Development of a Light-Weight Manipulator System for an Omni-Directional Multi-Rotor Platform

by

Zachary Guy

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Mechanical Engineering

The Faculty of Engineering and Applied Science

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

October 2020

THESIS EXAMINATION INFORMATION

Submitted by: **Zachary Guy**

Master of Applied Science in Mechanical Engineering

Thesis title: Development of a Light-Weight Manipulator System for an Omni-Directional Multi-Rotor Platform

An oral defense of this thesis took place on October 22, 2020 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Xianke Lin
Research Supervisor	Dr. Scott Nokleby
Examining Committee Member	Dr. Remon Pop-Iliev
External Examiner	Dr. Andrew Hogue, Faculty of Business and Information Technology

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

This thesis presents the development of a proof-of-concept physical prototype and control system for a six degree-of-freedom manipulator attached to an omni-directional multi-rotor system known as the OmniRaptor. This prototype was developed in order to provide the OmniRaptor with the ability to precisely position a sensor in environments where wind gusts are present. In addition to its ability to complete disturbance rejection using visual servoing, the manipulator system provides the potential for pick-and-place operations. Physical testing of the prototype verified its ability to detect and track objects in both stationary and moving configurations as well as operate in conjunction with the OmniRaptor while in manual and autonomous flight modes.

Keywords: manipulator; omni-directional; UAV, visual servoing

Author's Declaration

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Zachary Guy

Statement of Contributions

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The application of Unmanned Aerial Vehicles (UAVs) has grown over the years with advancements in control as well as robotic systems. A portion of these applications include visual inspections, object delivery, data collection, and object manipulation. As mobile ground platforms have been coupled with a manipulator or robotic arm, the same is being done with UAVs. Coupling the two robotic systems increases the applications of these mobile platforms. These applications many include taking the human element out of dangerous tasks such as inspecting and repairing electrical towers, where workers would have to climb the structure, or the inspection of bridges for cracks and other damage.

This thesis focuses on the development of a mobile manipulator designed for a omni-directional UAV platform to perform maintenance and inspection tasks. This platform is cable of autonomously flying to and landing on structures to perform inspections and simple maintenance operations with the operator remaining in a safe location.

1.1 Background

1.1.1 The AMS Project

This thesis is a portion of a larger project known as the Aerial Manipulator System (AMS) project being developed at the Mecatronic and Robotic Systems (MARS) Laboratory at Ontario Tech University. The intention of the AMS project is to develop a unmanned aerial system to perform inspection and maintenance tasks in hard to access and hazardous locations such as electrical towers and remote construction sites. This system is comprised of a central UAV that is tethered to an Aerial Support Vehicle (ASV) that transmits additional power to the UAV. The UAV a gripper mechanism designed for landing and perching on structures along with a manipulator for structural repairs. The layout of the system is shown in Figure 1.1. This project is described in further detail in [1] and [4]. For the AMS, the UAV platform is used as

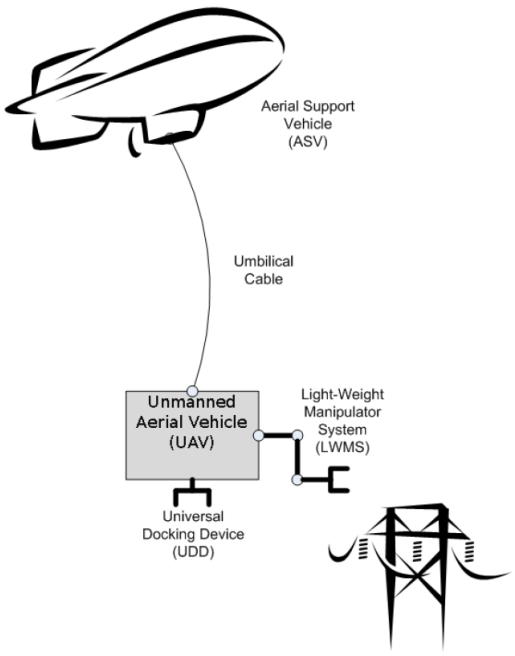

Figure 1.1: Overview of the AMS Project [1]

the mobile platform for the manipulator system in this thesis. The multi-rotor used

is known as the OmniRaptor. The OmniRaptor is equipped with four rotors parallel to the ground plane and 4 variable pitch rotors perpendicular to the ground plane that are able to change the direction of thrust. This configuration allows for omni-directional control and advanced trajectories that a conventional multi-rotor cannot achieve. The OmniRaptor is also equipped with a gripper mechanism that acts as landing gear as well as allows for the ability of grasping objects. The OmniRaptor is capable of advance trajectories that a normal multi-rotor cannot achieve and is able to perch on inclined and flat beams. The OmniRaptor and the its gripper mechanism are discussed in detail in [5].

1.2 Thesis Problem Statement and Goals

This thesis focuses on the design, development, and testing of a proof-of-concept manipulator system for object tracking and general manipulation. This system is known as the Light-Weight Manipulator System (LWMS).

