Trans-Gravitational Robot (TGR) with Linear and Angular Momentum Control

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Masters of Science

in

Robotics Engineering

by

Felix Sanchez

August 2020

APPROVED:

Professor Marko Popovic, Advisor

Professor Winola Lenore Rasmussen

Professor Micheal Gennert

Abstract

Due to the danger posed by the harsh environment of space, robots have been used since the beginning of space exploration as precursors to humans. As humans increase their presence in space, the risks associated with operating in such a dangerous environment increase as well. Every space and planetary mission requiring task-oriented locomotion and dexterous manipulation could benefit from having a robot capable of either assisting or replacing human presence in these unforgiving environments. Exploration, maintenance, and experimentation are a few examples of tasks that can be automated or simplified by robots with humans in-the-loop for higher level decision. This work presents the simulated implementation of the Trans-Gravitational Robot (TGR) which is a quadrupedal, lightweight, mobile and dexterous robot envisioned to operate in various gravitational environments including orbital, Lunar, and Martian.

Using an ISS interior provided by NASA in Gazebo as a simulation environment, calculating the momentum generated by segmented masses in each limb and adding each limb's momentum have allowed for precise control over the full body's linear and angular momentum. After pushing off and entering a free floating phase, 3D sensing via LiDAR in simulation paired with a full body pose controller allow for the TGR's limbs to be positioned for catching itself on a target handrail. When paired with the obstacle avoidance algorithm, the TGR is capable of avoid objects within its flight trajectory or preparing for impact. This work focuses on higher level controllers focusing on momentum instead of joint angles and expanding the capabilities of quadrupedal robots in situations where momentum impacts the robot greatly such as space environments or aerial situations under gravity.

Acknowledgements

I would like to acknowledge my advisor Prof. Marko Popovic of Worcester Polytechnic Institute(WPI) who made my thesis into reality with his teachings and advise. He provided crucial feedback and guided me towards my goal of space robotics. Through his teachings, I have improved as a researcher and as an engineer. Thank you to Prof. Winola Lenore Rasmussen and Prof. Micheal Gennert for taking their time to review my work and for being members of my review committee.

I would also like to thank my friends who inspired me to push forward and offered new perspectives on the obstacles I've faced. I'd like to give a special thanks to Juan Diego Florez for his irreplaceable assistance with this thesis and for Albert Enyedy for his help with writing advice and thesis process guidance.

Lastly, I'd like to acknowledge my parents and brother, as well as my partner Emma, for their unwavering support and constant enthusiasm for what I do. Being surrounded by these great people has fueled my drive towards my ambitions, without them none of this would be possible.

Thank you,

Felix Sanchez

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Space Robotics

As humanity continues to expand our grasp into outer space, robotics continues to offer solutions to obstacles faced by human body. The dangerous environment posed by operating in space is one that is difficult to mitigate when working with humans. Temperature, pressure, and radiation are just a few factors to consider when sending a human off-Earth. However, a robot is innately more robust than the human body. Heating and cooling can be done to crucial components, pressure tends to not pose much of an obstacle to metal limbs, and radiation can be shielded against. At the time of this writing, less than 1000 individuals have been in this environment, and for good reason [1]. Of this small pool of brave people, no more than 20 have perished in transport to/from or within space [2]. The need for safer and more reliable means to explore this new environment for future explorers is being pioneered by robots.

Robots sent to this environment are highly specialized to operate within a set environment. A good example would be the Mars Curiosity Rover operated by The National Aeronautics and Space Administration (NASA). This rover is about the size of a small SUV, equipped with various scientific sub-assemblies, and rocker-boogie suspension system for traversing the martian terrain [3]. Using the 6 wheels on the rover, it is able to crawl along the martian surface avoiding steep inclines and large rocks. The wheels on the Mars Curiosity Rover are special in that they are made to endure the corrosion caused by the martian soil [4]. These design choices stem from decades of previous work on other rovers such as the Pathfinder. However with more operations occurring in orbit or in transport to other locations, the need for free-floating robots is growing.

One such robot is the SPHERES (Synchronized Position Hold Engage and Reorient Experimental Satellite) robots which are volleyball sized robots designed to operate within the interior of the International Space Station (ISS) [5]. The SPHERES platform is a spherical robot, as the name suggests, that has over a decade of experience and was used as a test bed for sensors and hardware for operating within a free-floating environment. Its main form of propulsion is liquid carbon dioxide that is expelled from thruster solenoids [6]. The SPHERES platform was designed with adding attachments for future use such as RINGS (Resonant Inductive Near-field Generation System) and the Slosh Experiments [7] [8]. RINGS was used to test wireless power transmission between the satellite robots while the Slosh experiments were to test liquid dynamics in unconventional gravity situations.

This thesis aims to combine the concepts of various space robots in order to design a platform capable of operating in a variety of gravity environments and utilize different locomotion strategies. More specifically, this thesis will dive into the specifics of operating a multi-limbed robot in a free-floating environment such

as the ISS. The ultimate goal of this robot is to have high utility in more than one stage of a mission, being able to function in transit to a destination and on foreign terrestrial surfaces. Drawing inspiration from how humans move in space, the TGR will be able to grab the same handrails astronauts use to move and push off in a similar way [9]. It will be able to adjust its pose as it traverses to its target and prepare for catching the handrails but invoking moments about its center of mass utilizing calculated movements of its limbs. The main focus will be had on the push off stage as this is where the linear and angular momentum of the robot can be changed such that it aids in following a desired trajectory.

1.2 Trans-Gravitational Robot (TGR)

The TGR is a quadruped designed for operating both in microgravity and terrestrial environments. It utilizes four limbs, each with three degrees of freedom, for locomotion. It is being fully simulated in Gazebo with custom linear and angular momentum controllers. The TGR is currently equipped with a 3D LiDAR and six degree of freedom accelerometer for controller inputs. Real mass distributions and actuator parameters are programmed into the TGR model used for simulations to offer a more accurate simulation. The TGR is notable for its total mass of less than five kilograms and a max length less than one meter when fully extended.

1.2.1 Motivation

The main motivation behind the fundamental design of the TGR is to have a partner robot for astronauts to use in more than one situation. Similar to a multi-tool, we hope for the TGR to be seen as a highly functional machine that can perform useful action irrespective of the location it finds itself in. Whether that be floating in the ISS as a highly dexterous camera and sensor test bed or as a companion for astronauts as they traverse the martian surface.

We expect the TGR to be fully capable of teleportation and autonomous behaviors in order to minimize the amount of human life at risk. This can be in the form of going outside the ISS to assess damages to the outside of the hull or in setting up research equipment on the lunar surface. Missions that can be automated should be in order to increase the efficiency of the astronauts. The TGR would serve as an invaluable tool for humanity as it begins to find itself in new environments.

1.2.2 Problem Statement

We seek to develop in simulation a new platform potentially capable of surpassing the functionality seen by previous space robots and expand upon its utility by adding more limbs and focusing on using the environment to our advantage. The ultimate goal of the TGR is to function both on the ISS and with plans to make it capable of traversing lunar and martian surfaces. The usage of consumable propellant and high energy draining actuators limits the longevity of other robots however the TGR aims to mitigate this issue by using minimal energy to push off and move around the ISS similar to how human astronauts push off and grab hold of hand rails and reposition to push off again. There exists a trade off between energy and time required to complete a task where more energy is required for faster completion time.

For this thesis, a focus will be had on controlling global variables such as linear and angular momentum as opposed to individual joint positions and velocities. This greatly simplifies the mathematical complexity and allows for a generalization for future robots to implement. The TGR will be able to control linear momentum with respect to a locally static coordinate frame by controlling the force vector from pushing off. It should also be able to freely alter and control its angular momentum in the push off stage with the ability to control its pose as it floats to its destination. With these two controllers established, the TGR will be able to be expanded upon and add different end effectors and sensors for different mission objectives.

1.3 Outline

In this thesis, we will review related works and their impact on the TGR in Chapter 2. In Chapter 3, we will discuss the design philosophy behind the TGR and the experiments will be outlined and specifics about the testing environment will be explained. Lastly, Chapter 4 will offer the results and Chapter5 discussions on the results followed by conclusions in Chapter 6.

Chapter 2

Related Works

Space robots come in many shapes and forms however each is a highly advanced machine capable of completing more than one mission. This is a mandatory characteristic for these robots as it is a sizable investment for governments and companies to send these devices into space. When the Space Shuttle was in operation, it cost approximately $50,000 [10] kilogram to move matter into Low Earth Orbit (LEO). With advances in space travel, companies such as SpaceX are helping to reduce the cost per kilogram. At the time of this writing, SpaceX's Falcon 9 offers approximately $3,000 per kilogram in LEO [11]. Keeping costs and mission objectives in mind, the TGR draws from the benefits of other space robots. This chapter will present some of the related works that have impacted the current design and control of the TGR.

As robotics advances, limbed robots are increasing their ability to manipulate the world around them. A reasonable example is that found in rovers implemented by NASA. A quick look back at history one can see three notable generations of martian rovers. The oldest rover, NASA's Sojourner was a relatively small rover with the ability to roam around the martian surface capturing images and surface

data [12]. It was a huge achievement for humanity as this was the first wheeled rover to rove on another planet. The Sojourner paved the way for future rovers with more sensing and data collection such as the Spirit and Opportunity rovers.

The twin rovers, the Spirit and Opportunity, implemented rock abrasion tools and cameras on a single limb capable of limited movement in order to collect samples of the martian surface as well as high resolution panoramic cameras for better imaging [13]. These two rovers were larger and heavier than their predecessor and covered more ground due to advances in power management such as better batteries and solar panels. The lessons learned from the twin rovers then lead to the Curiosity rover. It is similar to the previous generation of rovers hosting advanced scientific equipment and cameras for high resolution panoramic photos, a comparison can be seen in Fig. 2.1. The Curiosity rover has a five degree of freedom robotic arm with 5 of 12 instruments on the end effector. With it's increased size, more scientific equipment can be held in the main body of the rover. The addition of this robotic limb allows for "robot selfies" to check the physical state of the rover over time.

