Sensing system monitors all sensor data, like the ultrasonic sensors for side and rear detection object detection, and the sensors found on the material handling system. The Localization system is in control of scanning, mapping, and path planning. This system is a part of the gofr_navigation package. This package relies on multiple nodes in order to accomplish its task. One of the more important nodes is called move_base. This node is what connects the localization system to the Drive control system. This node takes in map data that was converted from scan data from the lidar and odometry data from the drive control system to create a path for the GOFR to follow when a location goal is created. Without this node, the GOFR would not be able to autonomously travel. An example of the map and lidar data can be seen below.



This is a map that was captured by the GOFR and it is what the Navigation system uses. The light gray space is considered unoccupied space whereas the black and dark gray areas are considered occupied and unknown spaces, respectively. The GOFR is represented by the gray rectangle which serves as a footprint for the software to use when calculating paths. The green arrows represent where the algorithm thinks the GOFR could be based on the current data. The GOFR is considered to be in the center of the cluster. The denser the arrow grouping, the more likely the GOFR is actually in that location. The blue and pink colors near the map's walls represent where the algorithm thinks the walls are. It is similar to a heat map where the pink is higher probability than the blue. This updates and becomes more accurate as the GOFR changes positions.

Finally, the material handling system monitors and sorts the job queue based on priority.The job request function is valuable to this project because this is what makes the GOFR versatile. Unfortunately due to Covid-19 this aspect was unable to be completed. This function was to be accessed from an app or website. Once the user sends a job request, a job-planning node within ROS would receive it, prioritize it and then store it until the path planning node requests the next job. Once this happens the job node will send a one-time message back, stating the new location. This type of message is called a service message, it is only sent when requested as this helps reduce the load on the processor.[3] In cases where a workstation is connected directly to the AGV via Bluetooth, the workstation would be able to request a job in a similar way as the webserver.

Once the GOFR receives the request and is at the destination, the Drive control system will switch over to docking mode. Docking mode does not use lidar, instead it will use the Pixycam to navigate. First the robot will adjust its height, using the main linear actuator and linear rails, so that it matches the dock’s height using a time of flight sensor pointing towards the ground. Then it will find the color coded targets and slowly move in keeping itself aligned with the target. Once the target is a certain distance away, it will correct itself for the final alignment. Then it will move forward, until the proximity sensor in front detects an object. At this point the forks are in the pallet. The vertical lift (item 2, Appendix A8) will then raise the forks until the pallet clears the base and then switches back into Long range mode to deliver the pallet.

Frame Design

The initial design of the GOFR closely mirrored the design of the previous team. The GOFR was to have two powered wheels in the rear with two casters in the front, much like Figure 1.

Figure 1

One big advantage to this design was ease of integration. Using this design would mean there would have been less work ahead in making a functional autonomously guided vehicle. This was very appealing at first, but the drawbacks outweigh the advantages. One of the drawbacks of the caster wheel design is complexity in movements. For a robot of this design to move to a point that is not directly in front of it or behind it, it must use a series of complex turns and maneuvering to reach its objective. For example, if the wheelbase in Figure 2 needs to travel to point C it must first traverse to point B, then to C.

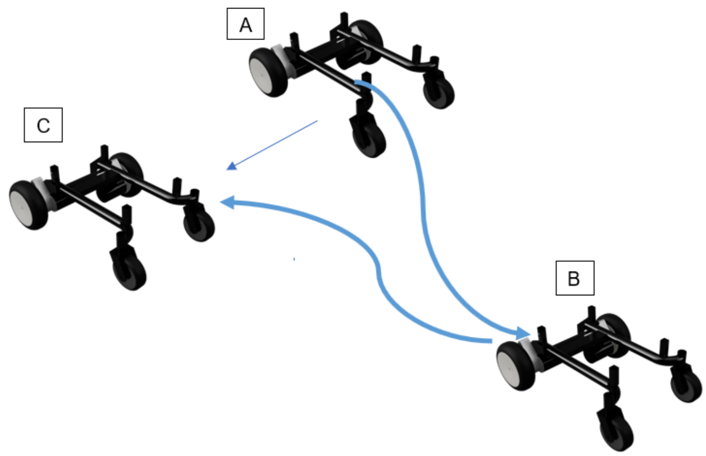


Figure 2

In most cases this would be fine, the robot would just need enough space to maneuver. Unfortunately, there are many spatial restraints in the area where the GOFR is to be implemented. In order to conform to the environment in which the GOFR was to operate it needed to move more efficiently.

After researching different drivetrains the mecanum wheel drivetrain was selected. The mecanum wheel is depicted in Figure 3.

Figure 3

In this case, each wheel consists of 12 individual rollers biased at 45° with respect to the endplates that sandwich the rolls. The four mecanum wheels are driven by four independent motors. This allows for omni-directional motion or motion in any direction as explained in Appendix A10. Instead of making several turns to get to the same point as the robot in Figure 2, the mecanum wheels would allow a robot to traverse directly to that point. Realizing this is the direction in which the GOFR should go, drastically changed the initial design. The mecanum drive train requires four independently driven mecanum wheels, therefore four motors. The initial design of the GOFR (Figure 1) could not support the additional motors that were necessary to implement the mecanum wheel design.

One solution to deal with this problem was to cut off the caster wheels and weld laser cut motor mounting plates to the frame. These custom plates would have to be identical to the existing motor mounts in order to effectively support the motors. The plates would also have to be held square on to the frame within 1° to ensure the frame was level once the motors and wheels were attached. This option was bypassed for a more elegant solution: using a duplicate wheelchair frame. A duplicate frame was cheaper than having custom plates made. It was also much easier to line up for welding to ensure everything was level. The duplicate frame also came with peace of mind, knowing that the additional motors were made to fit on the frame. The caster wheels were cut off each frame, along with a steel battery tray from the new duplicate frame. This created two identical frame halves (Figure 4).

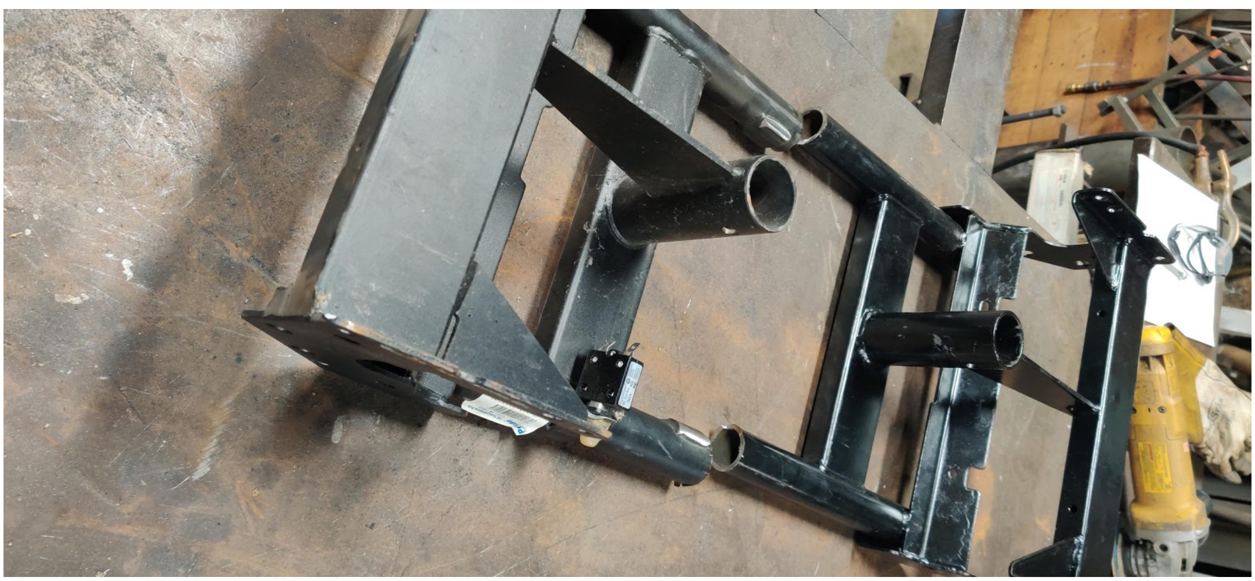


Figure 4

Steel tubes, that were machined to fit tightly inside each frame half, were inserted into the open ends. This was to give the joint rigidity and to ensure everything was aligned properly. The two halves were then welded together (Figure 8).

Figure 5

The battery tray was salvaged and modified to make a new component tray (for storage of motor controllers, power converters, and other electrical components). This was then welded parallel to the cross beams as seen in Figure 6.