The inspection process of structures using a UAV requires a dexterous platform that is able to navigate complex trajectories, manipulate on board sensors, and complete simple maintenance tasks. The biggest issue when it comes to inspections and maintenance is that the sensing equipment needs to be in close proximity to the area. In addition, the sensor must be held at a certain position and orientation relative to its target for a period of time. If the sensor is rigidly attached to the UAV, the senor will not be placed close enough to the area or if its extended outward the total space the system occupies increases which would be dangerous for operation. In addition to sensor placement, the operator also needs to identify to the UAV a point of interest via an on board camera. With the features listed above, the LWMS provides a live view to the operator and can identify and the position and orientation of features for the manipulator to track. The manipulator must be able to follow and track an

object presented by the vision system. Finally, the LWMS must be coupled to the dynamics and control system of the OmniRaptor.

The goal of this thesis was to design, develop, and test a control system and proof-of concept robotic manipulator specifically designed for basic inspection and manipulation of objects. The manipulator was designed to be mounted on a mobile UAV platform, the OmniRaptor, and coupled with its control system. In addition, a secondary goal was to develop an object tracking control system to ensure a sensor mounted on the end-effector of the manipulator would be able to stay on target in adverse conditions..

1.3 Summary of Contributions

The main contributions of this thesis are:

- Development of a proof-of-concept light-weight manipulator designed to be mounted on the OmniRaptor platform
- Development of a joint-space control system to control the manipulator.
- Development of a velocity-based control system to control the manipulator.
- Development of a control model to visualize the OmniRaptor and manipulator system.
- Application of object detection to an on-board vision sensor.
- Development of a tracking control system that allows the robotic manipulator to be controlled via an on board vision sensor.
- Integration of the LWMS on the OmniRaptor.

- Ground testing of the coupled robotic manipulator and OmniRaptor platform for object tracking and model synchronization.
- Flight tests of the coupled robotic manipulator and OmniRaptor platform for off-board control and basic object tracking.

1.3.1 Thesis Outline

The remainder of this thesis will proceed as follows: Chapter 2 contains a literature review on existing UAV mobile manipulator platforms and previous work done on this project. In addition, the theoretical background behind the design of a light-weight mobile manipulator system (LWMS) is presented. Chapter 3 provides detail on the physical design and construction of the LWMS prototype. Chapter 3 also discusses the integration of the LWMS with the OmniRaptor system. Chapter 4 outlines the development of the various control algorithms designed for basic control and disturbance rejection using visual servoing. Chapter 5 discusses the testing of the LWMS and the OmniRaptor as one complete system. Finally, Chapter 6 contains the concluding remarks and recommendations for potential future development and improvement of both the OmniRaptor and the prototype manipulator system developed during this thesis.

Chapter 2

Literature Review and Previous Work

This chapter provides an in-depth literature review on the use of manipulator systems on aerial platforms. This chapter also provides a basic overview of the previous work done on the manipulator portion of the AMS project by other members of the MARS Lab.

2.1 Literature Review

This literature review is split into separate sections related to this thesis. These sections include a review of the latest advancements in aerial manipulators and the method of control of these platforms.

2.1.1 Aerial Manipulators

A variety of aerial manipulators have been presented that attempt to solve the problem of sensor placement, aerial grasping, and force exertion on external objects. These

manipulators have been both coupled with omni-directional and conventional multi-rotor platforms in addition to other aerial vehicles.

2.1.2 Coupling Manipulator to Platform

In terms of physically attaching the manipulator to the platform two methods were found. The first being that the manipulator is rigidly attached to the aerial vehicle and the second being the manipulator is attached through a tethered platform. Manipulators that are rigidly attached to the aerial platform have the intention of interacting with an object by either grasping or applying an external force to that object. In the case of a tethered platform, it was found that these systems were used to position sensors as well as low force grasping operations while not being effected by the disturbances produced by the aerial platform. As the positioning of the manipulator is task dependant it was found that the majority of the manipulators researched were positioned at the bottom of the UAV.

The issue with rigidly attaching a manipulator to an aerial platform is the restriction of work space due to the risk of collision with the rotors on the aerial platform. This problem is solved in one of the platforms presented in [6] with the use of a 2-DOF (degrees-of-freedom) manipulator consisting of a revolute joint surrounding the main body of the multi-rotor and a prismatic joint used for sensor placement. This allows for 360° of rotation about the body of the platform which solves the limited work space problem presented in the majority of aerial manipulators. This large work space cannot be achieved in a stationary configuration as only two dimensions of travel for the sensor would exist. In order to achieve this large work space the manipulator and UAV platform must work together. A similar design was found in [7] where a omni-directional platform was coupled with a 4-DOF manipulator consisting of all revolute joints. Similar to the system found in [6], the base joint was able to rotate

360° relative to the UAV platform. This system was used for advanced disturbance rejection where the goal was to keep the end-effector of the manipulator in a stationary position as the UAV platform completed complex trajectories.