Figure 2.1: A comparison of the different martian rovers deployed by NASA

These robots all share in common the rocker boogie suspension system that is now symbolic with robotic rovers. This gives the robots great maneuverability and stability while also having redundancy in case of wheel errors like what Spirit experienced [14]. Keeping in mind the limitations of wheels, there are several approaches being looked into for locomotion on these new surfaces.

As time passes and a new era space exploration arrives, new designs for terrestrial rovers appear. The Asagumo rover is a quadruped robot meant for exploring the lunar surface, marking a stark a difference in locomotion when compared to other robots [15]. NASA's VIPER (Volatiles Investigating Polar Exploration Rover) will prospect the lunar surface for frozen water and other minerals [16]. Launched July 2020, NASA's Perseverance is the next step in martian rovers. Alongside the Perseverance, a new martian helicopter robot named Ingenuity will be the first powered flight on any planet beyond Earth [17] [18].

2.1 Momentum Based Control

The idea of computing the momentum generated by a robot's limbs is not a new concept however its utilization on the center of mass is a relatively new field. Utilizing a form of segmentation, several degrees of freedom can be blocked to be controlled by other systems better fit for the task. The focus on a hierarchy of tasks allows for compliant torque control while attempting to complete several objectives at once. This idea of blocking degrees of freedom and multi-task control inspired the duo momentum controllers and limited manoeuvrability [19].

Based on work done on an actual humanoid robot, HRP-2, in which the linear and angular momentum of the full body was controlled [20], we are able to determine that similar principles can apply when used on a non-static robot. However, due to

the simplification of our system there is much less computational load in order for this controller to run. This will serve as a benefit when real-time control is concerned for gravity based quadruped. These quadrupeds will need to jump under gravity and must do so in a way to not flip uncontrolled as well as land anticipating the angular momentum it generated from jumping. For this thesis, due to the nature of being free floating, instead of assigning specific angular momentum values, we choose to maintain it at zero in order to prevent the TGR from spinning out of control.

Previous projects have been able to develop a free-floating system generalized Jacobian matrix (GJM) which served as inspiration for the work we did on this thesis [21]. The situation in which we experiment with the TGR allows us to simplify the problem and reducing our calculations into a 2D solution. Due to the yaw and Y-axis being ignored, we are able to operate in this 2D environment. This meant we could circumvent using more complicated mathematical models to accomplish desirable results.

This is further expanded upon to the center of mass in a later project where the concepts of hierarchical control is applied to more complex scenarios and robots [22]. Focusing on reactant forces and pressure points, forces can be calculated about a robot's center of mass in order to developed a closed loop controller for its momentum. Keeping in mind constraints and desired postures, this improved controller setup allows for more controllability and expands the possibilities of how a robot can grasp its environment using multiple points of contact.

In terms of momentum in walking robots, previous experiments demonstrate that while angular and linear momentum are not conserved when experiencing ground reaction forces, the body does make an effort to control all six degrees of freedom in the form of its momentum. The angular momentum is a near zero value as a human body moves in order to maintain its balance while the desired direction of travel has a momentum point in that direction. [23]. A large focus on how the right side of the body cancels the angular momentum of the left side of the body throughout the gait offers insight on how to minimize the TGR's angular momentum in undesired axis as it moves its limbs.

2.2 Microgravity Robots

As briefly mentioned, one of the most impactful and easily recognized platforms for microgravity is SPHERES. The MIT (Massachusetts Institute of Technology) based project is a ball like robot with several attachment ports [5]. Having been used for over a decade, it has explored the dynamics of microgravity based robots and cohabitation on the ISS with astronauts.

The newer Astrobee is meant as a successor to the SPHERES platform, sporting expansion ports like SPHERES and being a free-floating robot as can be seen in Fig. 2.2. A key difference from the SPHERES project is that Astrobee comes with a perching claw allowing it to grasp onto the ISS's internal hand rails which can be found through the station's surfaces. Another notable difference is that Astrobee is slightly bigger and cubic in shape as well as using electric impellers for 6 degree of freedom propulsion around its environment [24]. Similar to its predecessor, the Astrobee platform will be for automating tests and recording crew member activities, freeing up a member that no longer needs to hold a camera.

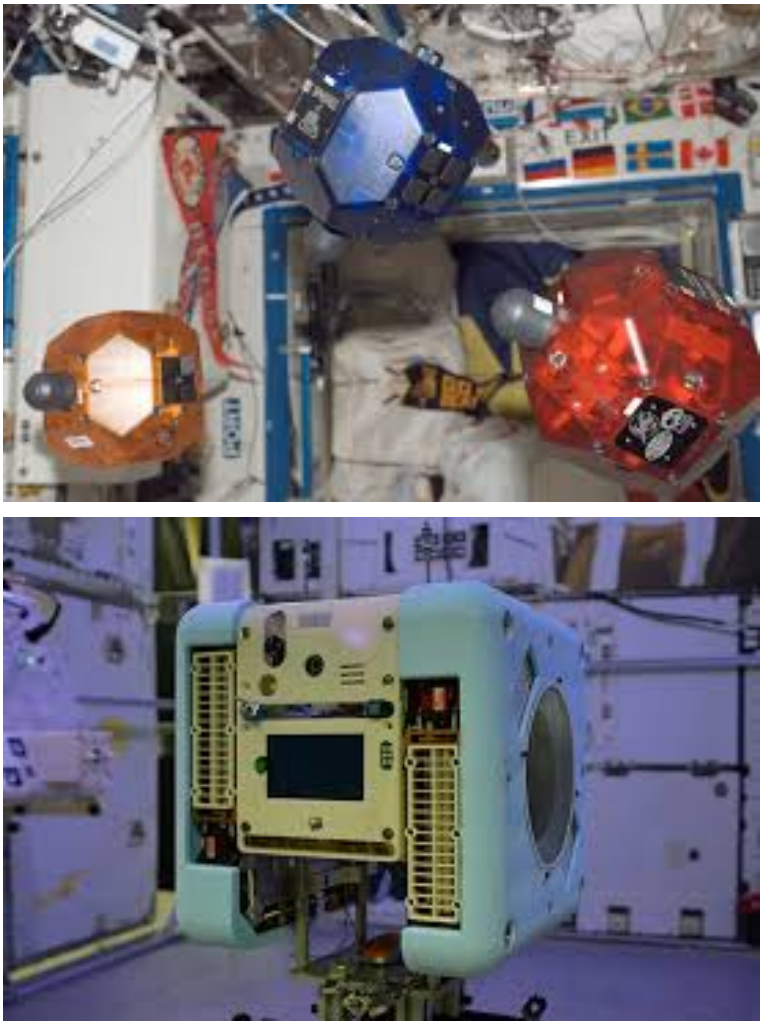

Figure 2.2: SPHERES (Top) and Astrobee (Bottom) physcial characteristics

Another robot designed for off-Earth applications is Robotnaut 2 as can be seen in Fig. 2.3. This robot is a humanoid robot with 2 arms and 2 prehensile limbs and the ability to attach its upper body onto other platforms such as a rover base [25]. This robotic platform served as a source of inspiration for this thesis as it explores the idea of increasing the technological readiness level of the robots that operate in space. The technology readiness level is a system used by NASA to measure the maturity of a particular technology ranging from level 1 to level 9. [26] The more these highly advanced tools are capable of doing, the more available time astronauts have to do more complex/sensitive tasks.

Figure 2.3: Robonaut 2 working within the ISS

Chapter 3

Methodology

3.1 Design

The TGR is designed with multiple mission environments in mind. Firstly, we want the TGR to be able to operate within the ISS and as such this reduces the maximum size it could be. It must be able to work around astronauts as well as be capable of cooperation with them. This would require some form of compliance and high level vision in order to reach this point. However, for this thesis we are focusing on higher level design concepts such as mass distribution and degrees of freedom. With further development, more specifics about the physical characteristics of the TGR will be established such as exact sensors and actuators. The simplified model used in this thesis is seen in Fig. 3.1.