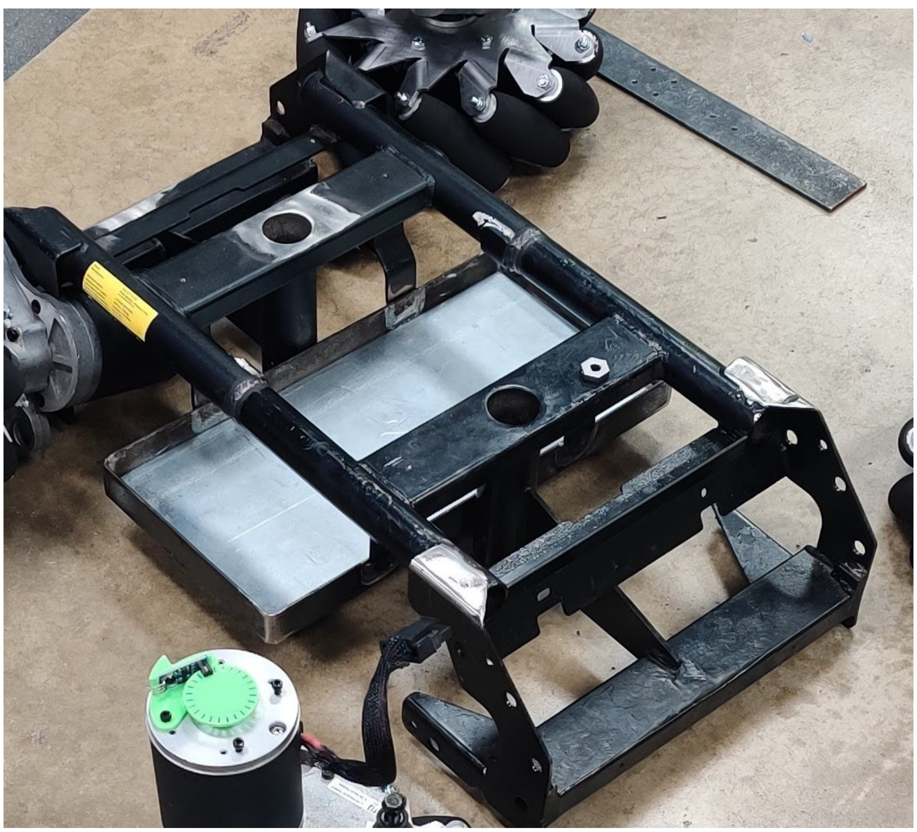


Figure 6

The frame consists mostly of round tubing as seen in Figure 6. This makes it difficult to mount anything directly to the top of the base because it lacks flat surfaces. A space to mount the Lidar along with the mini pc was also needed. This gave way to the designing of the GOFR's mounting frame. The mounting frame makes use of two lengths of steel angle that rest on top of the base frame's motor mounts as seen in Figure 7a and 7b.

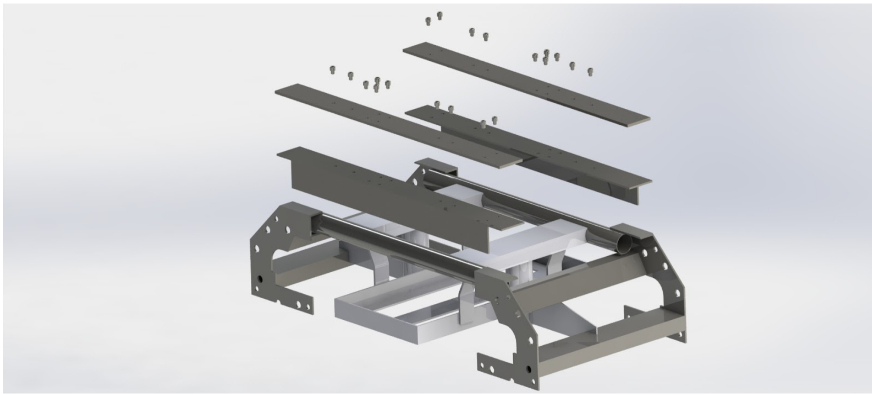


Figure 7a

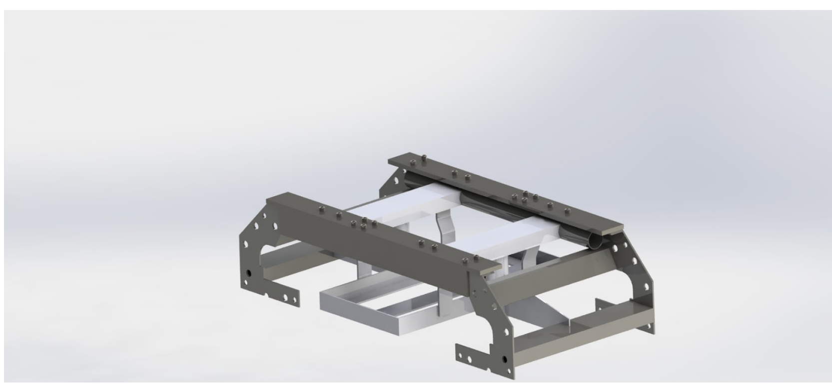


Figure 7b

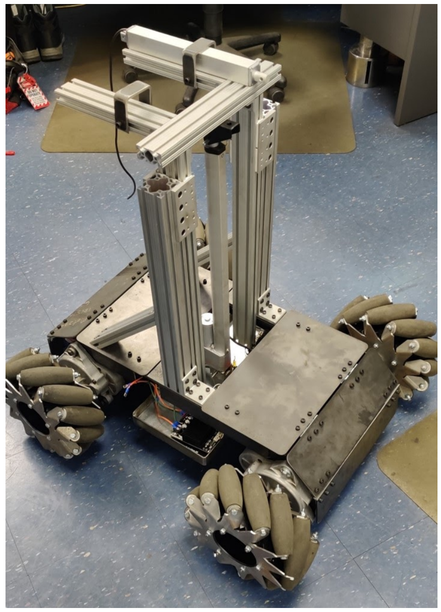


Figure 7c

On top of each steel angle was a steel plate of the same length. The angle and runner plate were drilled and tapped together. This was to allow for enough threads for the bolts to engage in when assembled. Two 14” X 8” steel plates were then bolted to the runner plate and steel angle. These two plates would act as surfaces to which the lidar and mini PC can mount. Two more 14” x 8” plates were cut in half and these plates were connected to the other 14” plates via a hinge system creating compartments in the front and rear of the GOFR as seen in Figure 7c. The mounting frame was assembled such that the steel angle squeezes tightly against the base frame when assembled, holding the mounting frame in place.

Our team decided to purchase materials that are quality, strong, rigid, and reasonably priced to build the AGV. It was also imperative to use materials that would make the finished product look good and well designed. After intensive research, the team agreed to use 80/20 aluminum extrusion to construct the lifting assembly. This extrusion has sleek lines with a professional look all while being rigid enough to trust. This extrusion is very flexible. It can be easily assembled without welding. We realized that using 80/20 extrusion would also make it easier for future students who might modify the AGV, because of the ease and flexibility it presents.

Atop the mounting frame sits an aluminum extrusion assembly. This assembly acts as a forklift mast. In Figure 7c the stationary stanchions are mounted directly to the mounting frame along with the support angle. The linear bearings are attached to the stationary stanchions. These linear bearings guide the mobile stanchion assembly as it travels up and down. Two linear actuators are affixed on top of this assembly acting as retractable forks.

Expected Findings

Initially the plan was to work as one group with the CNC router team. After much consideration, it was agreed that the scope of the GOFR could and should be much wider. It has the ability to be used as a shop-wide AGV with multiple waypoints, to service more working centers within the lab. The GOFR was designed to assist the other team's project along with all of the robots in the lab. A special attachment has been designed so that the AGV can carry a piece of plywood to be delivered to the CNC router. Unfortunately due to the circumstances surrounding Covid-19, the collaboration with the other team was unable to continue.

While planning, it was assumed that the wiring and some of the code from the team that built the first AGV could be reused, but there was very little documentation for it so this had to be designed from scratch using the wheelchair base of the last team's project and building up from there. The previous team purchased a wheelchair frame, not unlike Figure 1. The wheelchair was stripped down and built upon as seen in Figure 2. With the intention to only use the steel frame and motors from the original design. With little to go from with the previous group's iteration of the AGV a decision was made to restart the research and design instead of trying to decipher their work.

Figure 8

Figure 9

Navigation Research

The starting point for research was with navigation. There is plenty of literature on AGV routing [2] and mapping once it was established how the AGV would effectively "see". Fortunately a Slamtec RPLIDAR A1 (Figure 3) was found on a previous project that could be used on the GOFR. One source describes Light detection and ranging (Lidar) as a blind person with an excellent memory feeling their way around a room with an extending stick. They are able to build a map in their head of what the room looks like and what obstacles are around the room by judging how much of a stick length each object is distanced from them.[8] In this analogy the lidar system is the blind person and the laser that the lidar system emits is analogous to the stick. The lidar sends out pulses of light as it rotates, the pulses get sent back to it and then the system can calculate how far away an object is from itself based on how long the pulse takes to get back to the sensor.

Figure 10

Given the current design (Appendix A4) however lidar will not be sufficient in pedestrian or unplanned object avoidance when the lidar "sight" is blocked in certain directions. Ultrasonic and Infra-red (IR) sensors were used to supplement the lidar and mitigate blindspots.

Ordinarily this limited field of view would not present a problem because most AGVs travel forward and backwards only with large turning radii. The design however requires nearly 360° of coverage because the GOFR would be omnidirectional. This was achieved with the mecanum wheel(Figure 3). These wheels are designed such that when four are in the orientation shown in Appendix A10, varying each wheel speed or direction leads to forces being canceled out which allows for motion in any direction.

For example when all wheels are rotating in the same direction the forces along the x-axis are cancelled out by one another propelling the vehicle forward (or backward given your frame of reference). Each wheel is independently operated by a different motor so varying the directions of each wheel yields different directions. Initially the completely assembled wheels were going to be purchased, but pricing for the size required seemed to be overly expensive for the application, so it was decided that the wheels would be made inhouse. The prospect of building our own mecanum wheels came with many variables. A decision needed to be made between whether to purchase end plates, Figure 3, (they would be steel if purchased) or 3d print them. In the end, the plates were drawn up and sent out to be laser cut and formed (Appendix A11). The type of rubber to use was also researched based on a rubber compound guide[5]. This resulted in 60A durometer SBR to be used for the tread, given that this is the rubber that is widely used in the tire industry. SBR is known to have good traction properties along with great wear and resilience properties as can be seen in Appendix A9. Pinnacle Roller Company opened their shop to the GOFR team. And allowed for the production of the mecanum wheels.