Tethered mechanisms were presented in which a UAV platform would be connected to an aerial manipulator via a cable system. As these systems are not able to exert large external forces on an object they solve the problem of instabilities caused by object manipulation and grasping due to the dampening of force through the tethered mechanism. This system was found in [8,9] and a hybrid system found in [10] where a UAV coupled manipulator mounted to the bottom was tethered to an addition multi-rotor platform. A variety of applications of tethered platforms can be found in [11–13], in addition, [14] contains the latest advances in load transportation with tethered payloads . As these platforms are stand alone tethered aerial manipulator systems, the application of multiple tethered systems working together has been utilized in [15–18] for cooperative load transportation.

The most common method of aerial manipulator platforms consist of a multi-DOF manipulator composed of revolute and prismatic joints in various configurations mounted to the bottom of the UAV platform. This common configuration was found in [7, 19–32] with the application of aerial grasping, sensor positioning, and force exertion on external objects. In the platform presented in [30] a perching manipulator was used in conjunction with a manipulator for object interaction. Perching manipulators are not in the scope of this thesis. The common layout of manipulator rigidly mounted to the bottom of an aerial platform is expanded on in [6] where two standard multi-rotors were outfitted with dual arm manipulators, one with passive joints and the other with actuated joints. The use of dual manipulators allows for simple grasping and force exertion over a surface which cannot be achieved with a single manipulator. In addition to this application, two 3-DOF manipulators were

presented in [33] which were used for valve turning. The end-effectors used in the platforms listed in this literature review either contained a claw like gripper actuated by a single motor, a sensor, or a combination of both.

2.1.3 Method of Control

In terms of control of the aeiral manipulator and the attached UAV platform, two main approaches were found to control the system. A centralized approach were the manipulator and UAV are controlled as a single system and a decentralized approach where the two are seen as two independent systems.

Centralized Method of Control

In this method of control a dynamic model was found to be created using the standard Euler-Lagrangian formalization where a set of symbols represent the dynamics of the model in the form of a matrix [34] or the Newton-Euler method where forward and inverse kinematic equations are developed [35]. Due to computational expenses the Newton-Euler method was found to be the most commonly used method of control.

The main problems that arise for the task of aerial manipulation is not the model but the method to control that model. To solve the centralized modeling problem many control architectures were found. On a helicopter manipulator platform a simple full-state feedback linear quadratic regulator was designed near the equilibrium point of the whole dynamic system [27]. An adaptive sliding mode controller was used for a 2-DOF manipulator paired with a quad-copter was used in [26]. A back stepping-based controller for a 7-DOF manipulator that uses the coupled full dynamic model in [28], in addition to a admittance controller for the manipulator arm. Lastly, a serious of coupled equations of motion for a quad-rotor with a 3-DOF manipulator was found in [29].

Decentralized Method of Control

In this approach the two systems are separate and are modeled and controlled using known methods. For the manipulators found in [4, 24, 32], an inverse kinematic (IK) solver was used via the use of ROS and MoveIt!. While the systems in [6, 30, 31] used a custom IK solver and model built using Python and C++ languages. In the case of the multi-rotor, standard flight controllers were used and controlled with sensor data from external motion capture systems, an on-board Inertial Measurement Unit (IMU), cameras and other standard sensors found on a UAV [30]. The process of tuning the PID parameters on the platforms was done in a similar fashion, were the platform would be able to compensate for changes in the centre of gravity as well as wrenches produced by the manipulator and in a similar case for the manipulator for disturbances caused by the UAV.

The aerial manipulator systems listed in this literature review commonly used a motion capture system to command the position and attitude of the aerial manipulator platform in an indoor environment. In the platforms presented in [6], two of the platforms were tested in an outdoor environment using known terrain data in addition to a motion capture system.

2.2 Previous Work

Previous work has been done for this project by other members of the MARS Lab and students at Ontario Tech University. This work was expanded on to produce a functional prototype system for the OmniRaptor project.

2.2.1 The Light-Weight Manipulator System

The LWMS was originally designed by a group of fourth-year undergraduate students for their 2015-2016 Capstone Design Project. This system was comprised of only a 6-DOF robotic arm that presented a solution for a light-weight manipulator for a UAV platform (see Figure 2.1). There were several issues that needed to be addressed to integrate the system with the OmniRaptor.