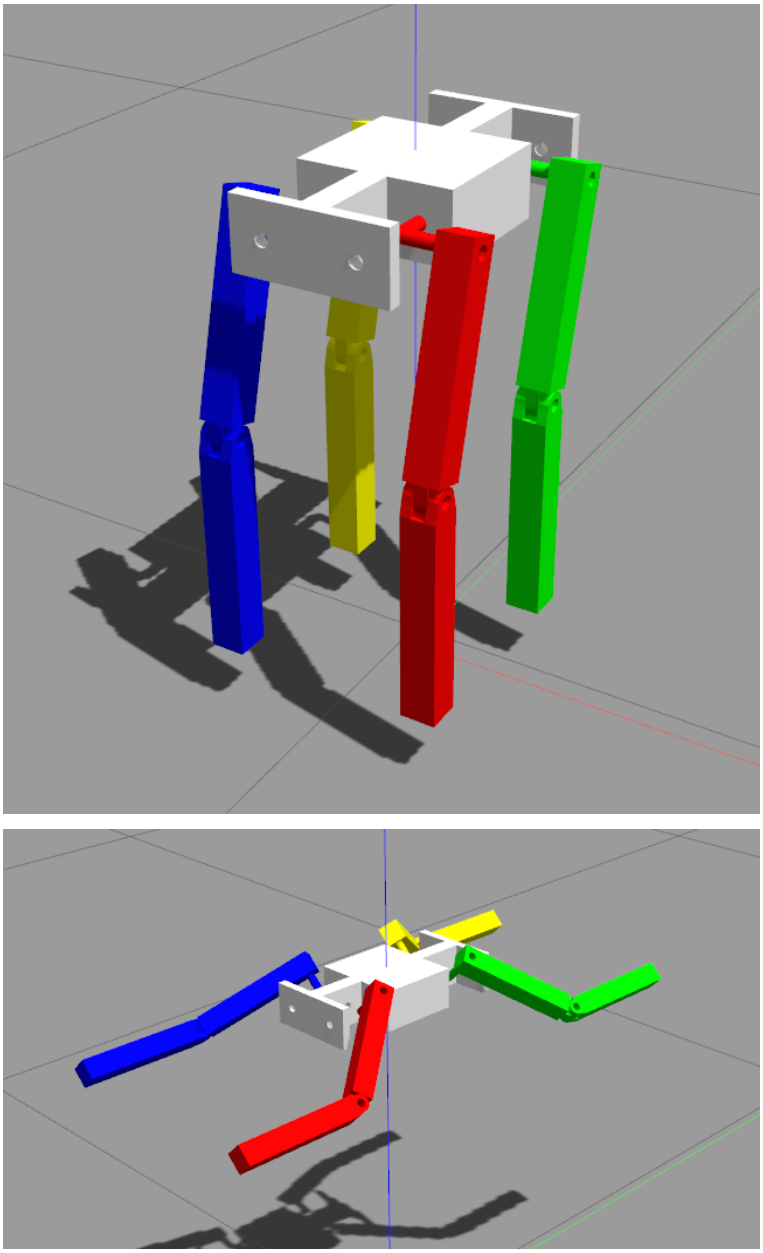

Figure 3.1: The TGR in a surface configuration and microgravity configuration

Keeping in mind the types of missions operated by other space robots. The TGR aims to do more with less. A focus on its prehensile limbs and compliance will be had as development on the TGR continues in the near future. The addition of additional forms of pose control may also be considered such as flywheels to better control the yaw of the TGR as it travels in microgravity or to prevent it from falling in case of slipping under gravity.

3.1.1 Limbs

Being a quadruped, the TGR will be capable of using different gaits in terrestrial environments such as the lunar or martian surface. Each limb is envisioned as both arms and legs depending on the scenario. This is similar to many primates that have the ability grasp with their feet. By increasing the number of grasping end effectors, the TGR is capable of anchoring itself and being able to manipulate objects at the same time. These three degree of freedom limbs are physically capable of reproducing motion similar to other quadrupeds such as the MIT mini cheetah [27] and the Boston Dynamics Spot [28]. The ends of each limb would feature an expansion port to swap between different end effectors as well as potentially using a two-in-one gripper capable of being walked on. The limbs are color coded in order to be able to determine the orientation of the TGR as it moves. Table 3.1 describes which color corresponds to each limb.

Color Code of Simplified TGR	
Red	Front Left
Blue	Front Right
Green	Back Left
Yellow	Back Right

Table 3.1: TGR Limb Color Coordination.

3.1.2 Main Body

The main body of the TGR, often referred to as the trunk, is where most of the mass of the robot is located. For the time being, it is seen as a uniformly distributed mass. As the project advances through different phases, more detail will be added such as the location of the batteries, CPUs, sensor arrays, and other electronics. The shape of the current trunk is rectangular in nature and serves as the reference point for the center of mass of the TGR as well as establishing the root transform frame when doing kinematics and center of mass calculations.

The goal of the TGR is to be under five kilograms and have its longest dimension be no more than one meter long when full extended in order to comfortably fit within the ISS's interior. This minimizes the cost of running the TGR and allows for the potential of having multiple TGRs working within a shared space. All robot physical parameters are elaborated in Table. 3.2 and can be seen in Fig. 3.2.

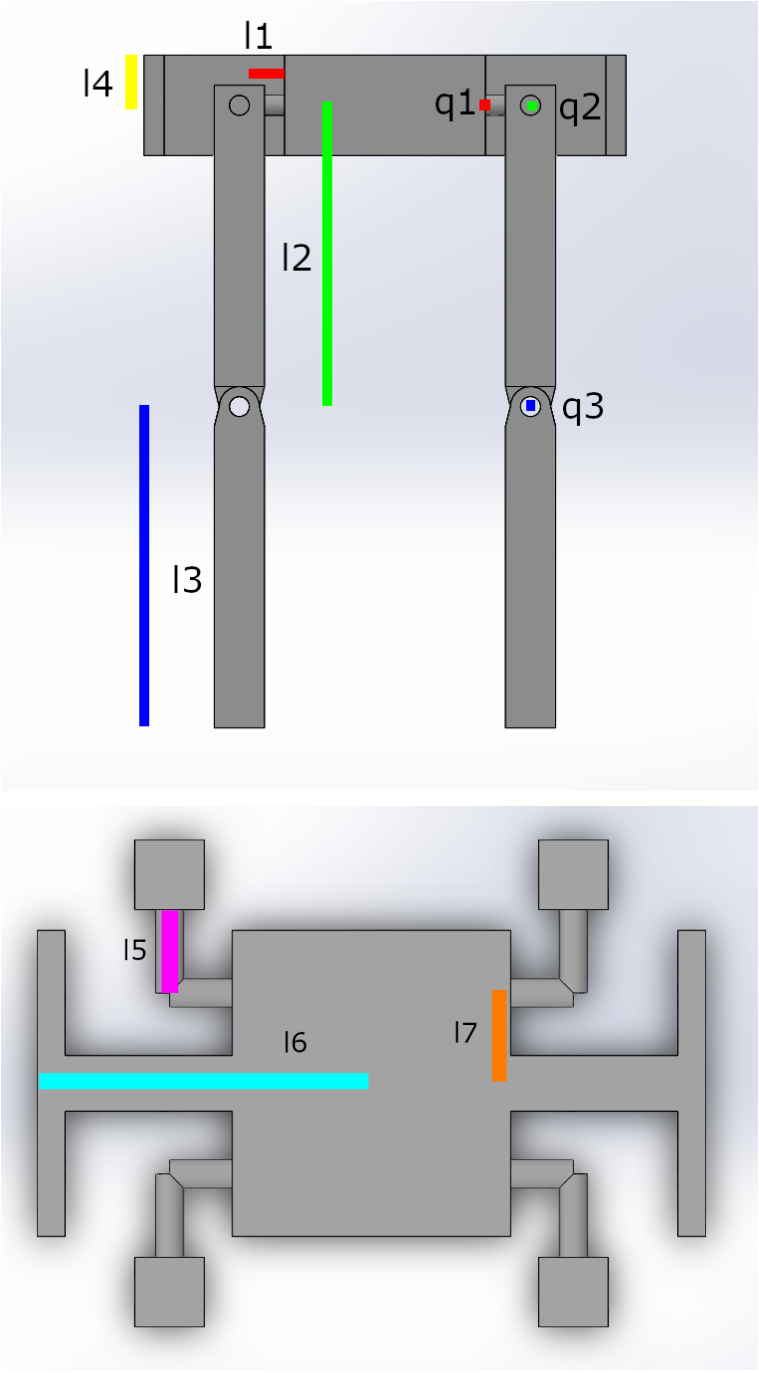

Figure 3.2: The TGR's Physcial Characteristics

TGR Physical Parameters	
m_1	Mass of Link 1, 0.06 kilogram
m_2	Mass of Link 2, 0.435 kilogram
m_3	Mass of Link 3, 0.450 kilogram
m_t	Mass of Trunk, 1 kilogram
q_1	Joint 1 in rads
q_2	Joint 2 in rads
q_3	Joint 3 in rads
l_1	Limb Length Between Joint 1 and 2, 0.0725 meters
l_2	Limb Length Between Joint 2 and 3, 0.15 meters
l_3	Limb Length Between Joint 3 and the end effector, 0.16 meters
l_4	Half Thickness of TGR, 0.025 meters
l_5	Limb Length Between Joint 1 and 2, 0.0725 meters
l_6	Half Length of TGR, 0.12 meters
l_7	Length from Center of TGR to Joint 1, 0.04 meters

Table 3.2: TGR Physical Symbols Matched with their values.

3.2 Control and Simulation

The control section of this thesis will cover the two momentum controllers that work alongside each other as well as the reasoning behind their setup. We initially determined that it would be optimal to limit the degrees of freedom that we wish to control of the TGR. Because we are utilizing a simplified model of the TGR, much of the complexities of real world applications are overlooked such as friction within the joints and power consumption. Furthermore, we use simulated sensors on the TGR model with Gausian noise introduced in order to better mimic real sensors.

The simulation software used is Gazebo, which is a free robot simulation software installed on Ubuntu 18.04 [29]. ROS (Robot Operating System) is installed as well running the Melodic version. As for the robot model used, it is a custom made URDF (Unified Robotic Description Format) file that describes the TGR simplified model which has realistic joint parameters based on potential actuator we can find on online marketplaces.

3.2.1 Angular Momentum Controller

The control over a global variable such as angular momentum requires that the center of mass of the TGR and the ZMP (Zero Moment Point) be known. However, recall that we decided to reduce the degrees of freedom of the system and as such we have reduced the dimensions in which we operate in. As such, we moved from a six degree of freedom system to a four degree of freedom system. We are eliminating the yaw (z-axis) rotation of the TGR and the Y-axis linear movement. This greatly simplifies the mathematical modeling and control scheme required as well as allows for better actuation methods for controlling specifically the yaw.