Mecanum Wheel Production

Pinnacle had nearly one hundred scrap rolls they were willing to donate as seen in Figure11. In the spirit of reusing materials and being environmentally conscious, the GOFR team made the decision to modify these rolls.

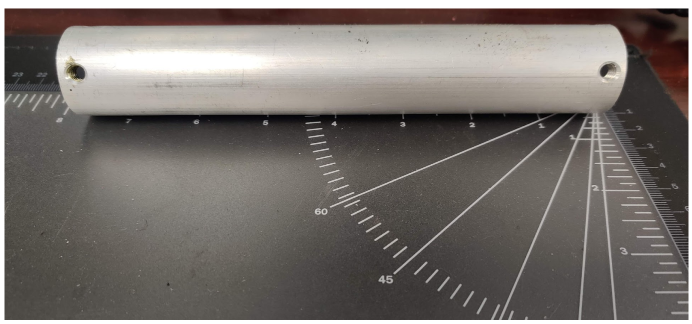


Figure 11

In total 72 rubber covered rolls were produced. This yielded enough rolls to produce five mecanum wheels with twelve extra replacement rolls. Using a cold saw, one inch was removed from one side of each roll. This process removed extra length and one pair of unneeded set-screw holes. Each roll was then chucked on in a manual lathe with the other set of set screw holes toward the head stock. The rolls were then turned down, producing the roll shown in Figure 12. Each roll was then flipped around in the lathe and faced to the desired length. The rolls were then given a rough surface finish with a corded angle grinder. A thermally activated chemical adhesive was then applied to each roll (Figure 13). At this point, lengths of uncured 60A durometer SBR rubber were cut to size and rolled on to each roll (Figure 14), building each roll well over the desired finish diameter. These rolls were then vulcanized in a pressurized autoclave at 300°F (Figure 15 shows the autoclave).This vulcanization process cures the rubber into one homogeneous piece and activates the chemical cement, adhering the rubber to the aluminum core.


Figure 12


Figure 13

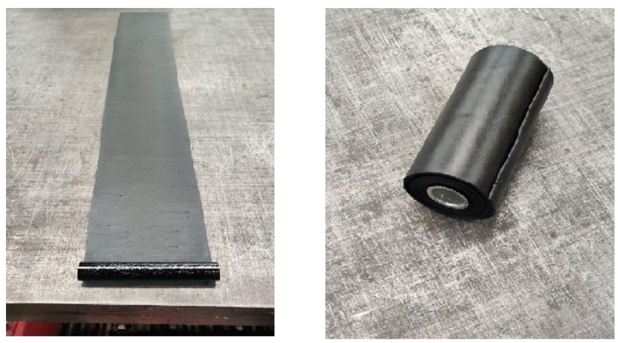


Figure 14

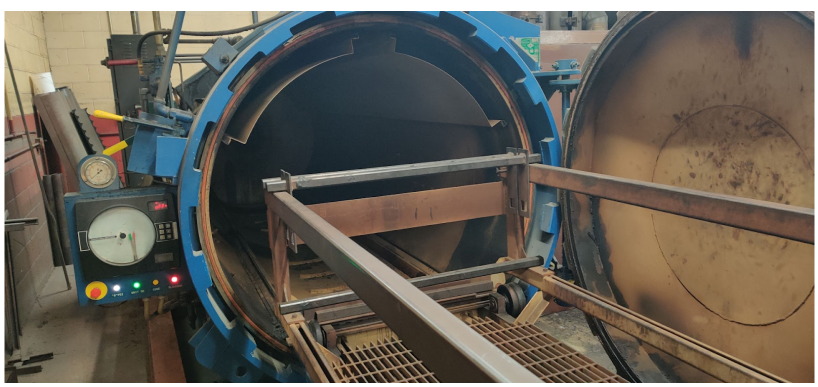


Figure 15

After the 8 hour vulcanization process is complete the rolls must cool down. The rolls are then ground in a CNC lathe with a grinding wheel attachment. This step removes the excess material and yields the finished part. The rolls must be ground in a CNC lathe to produce the crown that each roll requires. Figures 16 shows the grinding process. After grinding each roll the mecanum wheels were assembled as seen in Figure 3.

Figure 16

Results

Due to the events from Covid-19, the results from the project are slightly skewed. The GOFR team was unable to meet and integrate several physical components necessary to complete the project. In spite of this, there were many avenues of technology and problem solving explored.

Most of the testing and programming took place on a smaller robot with mecanum wheels, seen in Figure 17. While this method was effective when working away from the lab, integration and testing of ROS with the GOFR was not possible before the deadline. However the miniature robot turned out to be a success. Autonomous motion as well as mapping of an entire room was achieved.

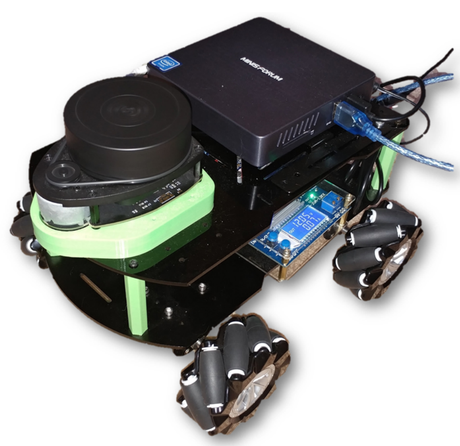


Figure 17

The implementation of the mecanum wheel drive train on the GOFR itself was also successful. A manual version of the GOFR was completed. Initially the motors were allowed to operate at full speed. Without testing in this manual mode, a full speed autonomous GOFR would have been dangerous. The four motor drive allowed the GOFR to reach very high speeds and in some instances it became unpredictable. This manual mode allowed the GOFR team to determine safe operating speeds and to clamp the motors' speed down, well out of the danger zone. These precautions ensured that the GOFR is ready to go fully autonomous.

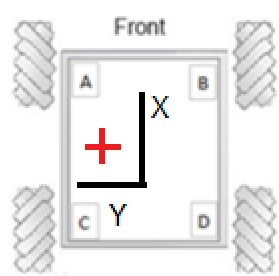
In the end the GOFR was tested via remote control, making full use of it's omnidirectional capabilities as well as operating the linear actuators. In order to make full use of the omnidirectional capabilities using a remote control, it was necessary to implement several equations. These equations (Equation1) are what drove the speeds and directions of each wheel. For example in order to move the GOFR at a 45° angle (with respect to horizontal) while staying forward-facing, it is necessary to go through each equation. Since the GOFR is to remain forward facing, *W* will be 0 for each equation. A 45° angle is achieved in this frame when X is positive, Y is negative and both are equal in magnitude. So if an arbitrary speed is chosen, say 10m/s, Wheels B and C will not move. Wheels A and D however will both move at 20m/s. This will propel the GOFR forward-facing at a 45° angle.

$$Wheel_A = X - Y - W \qquad Wheel_B = X + Y + W$$

$$Wheel_C = X + Y - W \qquad Wheel_D = X - Y + W$$

where X is forward speed, Y is Lateral speed, and W is the angular velocity

Equation 1



Conclusion and Recommendation for Further Study

While the GOFR was not able to be completed before the deadline, many milestones that went into designing and building it were achieved. That being said, the majority of tasks shown in the Gantt chart below were completed. This project allowed for a deeper view into robot automation with ROS, a more indepth perspective on the advantages of omnipositional drive systems, like mecanum wheels, and how to design them, and it has created many new opportunities for future students to learn.

The GOFR has tremendous potential for further study. Below are some areas that may require it:

Fork Development:

The GOFR was designed to pick up a pallet specially designed to fit on its forks, but since the forks have the potential to independently move they could be used as another degree of freedom if the correct linkage were installed onto it. This could open the door to many more applications.

Status implementation:

While the GOFR uses autonomous navigation, a way for it to keep track of certain statuses was not implemented. This would greatly improve the user experience by letting them know any vital information.