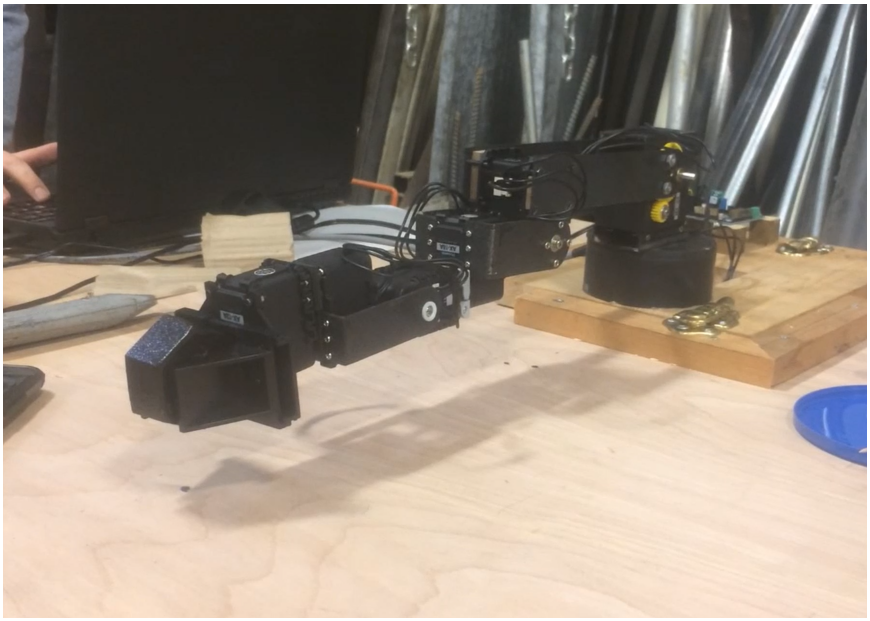

Figure 2.1: The LWMS Prototype

One of the main issues with the LWMS is the control method used to position the joints of the manipulator. The manipulator is controlled via an ArbotiX-M and was loaded with a closed loop C++ program based on Bioloid firmware and a software

control stack. A Python script was then executed that allowed the manipulator to perform a series of preset demo poses. However, real-time control was not achieved, thus a inverse kinematic solution was needed for the manipulator.

Another issue to address was the off-board control of the manipulator. Previously a predefined script was uploaded to the controller and then executed via a laptop connection USB (Universal Serial Bus). For the application to the OmniRaptor on-board control is needed and should remain functional if signal from the base station is lost. A base station computer used by an operator should relay the status of the LWMS as well as provide feedback of the on-board sensors related to the OnmniRaptor and LWMS. In addition to on-board control, the system was in need of feedback to position the end-effector of the manipulator for the intent of object tracking, object interaction, and various other applications.

The last issue to solve was the physical design of the manipulator. The links connecting each of the actuators of the manipulator were fabricated out of carbon fibre and epoxy. Although this provided a light-weight solution the links were very brittle and not easily reproduced upon failure. In addition, a gear box was introduced in the shoulder joint of the manipulator which allowed for increased load capacity of the system, but the gears used were plastic and would fail after repeated use.

2.2.2 OmniRaptor

The OmniRpator is the second prototype streaming from the OmniCopter developed by Florentin von Frankenburg [36, 37]. The OmniCopter proved that an omni-directional concept was viable with the ability to perform stable inclined attitude position holding and landing. There were many issues associated with the OmniCopter that were then solved with the OmniRaptor. The initial physical design for the OmniRaptor was done by Scott Kazinsky as a summer student in the MARS

Lab and then further developed by Christoper Baird into a complete working prototype with an increased payload, omni-directional, control and the ability to land and perch on inclined beam members with unknown geometry. A detailed description of the OmniRaptor and its capabilities can be found in [5]. A side-by-side comparison between both prototypes is shown in Figure 2.2.

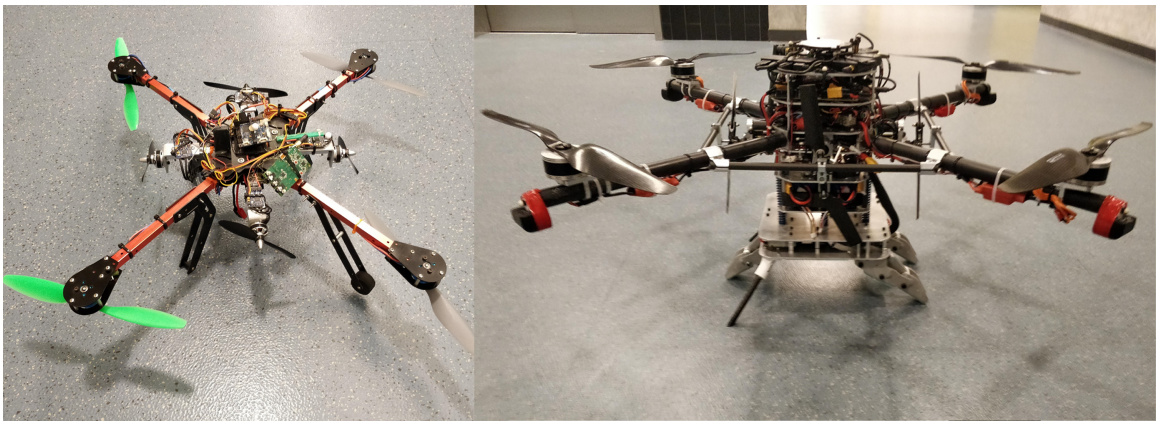

Figure 2.2: The OmniCopter Prototype (left) and the OmniRaptor Prototype (reft)

The only issues associated with integrating the LWMS with the OmniRaptor is the limitation of the physical work-space of the manipulator due to the risk of the manipulator interfering with the rotors of the platform. In addition to limiting the physical interaction between the two systems, the control logic used for the real-time control needed to be intergrated into the on-board computer of the OmniRaptor and not hinder the processing power used to control the OmniRaptor.