For pitch control, movements are mirrored about the sagittal plane of the TGR, or the plane when viewing the side view of the TGR. This ensures that the yaw and roll are not theoretically affected by movements of the limbs. After these changes, the robot can be simplified further into a 2D plane on the projected on the X-Z plane. In order to more accurately model the masses of the TGR, each link is split into 10 heterogeneous masses. This emulates a more realistic distribution of mass along the limbs due to actuators and sensors as well as proposed thickness of different parts of each limb. The jacobian matrix, J, for describing individual limb motion is Eqn. (3.1). Recall that all robot physical parameters are elaborated in Table. 3.2.

$$\mathbf{J} = \begin{bmatrix} l_2\cos(q_2) + l_3\cos(q_2+q_3) & l_3\cos(q_2+q_3) \\ \\ l_2\sin(q_2) + l_3\sin(q_2+q_3) & l_3\sin(q_2+q_3) \end{bmatrix} \tag{3.1}$$

With this in mind, we calculate the center of mass using the geometric center of the trunk as a reference point following Eqn. (3.2). We do this in every iteration of our controller in order to calculate how the center of mass is moving as well as what force is required to move it in a desired angular velocity. utilizing Eqn. (3.2), we are able to find the x, y, and z components of the center of mass, X_{CoM}, in order to have a proper understanding of the system dynamics for future control of other degrees of freedom. The mass of each segmented mass, m_i, and the distance to the aforementioned reference point, x_i, are also found via a kinematic analysis.

$$X_{CoM} = \frac{\sum_i m_i x_i}{\sum_i m_i} \tag{3.2}$$

As can be seen in Eqn. (3.3), the angular momentum is calculated based on each individual mass's moment of inertia, I, with their corresponding angular velocity, ω. The moment of inertia for each mass is modeled as a rod about an end following Eqn. (3.4) where the mass is M and the distance to the point that it rotates about is L.

$$L = I\omega \tag{3.3}$$

$$I = \frac{1}{3}ML^2 \tag{3.4}$$

Once the angular momentum of each individual mass is found and added together, we can calculate the control angular momentum about the center of mass of the TGR. Please note that this includes all four limbs in order to properly model the system even if we are restricting the degrees of freedom. Using this, we are able to describe the angular momentum at any given point as seen in Eqn. (3.5). The total angular momentum is L_{total}. The initial angular momentum present when already starting the controller is $L_{initial}$. Any external momentum from outside of the angular momentum controller such as that generated by the linear momentum controller is represented by $L_{external}$. The angular momentum being generated by the angular momentum controller is $L_{control}$. By tying a PD control loop to total angular momentum, we can use the control angular momentum as an output while the input is the desired angular momentum.

$$L_{total} = L_{initial} + L_{external} + L_{control} \tag{3.5}$$

The initial momentum is determined by the starting parameters of the simulation and is always set to zero. However, by including this within our controller we allow for the potential to expand the situations in which the controller can operate in. The external angular momentum can stem from actions that create angular momentum, more specifically momentum generated by the linear momentum controller or due to collisions in the environment. Lastly, the control angular momentum is that which is generated in an active effort to minimize the angular momentum in the system. Due to the law of conservation of momentum, the angular momentum does not change in the system as it is free floating and not touching anything. However Eqn. (3.5) applies while the TGR is in contact with a handrail and is pushing off. This is when the controller is actively minimizing the angular momentum of the whole system then they enter a dormant state and a pose controller is activated.

3.2.2 Linear Momentum Controller

Similar to the angular momentum controller, the linear momentum controller is focused on the TGR's center of mass. Taking advantage of the center of mass found with Eqn. (3.2), we find the linear velocity, v, of each mass,m, utilizing the jacobian in Eqn. (3.1) and altering the link lengths to match the position of each mass. Using Eqn. (3.6) we are able to determine the linear momentum, P, of each mass which we sum to find the control linear momentum of the system. This controller affects the second and third joint angles of each limb it controls. By tying a PD control loop to total linear momentum, we can use the control linear momentum as an output while the input is the desired linear momentum following the desired trajectory.

$$P = mv \tag{3.6}$$

In order to determine what action to take in order to achieve the desired linear momentum, we must know our desired trajectory. This trajectory is assumed to be a straight line with a slow in the X-Z global plane. By taking the desired trajectory as a unit vector we are able to generate a straight line equation for the trajectory. Once the equation of this line intersects the wall plane, we can calculate what the force vector would need to be for the end effector's normal force to travel directly though the center of mass in an effort to minimize angular momentum while still achieving desirable linear momentum. The magnitude of this force vector is a required input and is multiplied with this force vector in order to combine both force and direction. Utilizing inverse kinematics as seen in Eqns. (3.7) we are able to move the tip of the limb such that force generated is what we calculated it should be. We choose to use an "elbow down" configuration however this can be changed by editing the signs of Eqns (3.7). In addition, the TGR's physical parameters are presented in Table 3.2.

Recall that all robot physical parameters are elaborated in Table. 3.2.

$$q_3 = \arccos(\frac{x^2 + z^2 - l_2^2 - l_3^2}{2l_2l_3})$$

$$q_2 = \arctan(\frac{x}{z}) - \arctan(\frac{l_3 \sin(q_3)}{l_2 + l_3 \cos(q_3)}) \tag{3.7}$$

Following suit with the angular momentum controller, the total linear momentum controller is determined by the summation of the initial, external, and control linear momentum. This relation can be seen in Eqn. (3.8) which follows the same logic as its angular momentum twin. The total linear momentum is P_{total}. The initial linear momentum present when already starting the controller is $P_{initial}$. Any external momentum from outside of the linear momentum controller such as that generated by the angular momentum controller is represented by $P_{external}$. The linear momentum being generated by the linear momentum controller is $P_{control}$. Since momentum is conserved, we can only effect the linear momentum of the system during the push off phase, after which the linear momentum controller is dormant.

$$P_{total} = P_{initial} + P_{external} + P_{control} \tag{3.8}$$

In order to utilize both the linear and angular controller, a segmentation of responsibility was enacted to avoid overlap in the controllers. The linear momentum controller had control over the back two limbs used to push off while the angular momentum controller had control over the front two limbs in order to counter act the generated angular momentum of the back limbs motion. As more degrees of freedom are liberated and allowed to be controlled, this segmentation must be revisited and a hierarchy must be enacted such that multiple controllers may have an input on each limb in order to work together towards their respective goals without completely

nullifying the efforts of each other. This can come in a form of weighted inputs from controllers based on a higher level state controller that prioritizes different kinematic objectives.

3.2.3 Obstacle Avoidance and Pose Controller

A simple obstacle avoidance algorithm was made such that using 3D point cloud data from a front facing time-of-flight sensor is able to give enough information to the TGR about what approaches it in flight. In an effort to minimize complexity, the TGR was placed in a simulated ISS environment supplied innately by the Gazebo simulation software and an example LiDAR sensor was chosen from an online market place to simulate. The obstacle avoidance algorithm determines whether an object getting closer to the TGR is considered a threat to its trajectory. Objects within 1 meter are considered to be threatening and if they get within 0.5 meters the TGR will attempt to move its pose in such a way to avoid it by rolling $\pm 90^{\circ}$.

The pose controller is made up of two simple PD controllers, a roll and a pitch controller. The roll controller will change the first joint of each limb in an effort to affect the roll angle of the TGR with respect to its starting orientation. The pitch controller will change the second joint of each limb in an effort to drive the pitch angle of the TGR with respect to its starting orientation. Both of these controllers rely on the IMU in the trunk of the TGR and do not take into consideration any external forces let alone momentum generated by their motions. This is utilized during times when the total momentum can not be altered such as free floating to the next destination. Constraints can be made such that the front limbs are in a catching state while the back limbs are the ones adjusting the roll and pitch for grasping the incoming handrail however this require more advanced vision which is beyond the current scope of this thesis.

3.2.4 Experiment Layout

In order to test the functionality of the controllers, we designed a few experiments to test the range that the controllers can operate in. We ran two sets of experiments, one where only the linear momentum controller had control over the TGR and another set of experiments where both the linear and angular momentum controllers are active. Under the aforementioned constraints and simplifications, we were able to simulate a range up to $\pm 45^{\circ}$ with respect to the perpendicular direction of the push off surface. In Chapter 4 we review and discuss the results about the controllers as well as any aspects noted through experimentation and observation.

The testing for the obstacle avoidance is conducted in the simulated ISS environment. A cardboard box model provided by Gazebo is utilized as a sample obstacle. It is accelerated to a slow speed of 0.25 $\frac{m}{s}$ and pointed towards the TGR within the ISS. This is done as to simulate the TGR free floating through a corridor on the ISS with an obstacle in the way. Utilizing the pose controller, the limbs are moved in order to control the roll and pitch of the TGR in an effort to avoid the object. RVIS is a special robot data visualizer that integrates nicely with Gazebo and relies on the ROS communication network to show incoming data from the TGR. It is used to project the 3D point cloud from the LiDAR as well as the intensity of the points that represent "threat" level.

Chapter 4

Results

4.1 Push Off Phase

The controllers both performed in a satisfactory manner. To begin with, the linear momentum controller can be seen at three key angles: -12, 0, and +12 degrees with respect to the direction perpendicular to the handrail. The linear momentum controller is able to converge both the angular and linear momentum to desired points in less than a second during the push off phase as can be seen in Fig. 4.1. The desired angular momentum for all controller experiments is set to zero in an effort to minimize rotating out of control during the free floating phase. The desired linear momentum is different for each experiment differs only in its trajectory unit vector. Originally, we tested in 5° intervals up to 45° however the controller became more unstable as we diverged from 0°. It was found that when operating by itself, the linear momentum controller could reach to about 24° before instability rendered the controller ineffective however when paired with the angular momentum controller, the range was halved down to about 12°. As the angle increases, so too does the angular momentum generated by the linear momentum controller. Because of this

effect, the angular momentum controller begins further away from its desired position eventually being unable to reach it within the time frame of the push off action.