Job handling:

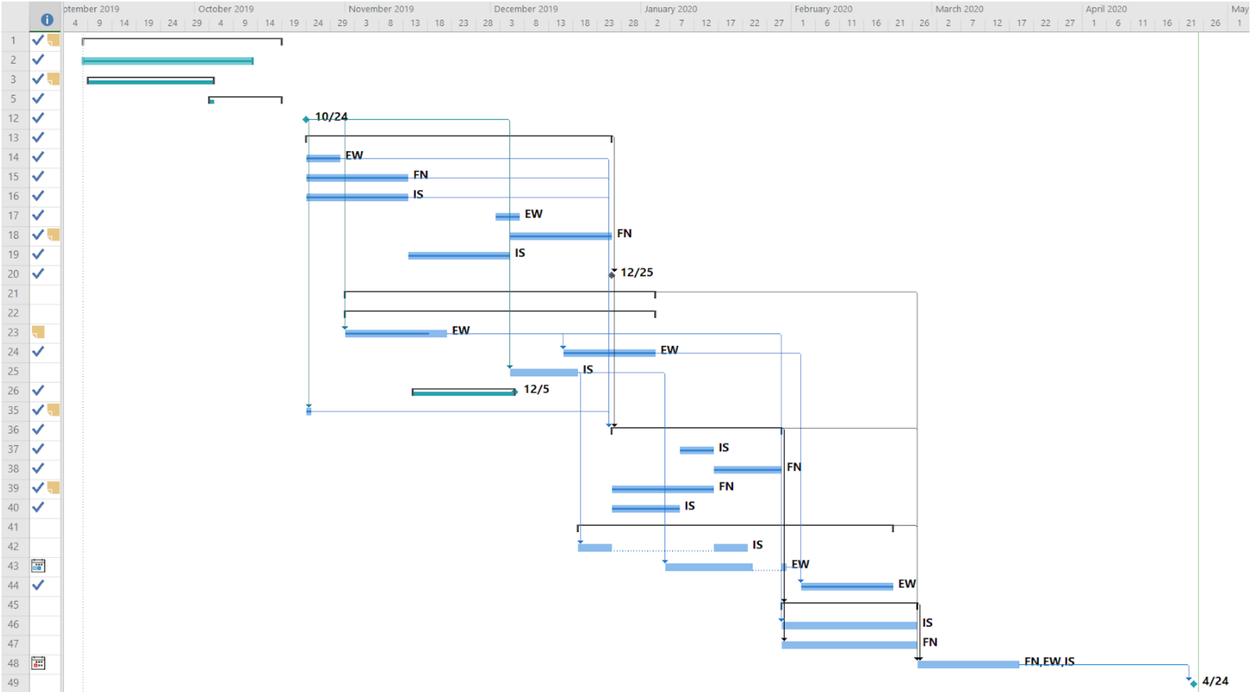
This was one item described above that could not be completed. This item will be a necessary part in bringing the GOFR out of the concept phase and into practical use in Phelps Hall.

The source code for the GOFR can be found at the following address:

https://github.com/TeamGOFR/GOFR

A 3d model for the GOFR can be found at the following address:
https://a360.co/3baEIlJ

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Assigned
1		**Planning/Proposal**	**30 days**	**Sun 9/8/19**	**Fri 10/18/1!**		
2		Form Group/Determine Pro	27 days	Sun 9/8/19	Sat 10/12/1		
3		**Find requirements**	**4 wks**	**Mon 9/9/1!**	**Fri 10/4/19**		
5		**Submit Proposal**	**11 days**	**Fri 10/4/19**	**Fri 10/18/1!**		
12		Proposal Returned	0 days	Thu 10/24/1	Thu 10/24/1		
13		**Design**	**45 days**	**Thu 10/24/**	**Wed 12/25,**		
14		Design Robot	1 wk	Thu 10/24/1	Wed 10/30/		EW
15		Design Frame	3 wks	Thu 10/24/1	Wed 11/13/		FN
16		Design Forks	3 wks	Thu 10/24/1	Wed 11/13/		IS
17		Pallets	5 days	Mon 12/2/1	Fri 12/6/19		EW
18		Design Loading/unloading	3 wks	Thu 12/5/1!	Wed 12/25/		FN
19		Electrical	3 wks	Thu 11/14/1	Wed 12/4/1		IS
20		Design Approved	0 days	Wed 12/25/	Wed 12/25/	13	
21		**Software**	**46 days?**	**Fri 11/1/19**	**Fri 1/3/20**		
22		**Navigation**	**46 days**	**Fri 11/1/19**	**Fri 1/3/20**		
23		Short Range	3 wks	Fri 11/1/19	Thu 11/21/1	12	EW
24		Long Range	3 wks	Mon 12/16/	Fri 1/3/20	23	EW
25		Webpage for Adding tasks	2 wks?	Thu 12/5/1!	Wed 12/18/	12	IS
26		**Oral Presentation**	**3 wks**	**Fri 11/15/1!**	**Thu 12/5/1**		
35		Test Previous Robot	1 day	Thu 10/24/1	Thu 10/24/1	12	
36		**Build**	**25 days?**	**Thu 12/26/**	**Wed 1/29/:**	**35,20,14,15,16**	
37		Wiring	1 wk	Thu 1/9/20	Wed 1/15/2		IS
38		Elec. Components Tray	2 wks	Thu 1/16/2(	Wed 1/29/2		FN
39		Frame	3 wks	Thu 12/26/1	Wed 1/15/2		FN
40		Fork	2 wks?	Thu 12/26/1	Wed 1/8/2C		IS
41		**Software Integration**	**47 days**	**Thu 12/19/**	**Fri 2/21/20**		
42		Task Entry	2 wks	Thu 12/19/1	Wed 1/22/2	25	IS
43		Task Planning	3 wks	Mon 1/6/2C	Thu 1/30/2(	25	EW
44		Room Mapping	3 wks	Mon 2/3/2C	Fri 2/21/20	43,24	EW
45		**Mechanical Integeration**	**20 days**	**Thu 1/30/2**	**Wed 2/26/:**	**36**	
46		Docking	4 wks	Thu 1/30/2(	Wed 2/26/2	23	IS
47		Pallet Moving	4 wks	Thu 1/30/2(	Wed 2/26/2	36	FN
48		Testing	3 wks	Thu 2/27/2(	Wed 3/18/2	36,21,41,45	FN, EW, IS
49		Final Presentations	0 days	Fri 4/24/20	Fri 4/24/20	48	


September 2019
October 2019
November 2019
December 2019
January 2020
February 2020
March 2020
April 2020
May
10/24
EW
FN
IS
EW
FN
IS
12/25
EW
EW
IS
12/5
IS
FN
FN
IS
IS
EW
EW
IS
FN
FN,EW,IS
4/24

References

[1] Fazlollahtabar, H and Saidi-Mehrabad, M, 2015, *Autonomous Guided Vehicles : Methods and Models for Optimal Path Planning, Springer, Boston, MA.* http://rave.ohiolink.edu.proxy.lib.miamioh.edu/ebooks/ebc/9783319147475

[2] Fahimi, Farbod, 2009, Autonomous Robots : Modeling, Path Planning, and Control, *Springer, Boston, MA.* http://rave.ohiolink.edu.proxy.lib.miamioh.edu/ebooks/ebc/9780387095387

[3] Matsinos, Evangelos, 2012, Modelling of the motion of a Mecanum-wheeled vehicle., Institute of Mechatronic Systems, Zurich University of Applied Sciences

[4] Pyo, Y, Cho, H, Jung, R, and Lim, T, 2017, *ROS Robot Programming*. 1st ed., ROBOTIS Co.,Ltd, Seoul

[5] “Rubber Materials & Elastomer Compound Information,” *Rubber Materials, Rubber Material Selection Guide, Rubber Elastomers*. [Online]. Available: http://usa.datwyler.com/materials.html. [Accessed: 05-Nov-2019].

[6] “US4996468A - Automated guided vehicle,” *Google Patents*. [Online]. Available: https://patents.google.com/patent/US4996468A/en. [Accessed: 10-Nov-2019].

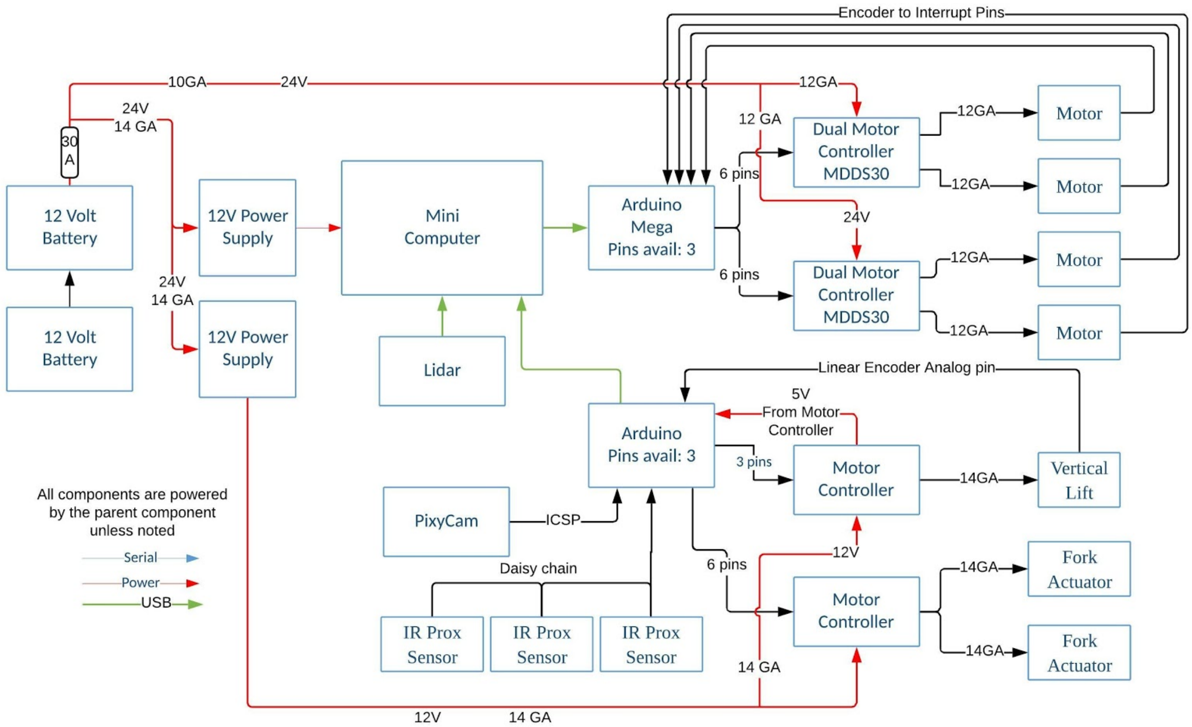
[7] Y. Li, S. Dai, Y. Zheng, F. Tian, and X. Yan, “Modeling and Kinematics Simulation of a Mecanum Wheel Platform in RecurDyn,” *Journal of Robotics*, vol. 2018, pp. 1–7, 2018.