Chapter 3

Prototype Physical Design

This chapter provides a detailed overview of the physical design and electrical systems behind the proof-of-concept manipulator system developed in this thesis. Within this chapter can be found the specific physical and functional requirements of the prototype system and an overview of the prototype's physical design and the hardware components used in its construction.

3.1 System Requirements

The OmniRaptor is a fully functional robotic platform intended for indoor use within a laboratory setting. The specific requirements of the LWMS have been divided into physical and functional requirements.

3.1.1 Physical Requirements

The following is a list of the physical requirements that the LWMS must fulfill:

- The LWMS must be designed to interface with the OmniRaptor UAV platform.
- The total weight of the LWMS must not exceed 1.5 kg.
- The LWMS must not interfere with the rotors and motors of the OmniRaptor.
- The vision and manipulator system must be a compact package so the dexterity of the OmniRaptor is not impacted
- The LWMS must have a working payload 0.15 kg.
- The LWMS must have 6-DOF.

The payload of 0.15 kg was chosen based on the application of the manipulator working with bolts that are typically found on electrical towers. These bolts can weigh 0.085 kg for a 5/8' x 1-1/2' bolt [38].

3.1.2 Functional Requirements

The following is a list of the functional requirements that the LWMS must fulfill:

- The LWMS must be able to operate while the OmniRaptor is in control autonomously or manually by an operator.
- The LWMS must be able to detect and track objects.
- The LWMS must be able to position a sensor to track a specified object.

3.1.3 Assumptions

A number of assumptions were made in order to develop the LWMS presented in this thesis. The assumptions made are as follows:

- The object/feature to be tracked by the system is known.
- The object to be tracked will remain in a stationary position.
- The OmniRaptor will hold its position, i.e., station keep, as the LWMS tracks the designated object.

3.2 Prototype Overview

The LWMS design was done with the Computer Aided Design (CAD) software package Autodesk Fusion 360. The result of the final CAD model of the prototype LWMS can be seen in Figure 3.1.

The LWMS contains 6-DOF robot manipulator consisting of revolute joints (R) in a $(R \perp R \parallel R \perp R \perp R \perp R)$ configuration. This layout was chosen because the manipulator was required to travel and rotate in the x, y, and z planes which is considered to be a fully free manipulator. 6-DOFs is the minimum amount required for a manipulator to fulfill the task of rotating and traveling along each axis. In addition, this layout provides the rotation of the end-effector for added dexterity,

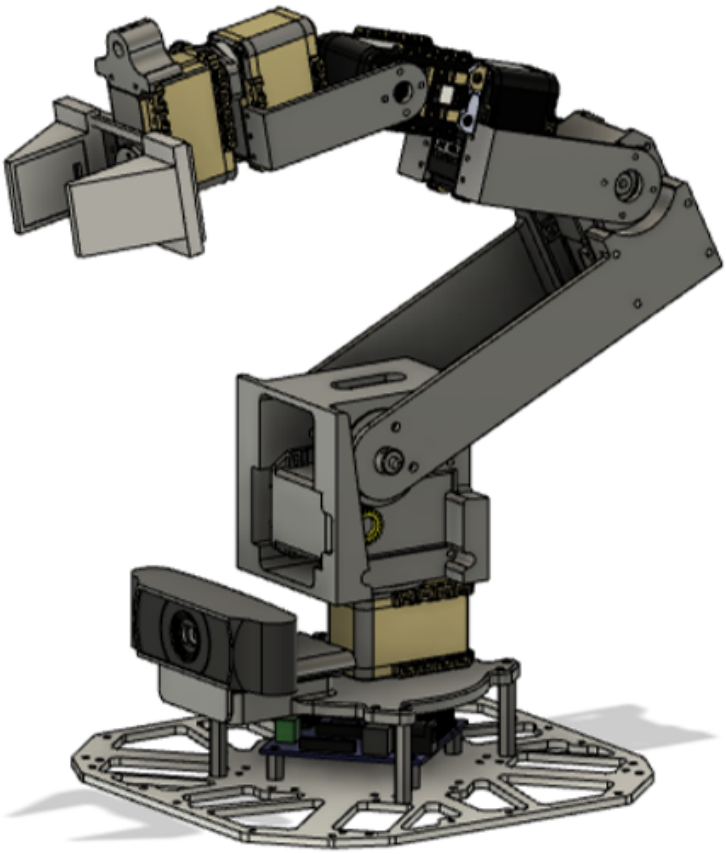

Figure 3.1: CAD Model of the LWMS

as the intent of this system is to track objects and manipulate small objects such as screws. The system also has a simple webcam used for object tracking, a parallel gripper as an end-effector, and a red laser to visualize object tracking and manipulator movement.