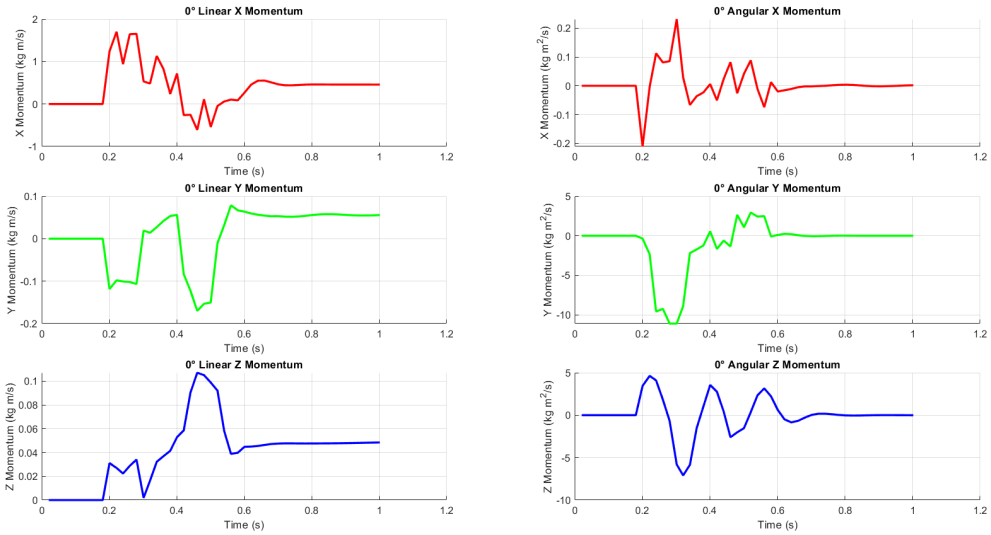

Figure 4.1: Performance of the Linear Momentum Controller at 0°

The following experiments changed the angle at which the desired unit vector trajectory is based on. Figs. 4.2 and 4.3 cover the linear momentum controller at $-12°$ and $+12°$ respectively. It can be seen that without an angular momentum controller, the angular momentum of the TGR has some error that results in the TGR rotating in the free floating phase, complicating any operations starting in the free floating phase such as obstacle avoidance and catching handrails. Avoiding complexities is a priority especially for a 12 degree of freedom system such as the TGR. As the operations of the TGR expand, the control becomes increasingly complex as well.

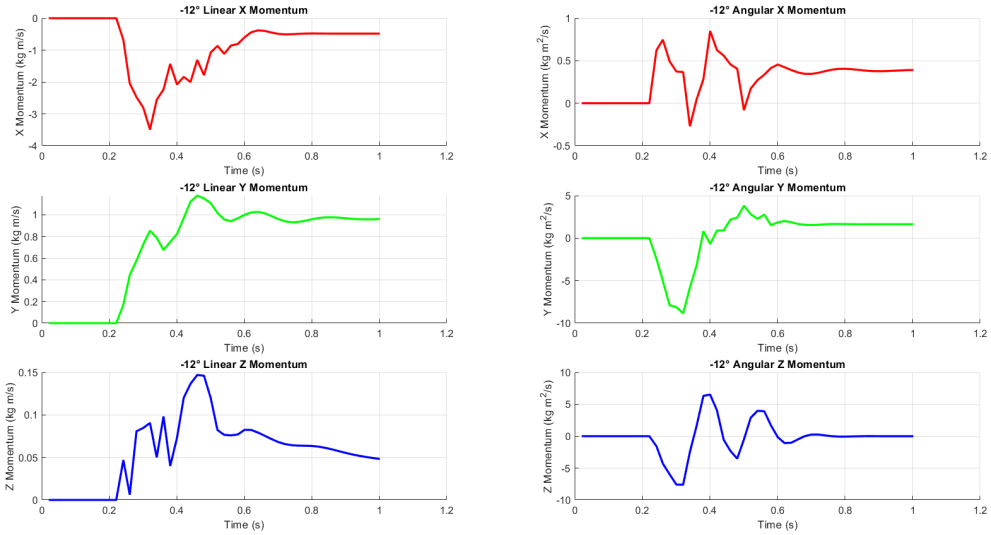

Figure 4.2: Performance of the Linear Momentum Controller at $-12°$

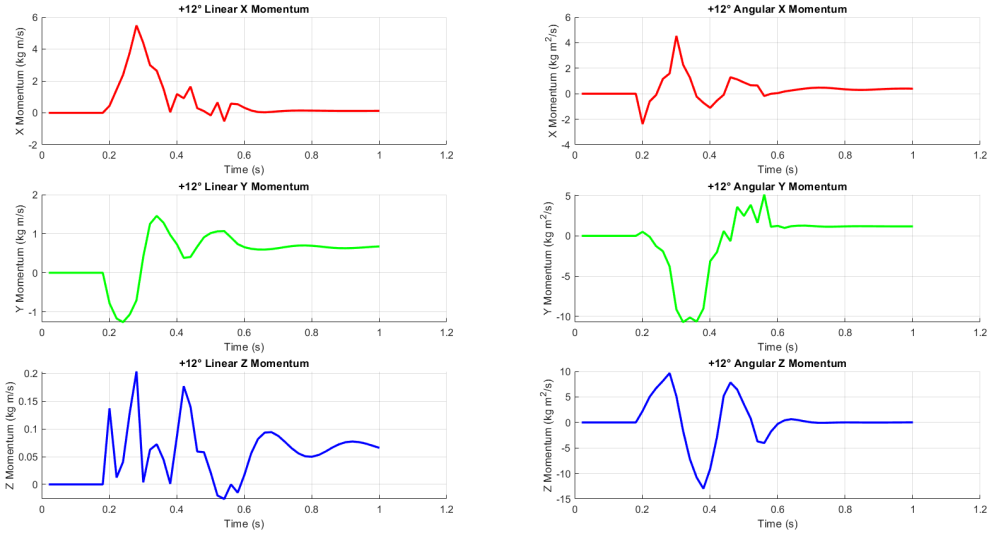

Figure 4.3: Performance of the Linear Momentum Controller at $12°$

In order to counter act the accumulation of angular momentum, the angular momentum controller must be added to the simulation. As such the results of repeating the experiments done on the linear momentum controller can be seen in Figs. 4.4, 4.5, and 4.6 with 0°, −12°, and +12° unit vector trajectory angles respectively. It can be clearly seen that as the linear momentum is relatively the same between the experiments with just the linear momentum controller and both momentum controllers. However, the angular momentum generated in the first half of the experiments using only the linear controller is converging to zero. This proves the functionality of using the angular momentum controller to calculate the angular momentum being generated throughout the entire TGR.

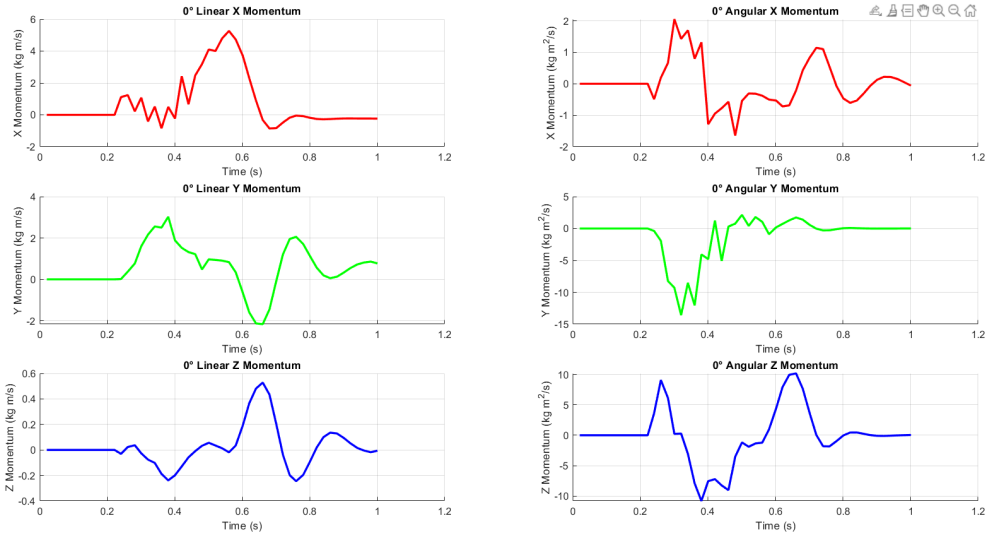

Figure 4.4: Performance of the both Linear and Angular Momentum Controllers at 0°

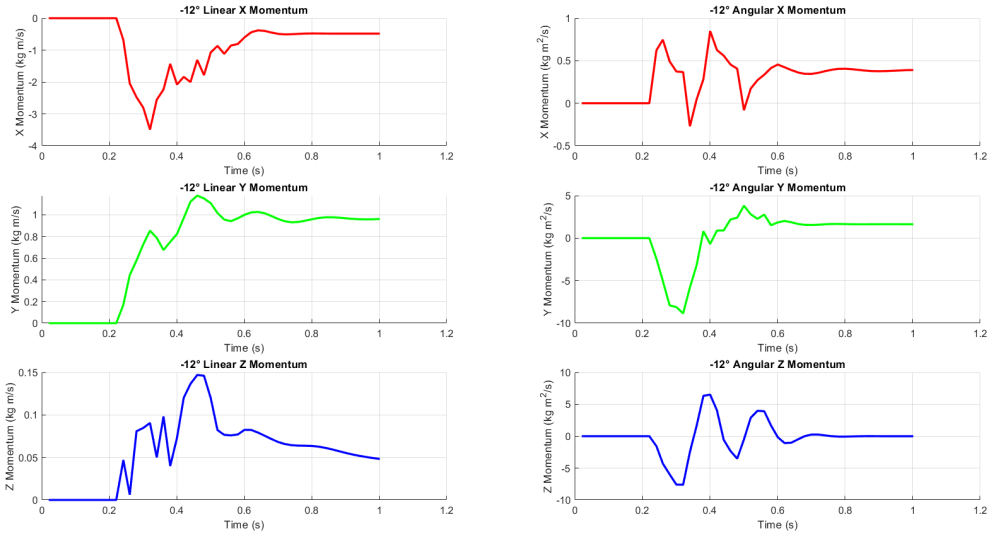

Figure 4.5: Performance of the both Linear and Angular Momentum Controllers at $-12°$