[8] R. T. Staff, “What is Lidar and How Does it Help Robots See?,” *Robotics Business Review*, 10-Oct-2019. [Online]. Available: https://www.roboticsbusinessreview.com/rbr/what_is_lidar_and_how_does_it_help_robots_see/. [Accessed: 05-Oct-2019].

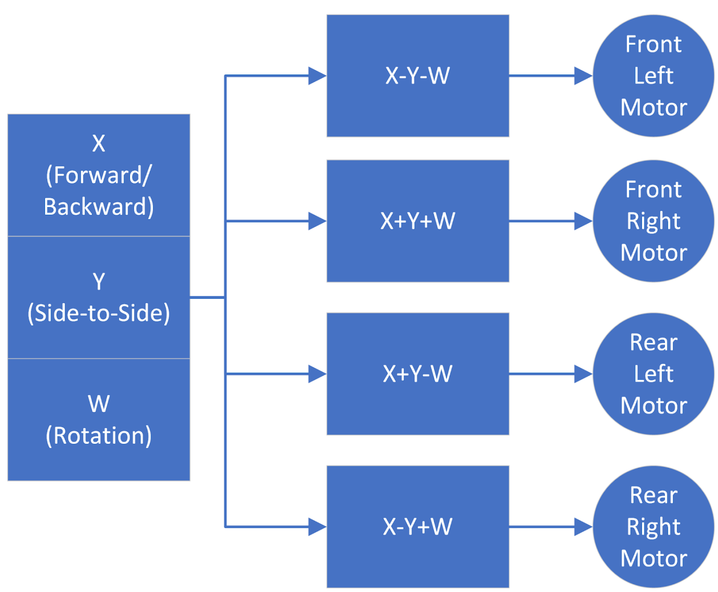
[9] L. Koval, J. Vaňuš, and P. Bilík, “Distance Measuring by Ultrasonic Sensor,” *IFAC-PapersOnLine*, 23-Dec-2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896316326623. [Accessed: 01-Dec-2019].

[10] *Driving Mecanum Wheels Omnidirectional Robots*. [Online]. Available: https://www.roboteq.com/index.php/applications/applications-blog/entry/driving-mecanum-wheels-omnidirectional-robots. [Accessed: 01-Dec-2019].

Appendix



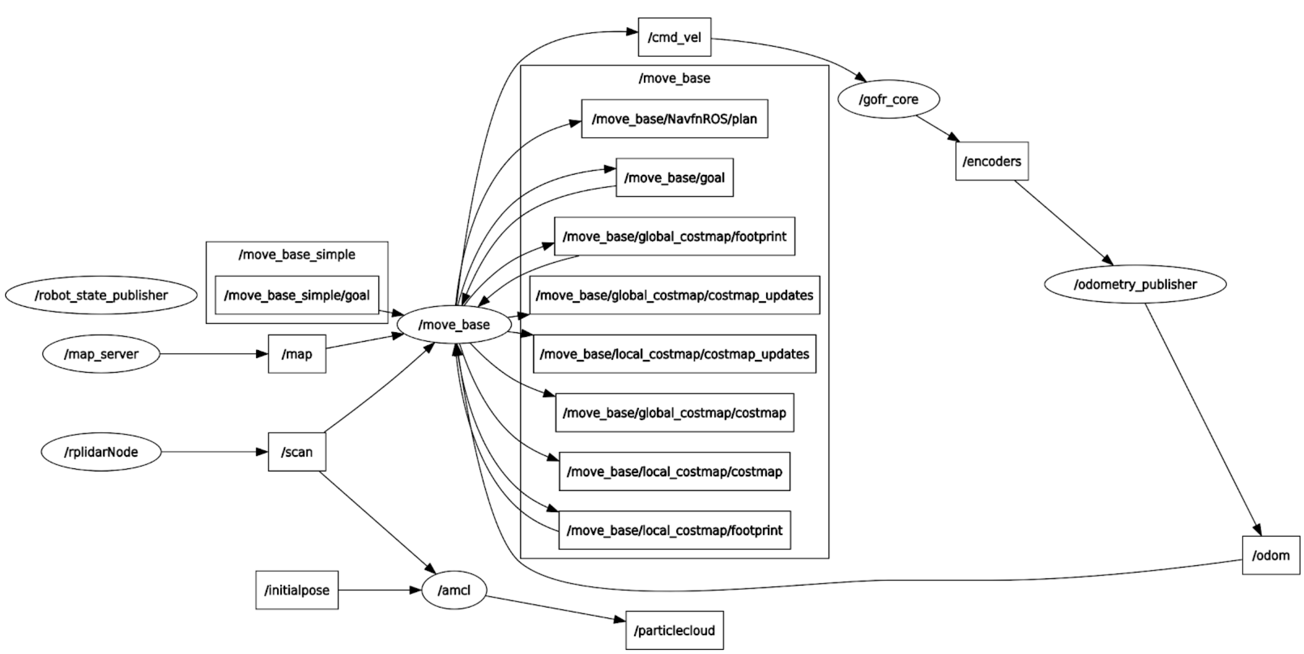
Appendix A1



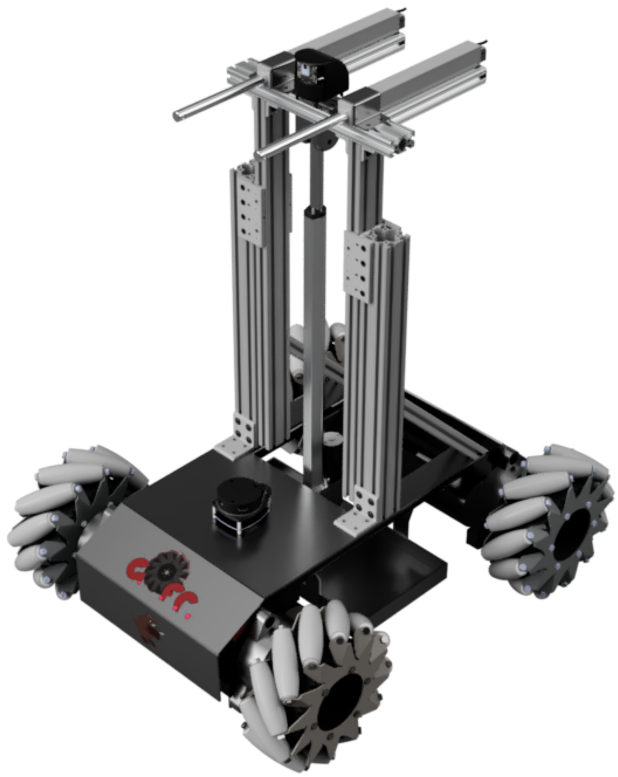
Appendix A2a
Flowchart of motor signal conversion (W = xy)

```
void controller_cmd_vel(byte x, byte y, byte xy)

{
   // normalize
  int x2 = x - MAXBITS/2;
  int y2 = y - MAXBITS/2;
  int xy2 = xy - MAXBITS/2;
  int sum = abs(x2) + abs(y2) + abs(xy2);

  if(sum > (MAXBITS/2)-1) {
    x2 = (x2 * MAXBITS / sum);
    y2 = (y2 * MAXBITS / sum);
    xy2 = (xy2 * MAXBITS / sum);
  }
  int pwma =x2 - y2 - xy2;
  int pwmb =x2 + y2 + xy2;
  int pwmc =x2 + y2 - xy2;
  int pwmd = x2 - y2 + xy2;
  DRIVE(pwma,pwmb,pwmc,pwmd);
}
void DRIVE (int pwmA, int pwmB, int pwmC, int pwmD) {

  motorA.drive(pwmA);
  motorB.drive(pwmB);
  motorC.drive(pwmC);
  motorD.drive(pwmD);
}
```

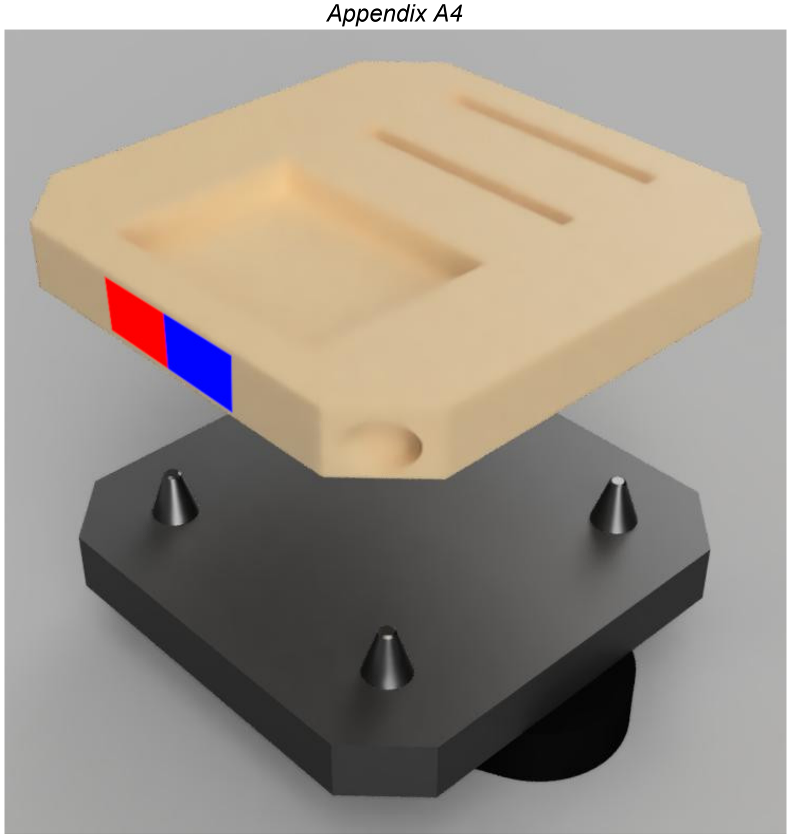
Appendix A2b
Arduino code of motor signal conversion (W = xy)