The LWMS is compromised of six servo assemblies to control the 6-DOF. The shoulder of the manipulator is directly connected to a base plate that contains the vision sensor, that plate is then rigidly connected to the OmniRaptor via standoffs. A laser is rigidly mounted to the end-effector, with an offset relative to the centre, to depict the end-effector's relative x, y, and z movement on a moving or stationary target. The entire assembly fixed to the OmniRaptor is displayed in Figure 3.2.

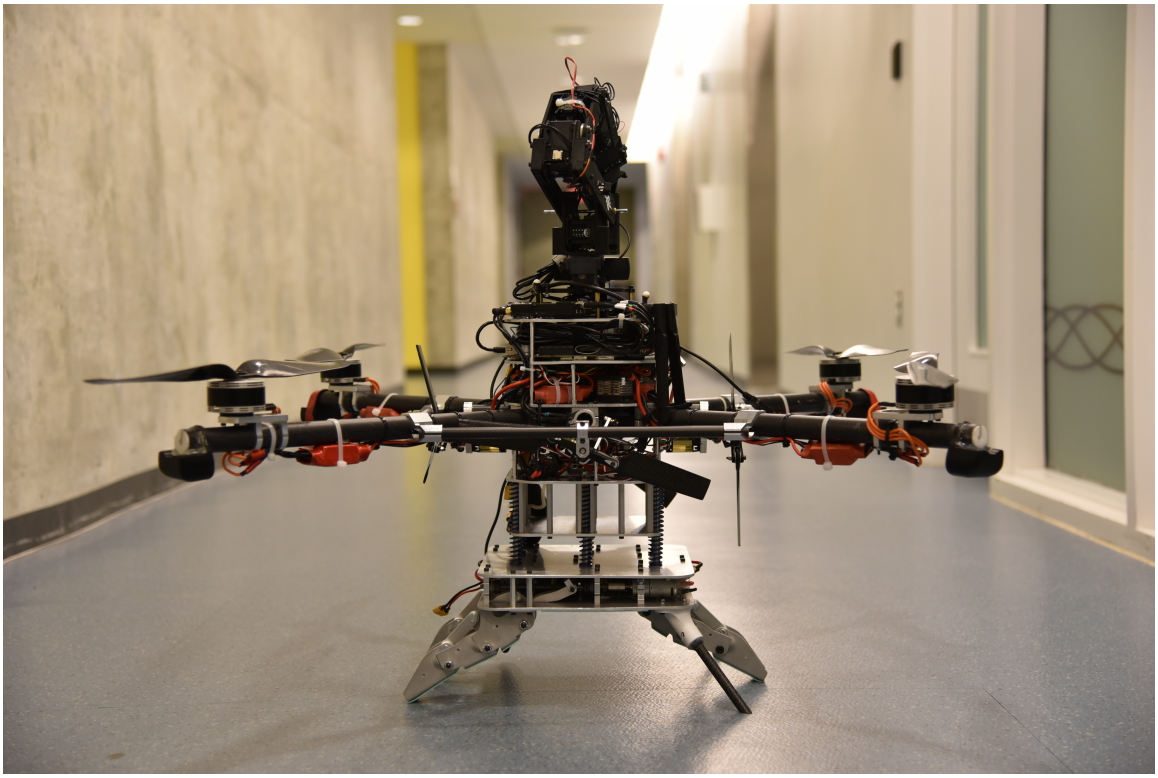

Figure 3.2: LWMS Mounted to the OmniRaptor

3.2.1 Physical Prototype

As the main design parameter of the manipulator was that it was to be lightweight, this needs to be defined. The operational payload of the OmniRaptor was not tested but with the results provided in [5], the OmniRaptor with a mass of 11.75kg and additional 8 kg of weight in the form of a steel plate (19.75 kg total), the OmniRaptor was able to achieve flight at 55% throttle. Comparing the total mass of the LWMS to this tested flight weight of 19.75 kg, the is 5.3% of the OmniRaptor and the additional weight. As it is ideal that OmniRaptor to fly at 50-60% throttle during operation, the manipulator system can be considered lightweight. A carbon fiber design was the initial choice of material but due to manufacturing limitations and the desired accuracy of the manipulator, a more accurate and repeatable method of manufacturing was chosen. All links of the LWMS were designed using CAD software and fabricated

using a 3D printer with polylatic acid (PLA) plastic as the material of choice. As the LWMS is a prototype, many changes were done to its design which resulted in manufacturing multiple versions of parts. The method of manufacturing used allowed an efficient process of iteration of the LWMS. The total mass of the LWMS presented in this thesis is 1.05 kg. In addition to the system being lightweight, an additional requirement was that the work-space of the manipulator should not interfere with the rotors of the OmniRaptor. To prevent this, joint limits were introduced in the control architecture of the manipulator, this is further discussed in Section 4.1.3 of this thesis. In the situation of actuator or controller failure, hard stops were introduced into the main shoulder assembly of the manipulator as seen highlighted in red in Figure 3.3. These hard stops physically limit the work space of the manipulator in case of power failure.