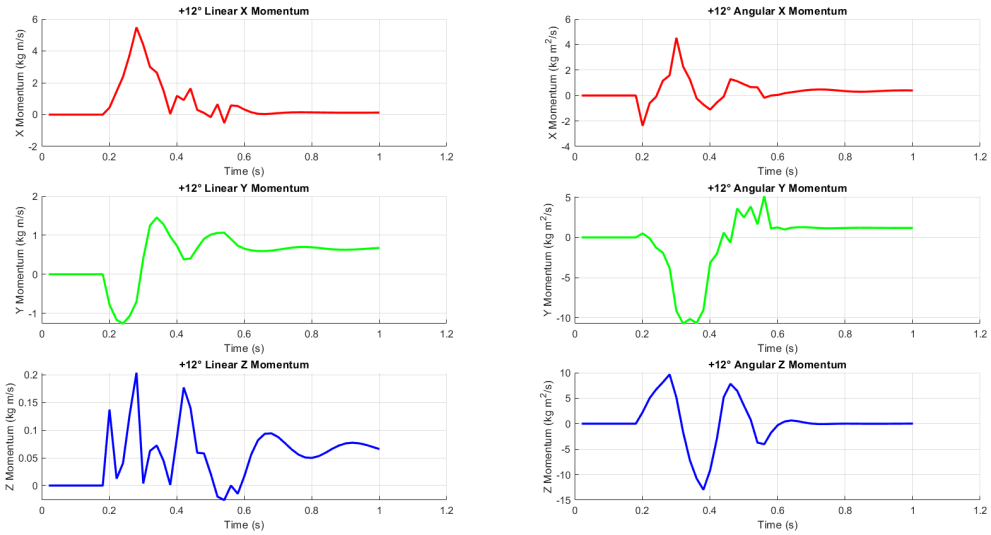

Figure 4.6: Performance of the both Linear and Angular Momentum Controllers at $12°$

4.2 Free Floating Phase

During the free floating phase, the momentum of the center of mass can no longer be manipulated due to the conservation of momentum as we are no longer in contact with anything and are drifting towards our goal. As a proof of concept, the pose controller is capable of rotating the roll of the TGR approximately $\pm 90^{\circ}$, which was achieved experimentally. Due to the realistic joint constraints, the limit is mostly in the form of joint velocities and limb mass. By altering these values, we would be able to achieve more or less range of rotation. It has been found that the obstacle is detected at 1.5 meters away and is able to detect the velocity of the obstacle.

The algorithm attempts to find the centroid of the object and uses that as the reference point to find the velocity. By comparing the distance of this centroid over iterations of data collection, a rough estimate can be had of the speed of the object. The distance and velocity of the obstacle is passed to the pose controller which chooses to dodge by rolling left, rolling right, pitching up, pitching down, or a combination of a roll and a pitch maneuver. It was found that if an object is within 5 centimeters of the left or right side of the TGR, it can roll to dodge it. However this algorithm does not perform well when trying to dodge objects above or below the TGR as rolling or pitching will make the situation potentially worse. Figs. 4.7 and 4.8 show the cardboard box obstacle approaching the TGR. Equally, the 3D point cloud data can be visualized from RVIZ in both situations in Figs. 4.9 and 4.10.