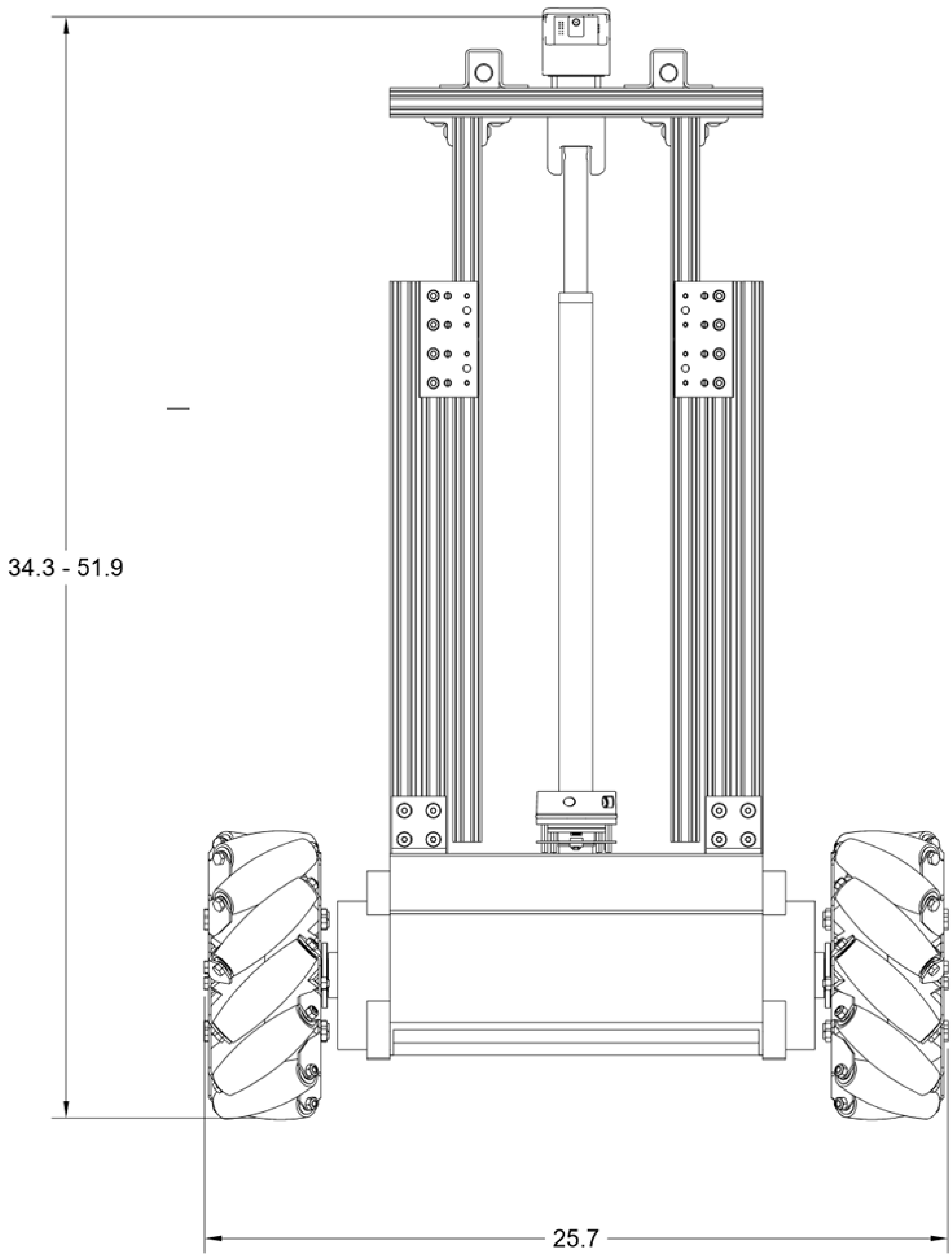
Appendix A3



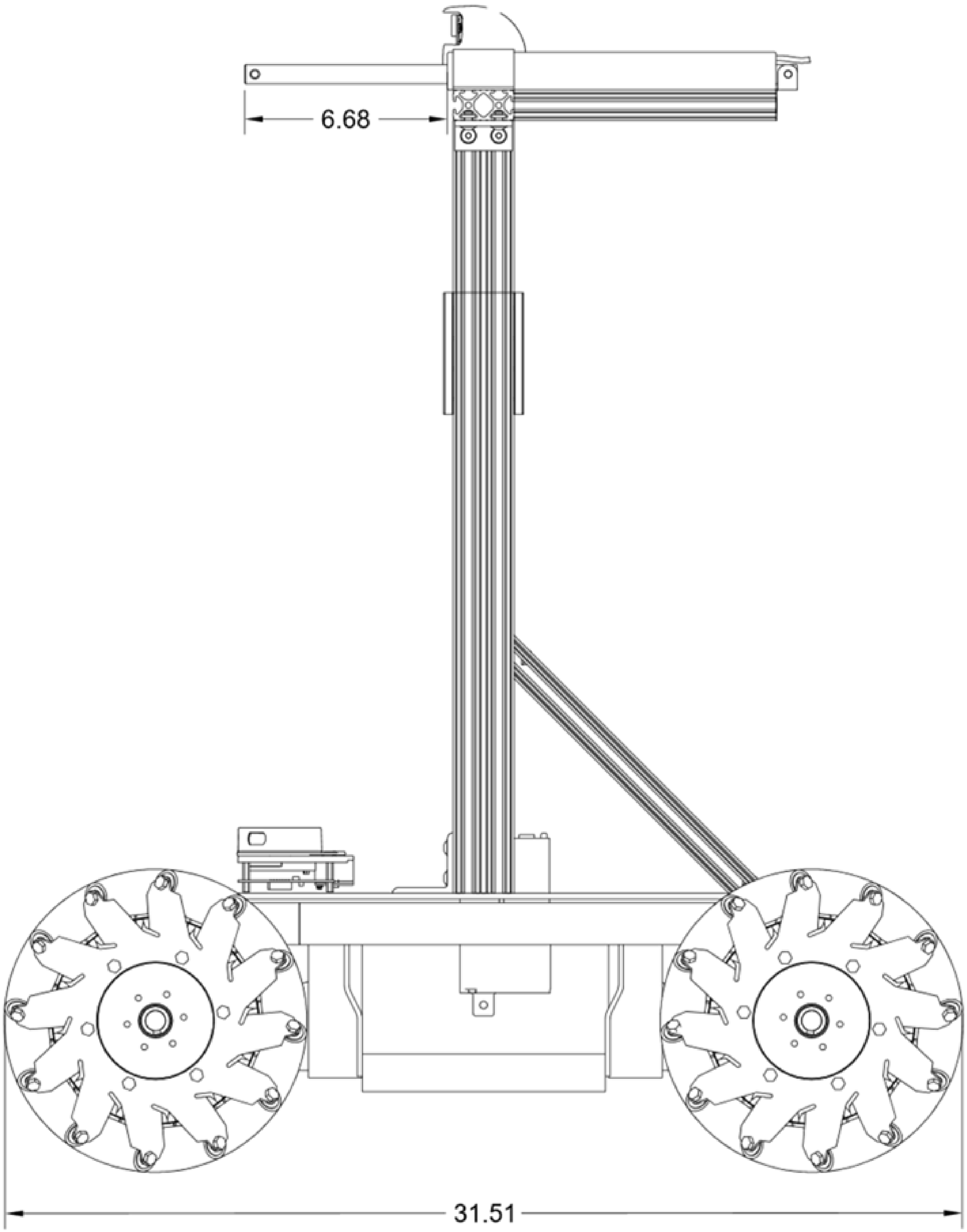
Appendix A4



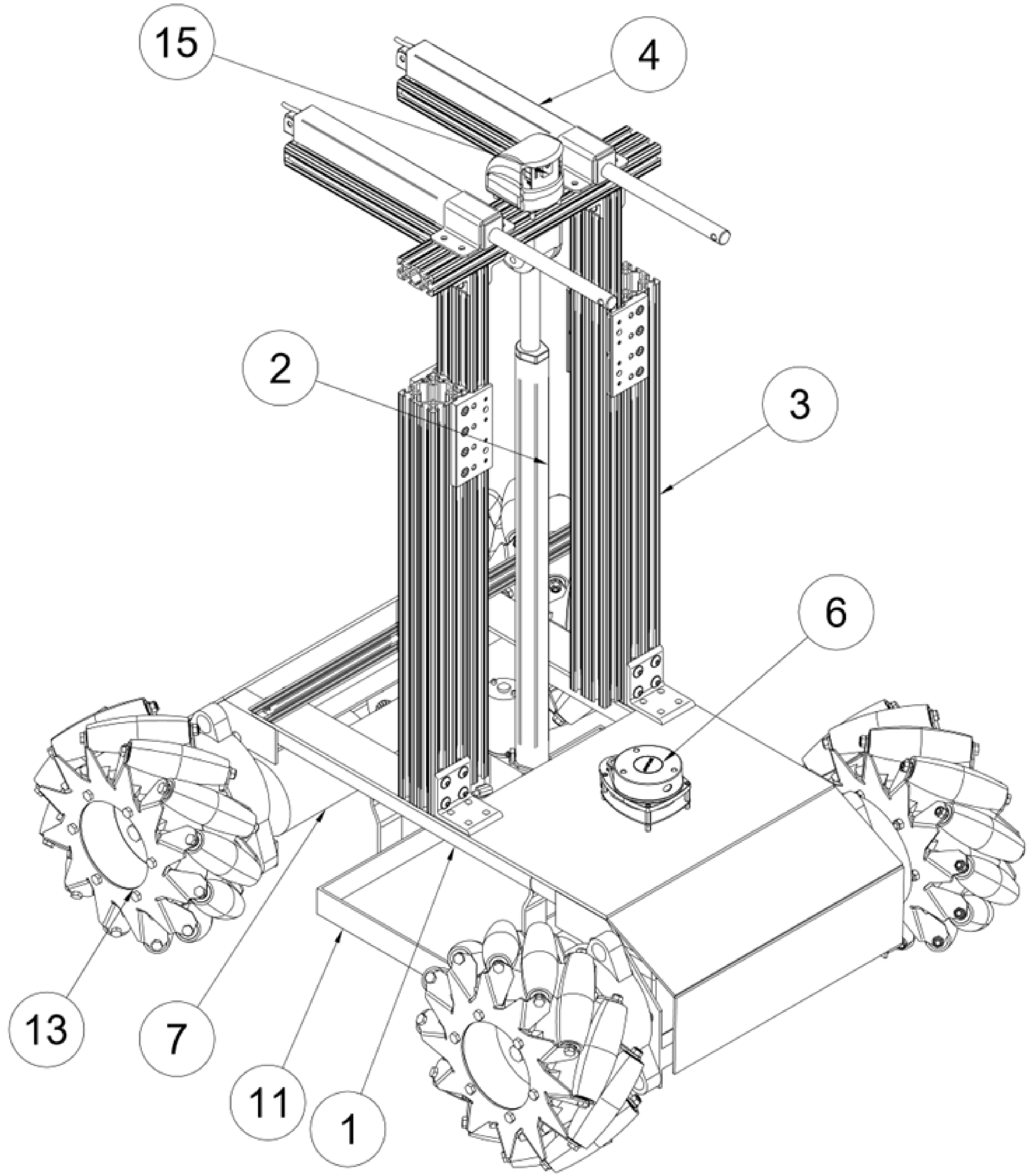
Appendix A5
Representation of pallet



Appendix A6
Front view of the GOFR



Appendix A7
Side view of the GOFR



Item 1	*GOFR Frame*		*Item 6*	*Rplidar A3 Lidar*
Item 2	*18" Vertical Lift, 12V, Potentiometer Encoder*		*Item 11*	*Component Tray*
Item 3	*80/20 Linear Slide Assembly*		*Item 13*	*Mecanum Wheels, 10" Diameter, 4" Width*
Item 4	*Pallet Forks, 6" Linear Actuator, 12V input*		*Item 15*	*Pixy2 Object Recognition Camera*

Appendix A8 Isometric View of the GOFR with List of main parts

ELASTOMER RUBBER COMPOUNDS TYPES AND REFERENCES					
General Description	**Chemical Description**	**Abbreviation (ASTM 1418)**	**ISO/DIN 1629**	**Other Trade names & Abbreviations**	**ASTM D2000 Designations**
Nitrile	Acrylonitrile-butadiene rubber	NBR	NBR	Buna-N	BF, BG, BK, CH
Hydrogenated Nitrile	Hydrogenated Acrylonitrile-butadiene rubber	HNBR	(HNBR)	HNBR	DH
Ethylene-Propylene	Ethylene propylene diene rubber	EPDM	EPDM	EP, EPT, EPR	BA, CA, DA
Fluorocarbon	Fluorocarbon Rubber	FKM	FPM	Viton ®, Fluorel ®	HK
Chloroprene	Chloroprene rubber	CR	CR	Neoprene	BC, BE
Silicone	Silicone rubber	VMQ	VMQ	PVMQ	FC, FE, GE
Fluorosilicone	Fluorosilicone rubber	FVMQ	FVMQ	FVMQ	FK
Polyacrylate	Polyacrylate rubber	ACM	ACM	ACM	EH
Ethylene Acrylic	Ethylene Acrylic rubber	AEM	AEM	Vamac ®	EE, EF, EG, EA
Styrene-butadiene	Styrene-butadiene rubber	SBR	SBR	SBR	AA, BA
Polyurethane	Polyester urethane / Polyether urethane	AU / EU	AU / EU	AU / EU	BG
Natural rubber	Natural rubber	NR	NR	NR	AA

Vamac ® and Viton ® are registered trademarks of E. I. du Pont de Nemours and Company or affiliates.
Fluorel ® is a registered trademark of Dyneon LLC