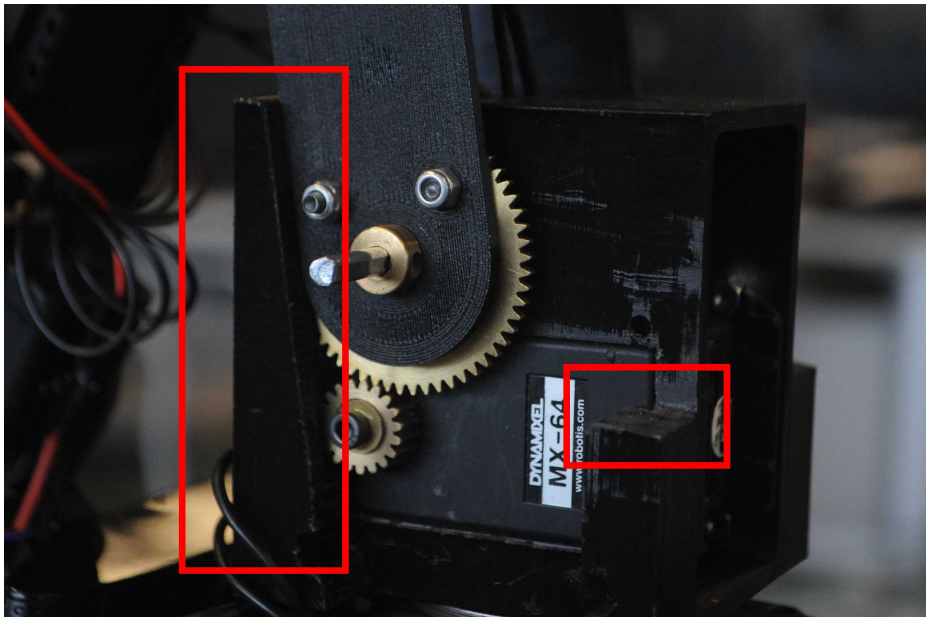

Figure 3.3: Shoulder Assembly Gearbox of LWMS

In addition to the hard stops, in the shoulder assembly of the manipulator a gear set was also introduced. The actuators used in small scale manipulators such as the one in this thesis are ideal for lightweight applications, but they have limited torque capabilities resulting in a small payload for the system. The moment produced by the

shoulder actuator is limited by the weight and payload of the manipulator. To combat this, the shoulder assembly contains a gear set that increases the torque produced by the shoulder pitch joint. The gear reduction used in this assembly was a 1:3 gear ratio resulting in a maximum working payload of 0.2 kg, exceeding the desired payload of 0.15kg. As the hard stops limited the work space of the manipulator, a visualization and of the works space is shown in Figures 3.4 and 3.5. This work space was chosen due to the structural limitations of the OmniRaptor and the propellers of the system.