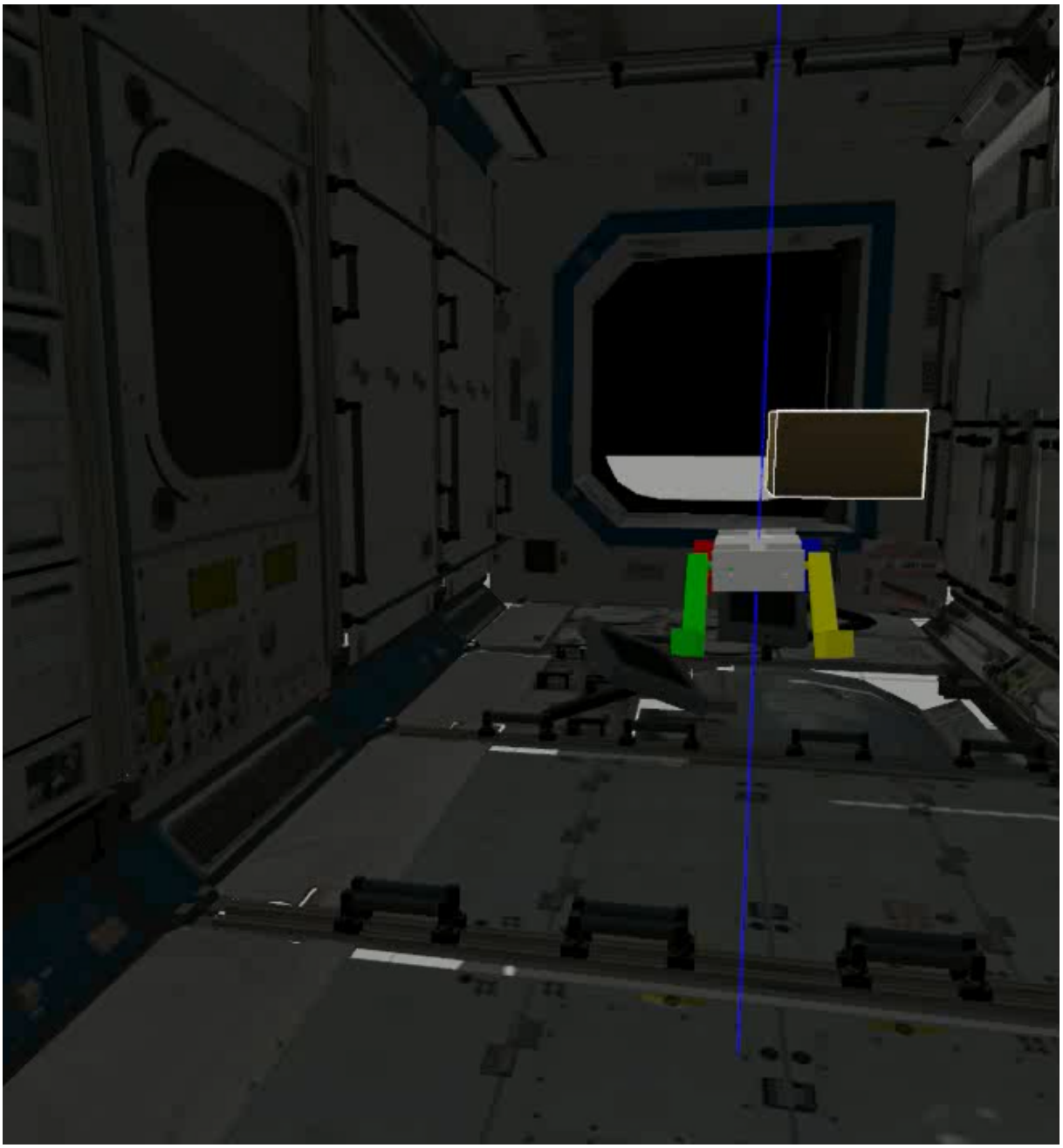

Figure 4.7: Obstacle approaching in Gazebo Simulation

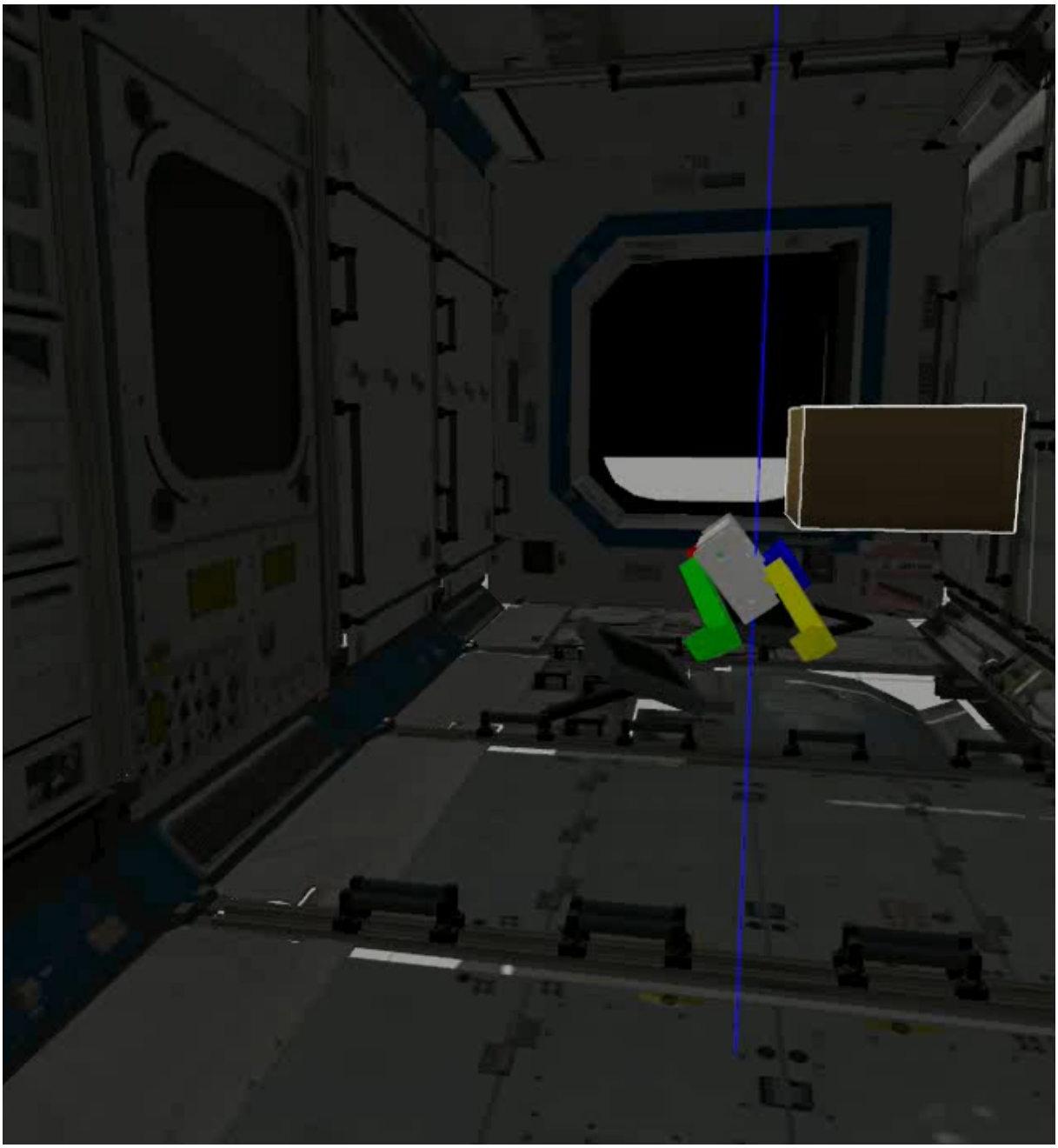

Figure 4.8: Obstacle much closer in Gazebo Simulation

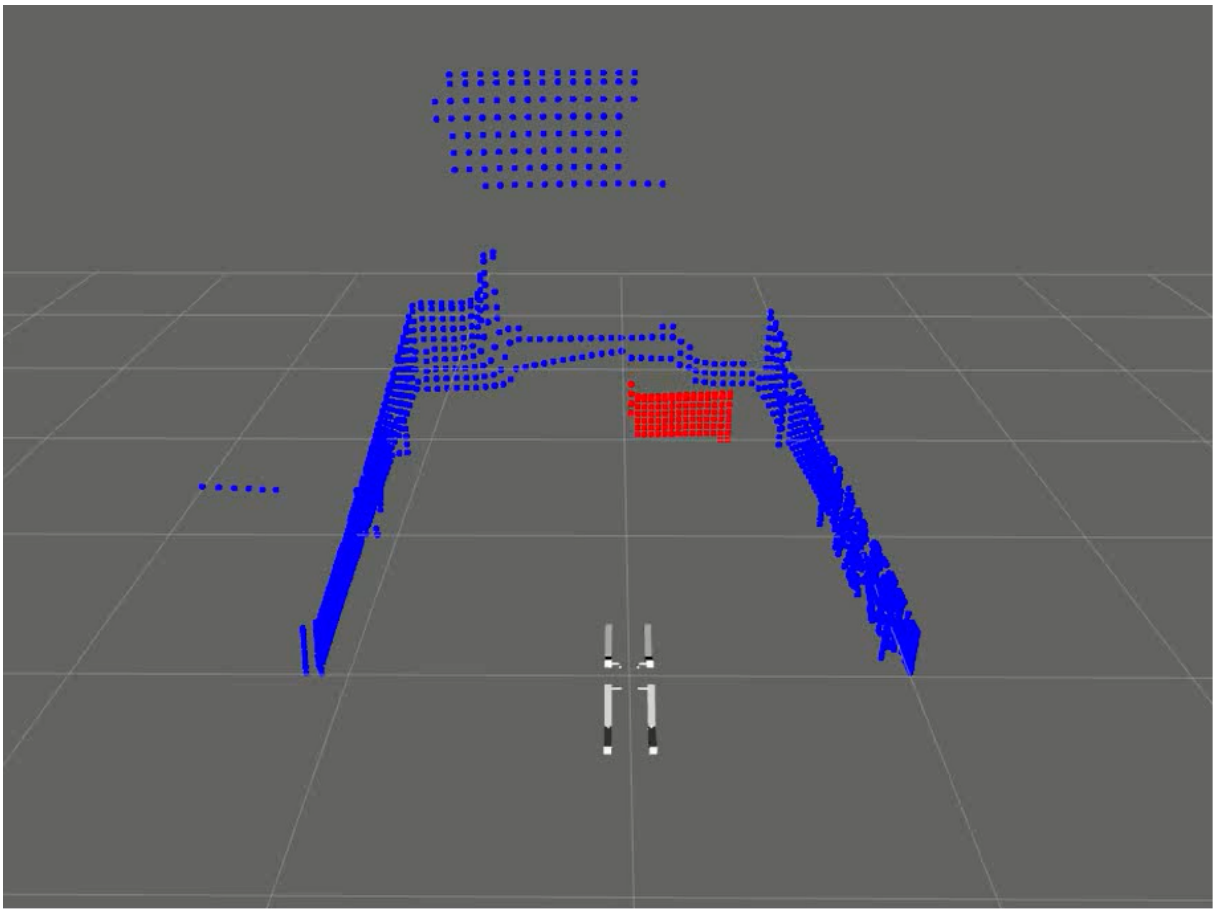

Figure 4.9: Obstacle approaching as seen in RVIZ

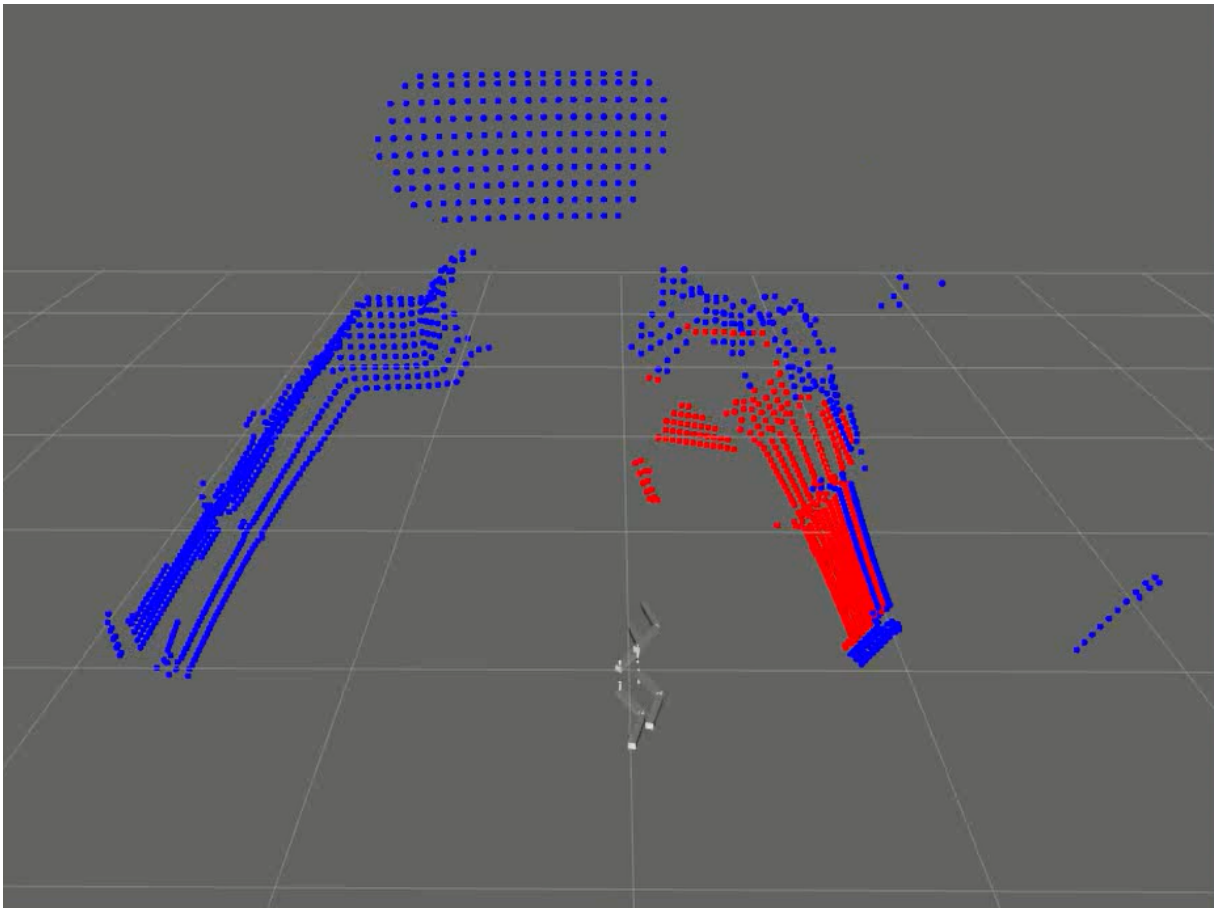

Figure 4.10: Obstacle much closer as seen in RVIZ

Chapter 5

Discussion

It was found that the momentum controllers operated at an excellent level within a range of $\pm 24^{\circ}$ when operating one controller at a time. If both controllers are run at the same time, the range is reduced approximately by half to a range of $\pm 12^{\circ}$ however considering that this is without a proper implementation of grippers to anchor in place, the potential for these controllers to work better in the future is very high. Even though the yaw and y-axis degrees of freedom are limited in this implementation of the simulated TGR, the other four degrees of freedom offer excellent controllability as we were able to get the momentum to converge to desired levels within the small time frame of pushing off from a handrail. Fig. 5.1 captures an instance of the TGR pushing off.

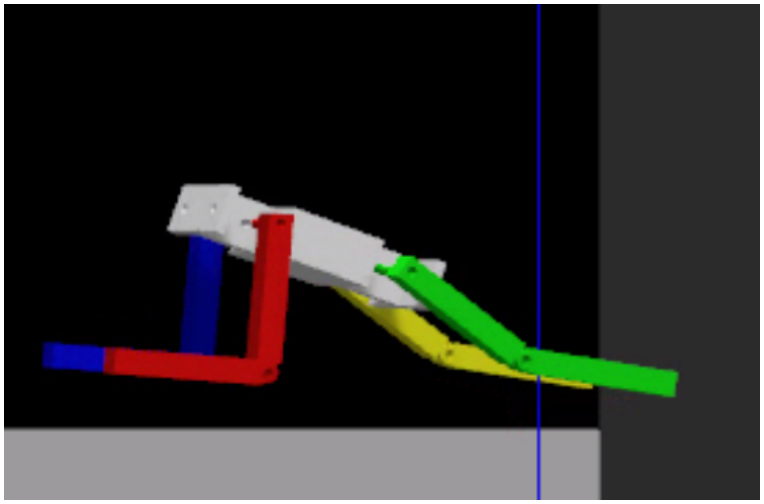

Figure 5.1: TGR pushing off at +24° with respect to the direction perpendicular to the handrail.