General Properties of Elastomer Classes & Rubber Compounds:

Very Good	Good	Average	Poor	Not Recommended
(green)	(blue)	(yellow)	(red)	(grey)

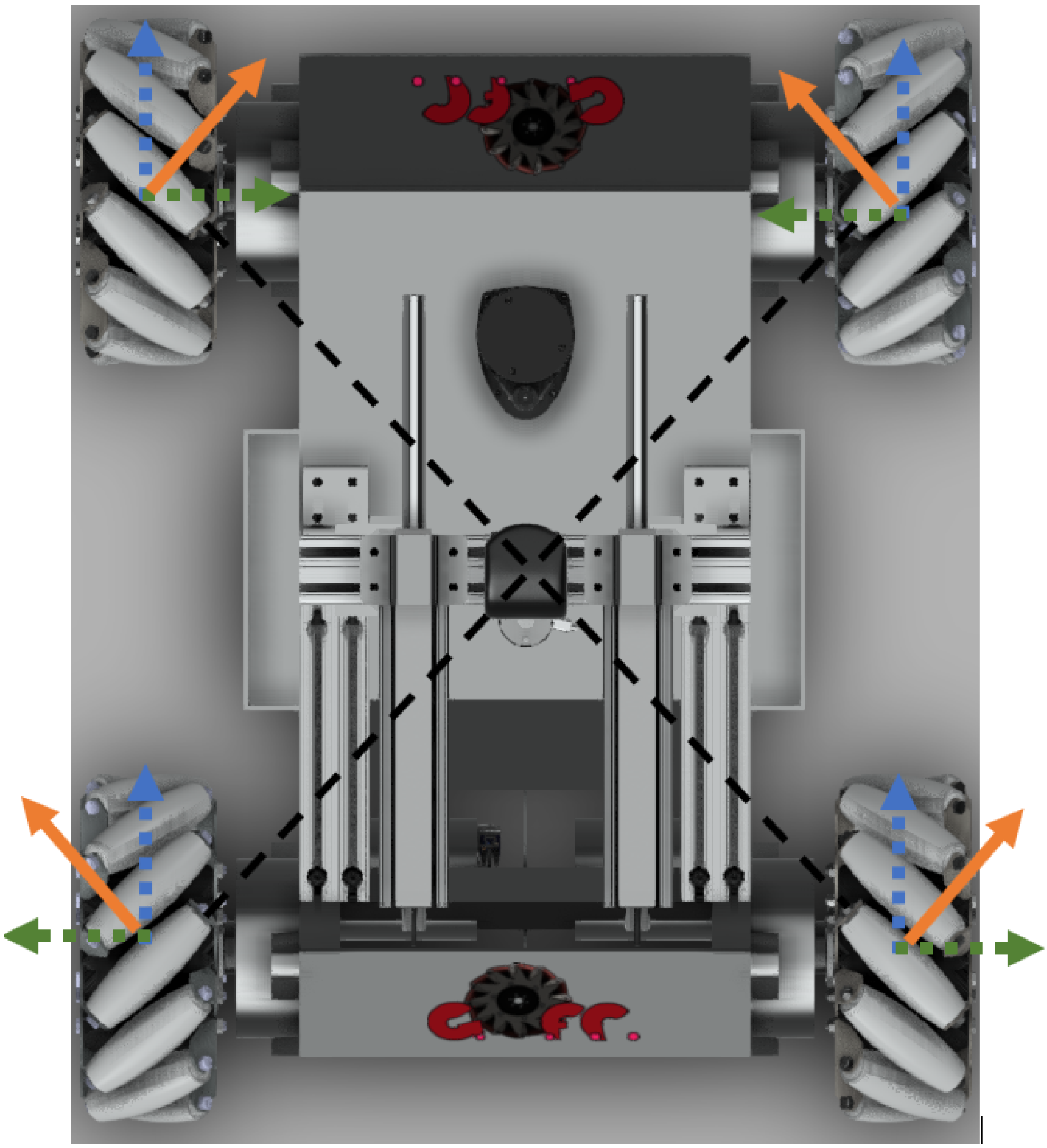
Basic Property	NBR	HNBR	EPDM	FKM	CR	ACM	AEM	SBR	AU/EU	VMQ	FVMQ	NR
Economy of Material	Very Good	Poor	Good	Average	Good	Average	Poor	Very Good	Average	Average	Poor	Very Good
Compression Set Resistance	Very Good	Very Good	Very Good	Very Good	Good	Poor	Good	Good	Average	Good	Good	Very Good
Resilience (Rebound)	Good	Good	Good	Good	Good	Average	Good	Good	Good	Good	Good	Very Good
Tear Strength	Good	Very Good	Good	Good	Good	Average	Good	Average	Good	Poor	Average	Very Good
Heat Aging Resistance	Average	Good	Good	Very Good	Average	Very Good	Very Good	Average	Very Good	Very Good	Very Good	Average
Ozone Resistance	Poor	Good	Good	Very Good	Good	Good	Very Good	Poor	Very Good	Very Good	Very Good	Poor
Resistance to Oil & Grease	Good	Good	Poor	Very Good	Good	Very Good	Average	Poor	Good	Average	Very Good	Poor
Fuel Resistance	Poor	Average	Poor	Good	Poor	Very Good	Poor	Poor	Average	Poor	Good	Poor
Water Swell Resistance	Good	Good	Very Good	Good	Average	Poor	Good	Very Good	Poor	Very Good	Very Good	Very Good
Gas Impermeability	Good	Good	Average	Good	Good	Average	Good	Average	Good	Poor	Poor	Average
Dynamic Service / Abrasion Res.	Good	Good	Good	Average	Good	Good	Good	Very Good	Very Good	Poor	Poor	Very Good
High Temperature - Standard	212 °F	300 °F	300 °F	390 °F	250 °F	300 °F	300 °F	212 °F	175 °F	450 °F	400 °F	220 °F
High Temperature - Special	250 °F	-	-	-	-	-	-	-	-	480 °F	-	-
Low Temperature - Standard	-22 °F	- 22 °F	-60 °F	5 °F	-40 °F	-60 °F	-40 °F	-50 °F	-60 °F	-75 °F	-75 °F	-60 °F
Low Temperature - Special	-60 °F	-40 °F	-	-30 °F	-	-	-	-	-	-	-	-