It must be noted that there is was an unexpected issue with not having control over the yaw of the TGR. If during the push off phase the legs are not synced up as they are pushing or have uneven force, there is an uncontrollable yaw rotation that diverges the controllers. However with the addition of a control moment gyroscope for the yaw, this issue will be hopefully completely overcome. Also, segmenting the masses of each limb as opposed to having just the center of mass for each limb improved the accuracy of the momentum controllers and is highly recommended for any other momentum based controllers. The more subdivisions one can reasonably fit into the link, the more accurate the center of mass the momentum calculations are.

The obstacle avoidance worked in utilizing the 3D point cloud data to identify potential threats and classify them as dodge-able or non-dodge-able. For the time being, the functionality is limited for this thesis however as the TGR continues to be developed, this algorithm will be the basis for more complicated maneuvers such as bracing for impact and potentially recalculating the momentum it experiences as it impacts the obstacle. The current state of the pose controller is also satisfactory as

it is capable of rotating the trunk to most desired orientations in the roll and pitch dimensions. Due to the four limbs, the front two can be positioned for catching a target handrail while the back limbs are moved about to pose the robot for grasping.

It must be noted that the impact on rotating LiDAR sensors are also not modeled in this simulation and could present an issue unless a passive form of point cloud generation is used in a microgravity environment. The inability to dodge obstacles about it poses an issue for future implementations of the TGR that must be addressed. The TGR currently does not include any form of compliance however this can be easily added, especially if the TGR is going to be working around sensitive equipment and people alike.

Chapter 6

Conclusion

To summarize this thesis, the linear and angular momentum controllers functioned as expected and were able to control the momentum of the TGR as it was in the push off phase and transitioning to the free floating phase. Once in the free floating phase, the TGR is able to avoid some obstacles by using 3d point cloud data to observe its environment. The TGR's current state is that of the initial conception and proving concepts central to its operation and serves as a foundation for future simulation work with the TGR and eventually moving into a physical implementation of the TGR.

The contribution of this thesis is mostly in the linear and angular momentum controllers which can be used in more than just microgravity situations. When a quadruped is jumping and making initial ground contact, the momentum controllers can be used to mitigate any unwanted momentum. The TGR is on its way of fulfilling its higher level goals of being a foundation for trans-gravitational robots capable of operating in multiple gravity environments. As such, using global variables such as momentum to control the TGR is crucial and, as proven with this thesis, highly feasible.

6.1 Future Work

Adding a form of controlling the yaw of the TGR is possibly the most important and obvious next step. This paired with improving the hierarchy of controllers would drive the control scheme of the TGR closer to completion. Along with using more detailed models of the TGR and improved actuators, beginning work on making a physical model of the TGR would allow for real world testing to validate the simulation based testing. However this will require the infrastructure to mimic microgravity, ideally in at least 2 degrees of freedom. A reasonable alternative could be using a reduced gravity aircraft aptly dubbed Vomit Comet [30]. Expanding the degrees of freedom for each limb will also assist with when the TGR is perched on a handrail. Once the TGR is advanced in its design, revisiting the idea of a modular end effector may lead to improvements in the capabilities of the TGR. Ideally the end effector would be capable of being walked on as well as used as a gripper for objects and handrails.

Bibliography

[1] NASA. Total duration of man-flights for different spacecrafts. 2007.

[2] W. Harwood. Astronaut fatalities. 2005.

[3] NASA. Mars curiosity rover. 2018.

[4] A. Rankin A. Steffy G. Meirion-Griffith D. Levine M. Schadegg M. Maimone O. Toupet, J. Biesiadecki. Terrain-adaptive wheel speed control on the curiosity mars rover: Algorithm and flight results. *California Institute of Technology*, 2019.

[5] B. McCarthy D. Miller A. Saenz-Otero C. Jewison, D. Sternberg. Definition and testing of an architectural tradespace for on-orbit assemblers and servicers. 2014.

[6] D.W. Miller J.G. Katz, A. Saenz-Otero. Development and demonstration of an autonomous collision avoidance algorithm aboard the iss. 2011.

[7] NASA. Department of defense synchronized position, hold, engage, reorient, experimental satellites-rings. 2016.

[8] NASA. Thether slosh. 2018.

[9] W. Bluethmann M. Goza R.O. Ambrose-K. Alder F. Rehnmark, I. Spain. An experimental investigation of robotic spacewalking. pages 366 – 384 Vol. 1, 12 2004.

[10] D.R. Jenkins. Space shuttle: Developing an icon 1972 2013. *Specialty Press*, 2016.

[11] SpaceX. Capabilities services. 2014.

[12] JPL/NASA. Mars pathfinder / sojourner rover. *California Institute of Technology*, 2018.

[13] NASA. Spirit and opportunity. 2018.

[14] NASA Patent Abstracts Bibliography Section 1. Abstracts. Articulated suspension system. page 19, 1990.

[15] Y. Yi J. Yu J. Haruyama I. Hong, E. Cho. 3d printed structure of lacus mortis pit crater with assumption of a cave underneath. *2nd International Planetary Caves Conference*, 2015.

[16] S. Loff. New viper lunar rover to map water ice on the moon. *NASA*, 2019.

[17] S. Potter. Nasa, ula launch mars 2020 perseverance rover mission to red planet. 2020.

[18] G. Hautaluoma A. Johnson, D.C. Agle. How nasa's mars helicopter will reach the red planet's surface. *NASA*, 2020.

[19] O. Khatib L. Sentis. Control of free-floating humanoid robots through task prioritization. *IEEE*, pages 1718–1723, 2005.

[20] K. Kaneko K. Fujiwara K. Harada-Kensuke K. Yokoi H. Hirukawa S. Kajita, F. Kanehiro. Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2:1644 – 1650 vol.2, 11 2003.

[21] Y. Umetani, K. Yoshida, et al. Resolved motion rate control of space manipulators with generalized jacobian matrix. *IEEE Transactions on robotics and automation*, 5(3):303–314, 1989.

[22] O. Khatib L. Sentis. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, pages 505–518, 2005.

[23] H. Herr M. Popovic. Angular momentum in human walking. *The Journal of Experimental Biology*, pages 467–481, 2008.

[24] E. Ackerman. How nasa's astrobee robot is bringing useful autonomy to the iss. *IEEE Spectrum*, 2017.

[25] NASA. Robonaut 2. 2011.

[26] NASA. Technology readiness level. 2012.

[27] B. Katz G. Bledt S. Kim D. Kim, J.D. Carlo. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. 2019.

[28] Boston Dynamics. Spot. 2019.

[29] Gazebo. Robot simulation made easy. 2019.

[30] NASA. Nasa reduced gravity research program. 2009.

Appendix

Please email the author at (fasanchez AT wpi DOT edu) for GitHub access to the project repository.

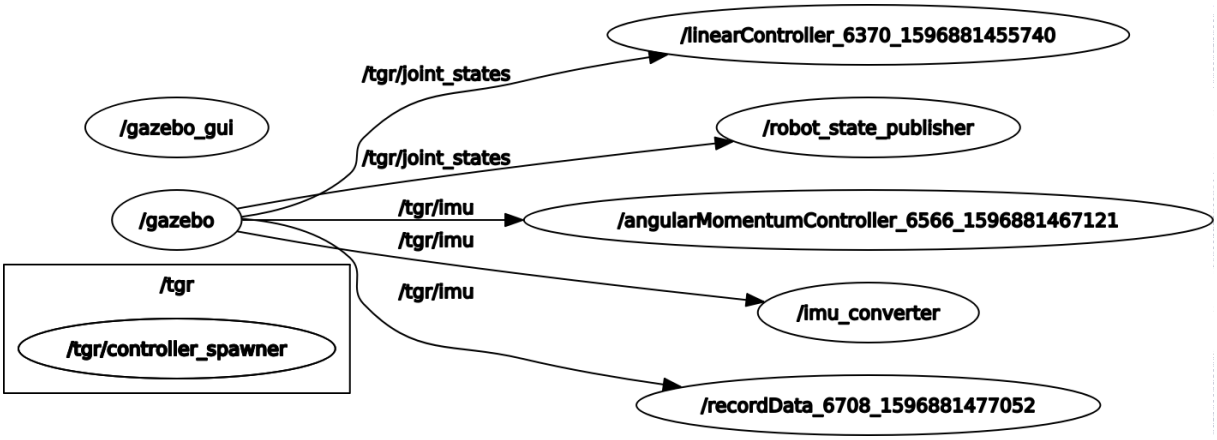

Figure 6.1: ROS Node Graph of Push Off Phase

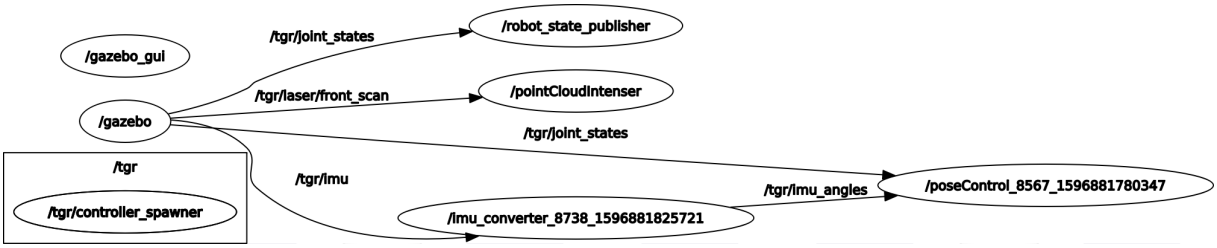

Figure 6.2: ROS Node Graph of Free Floating Phase