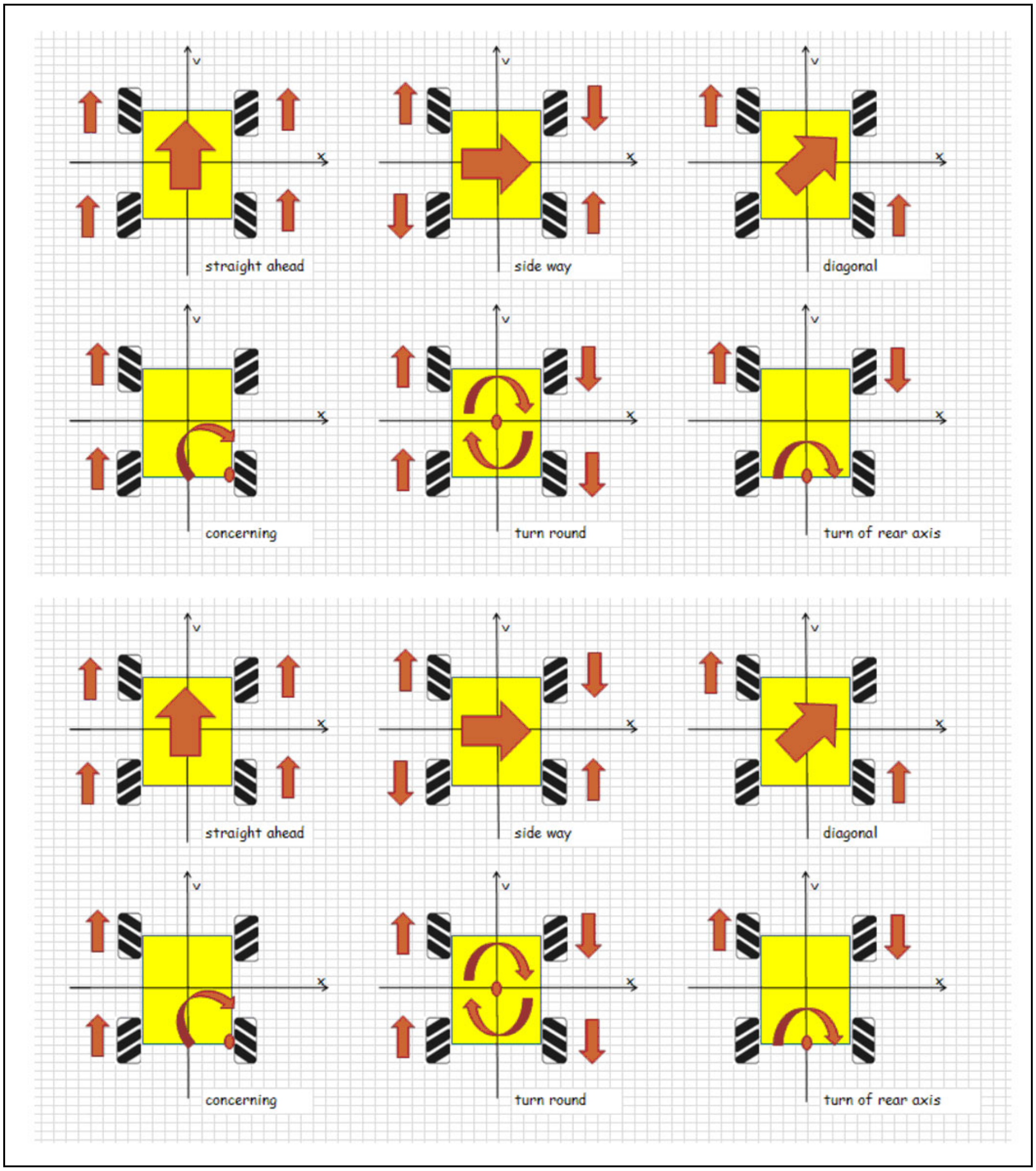
^ T

Appendix A9

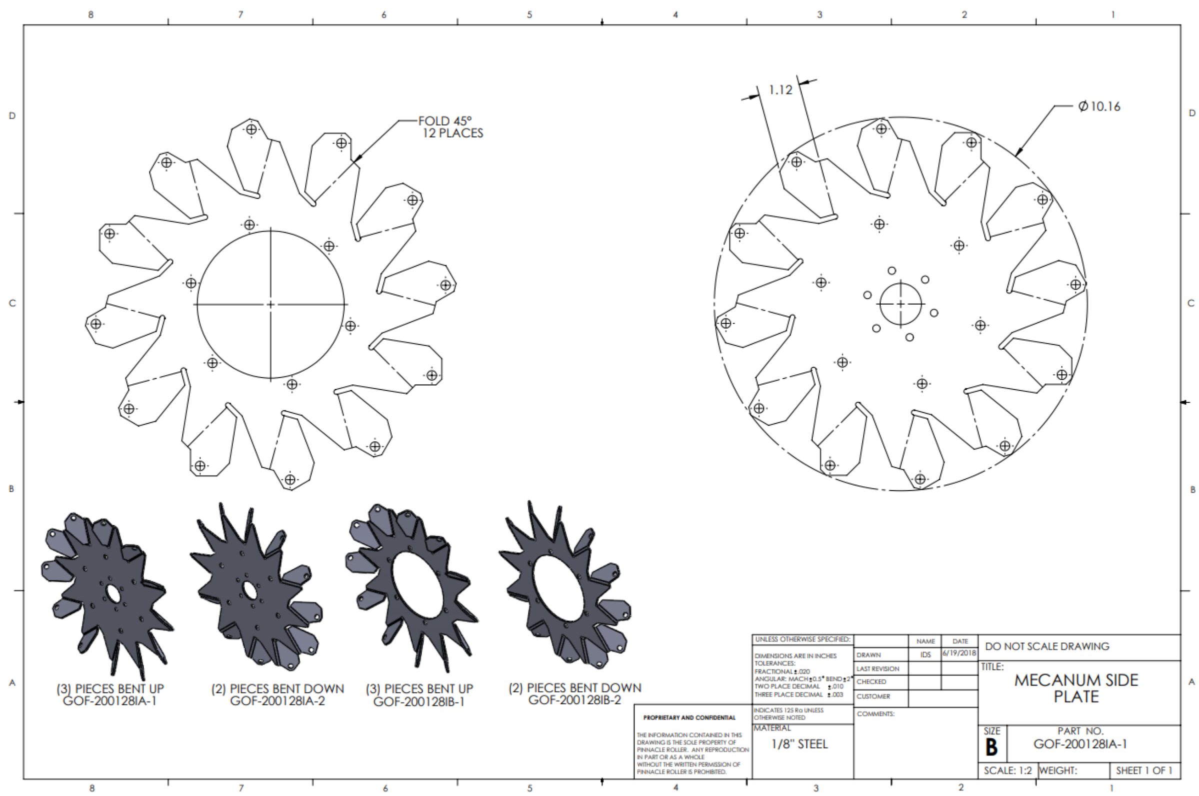
Appendix A10

In order for the mecanum wheel design to work properly the wheels must be arranged as below, with an “X” being formed by the orientation of the wheels. It is evident from below that mecanum wheels work on the principle of canceling forces. For example when traveling forward all wheels must rotate in the same direction, when this happens all of the green component forces are opposed by another green force therefore canceling out and traveling in the direction of the blue force.





Appendix A10



Appendix A11