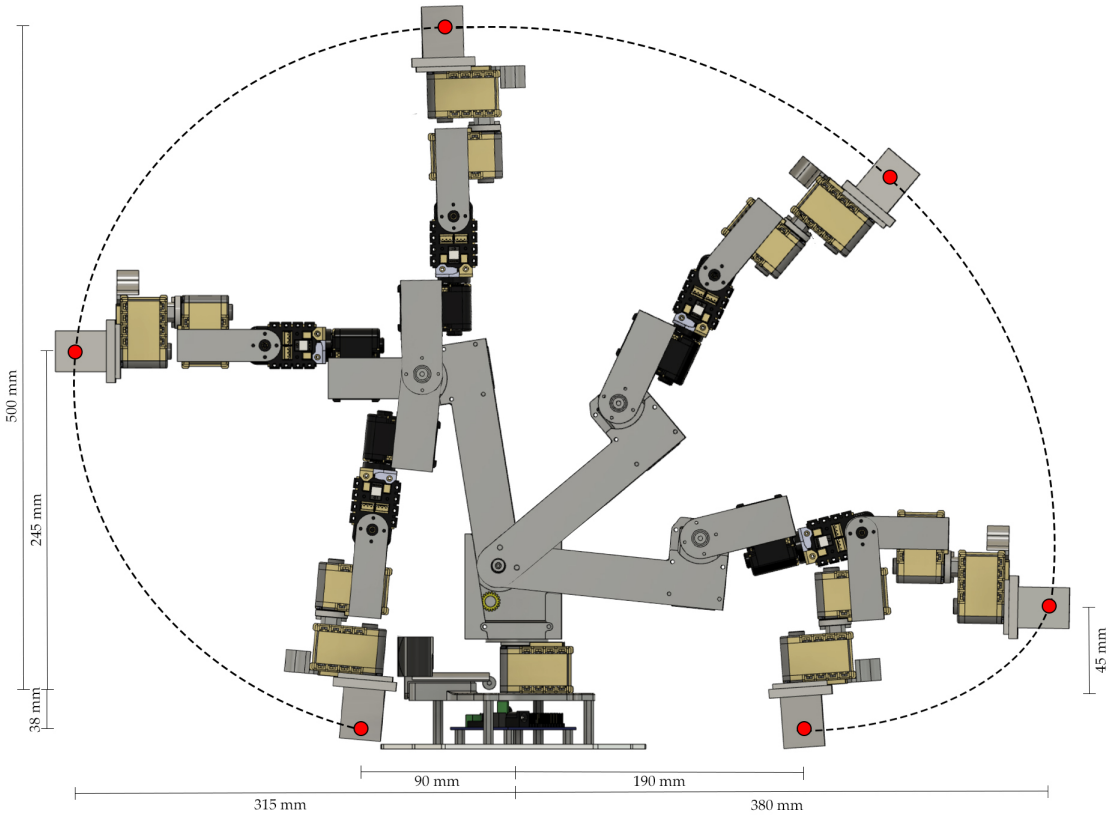

Figure 3.4: Side Profile of the LWMS Work Space

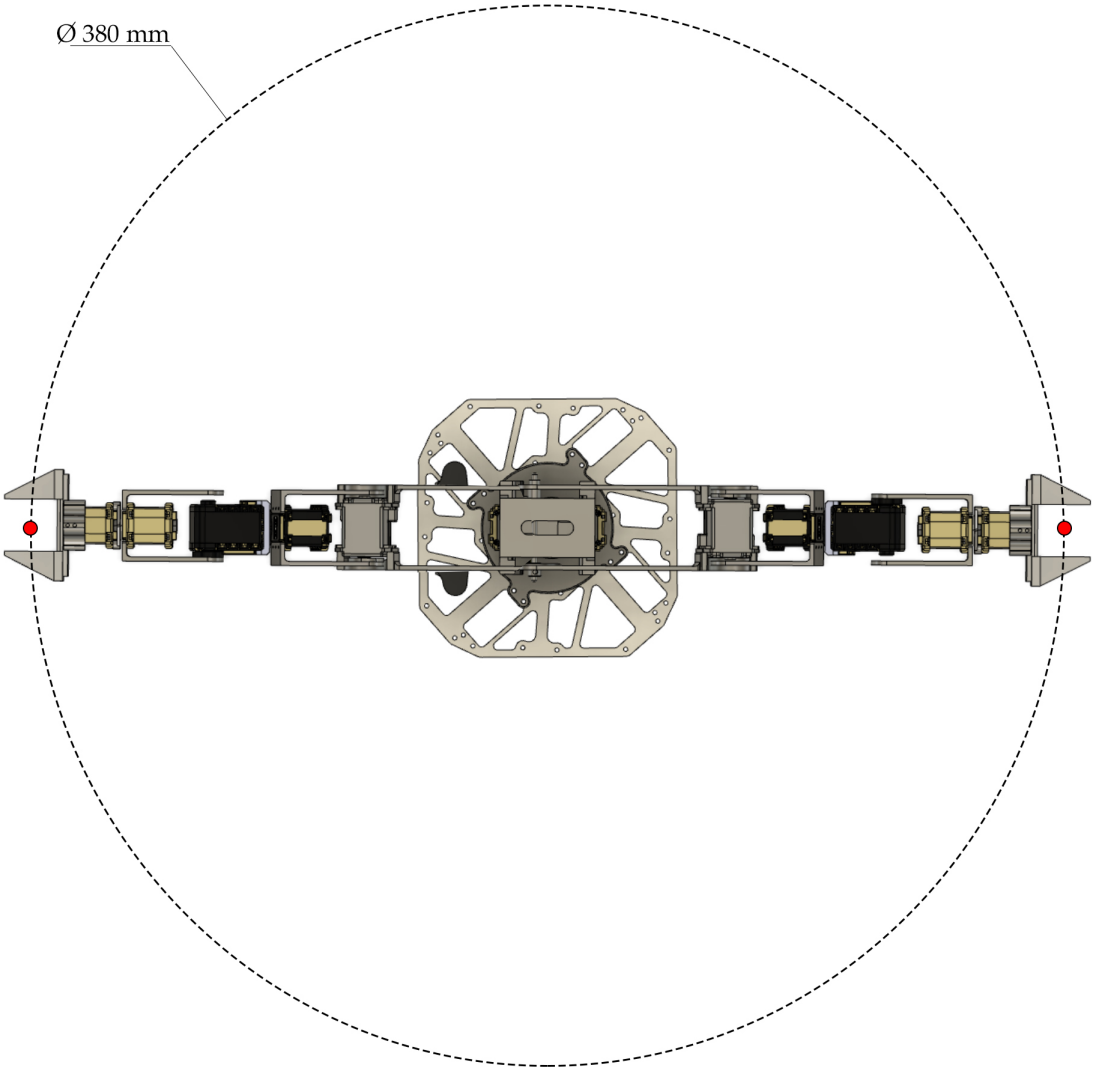

Figure 3.5: Top Down Profile of the LWMS Work Space

The end-effector of the LWMS is a commercially available design used in the PhantomX Pincher and WidowX arms developed by Trossen Robotics. It is a simple parallel gripping mechanism that is driven by an Dynamixel servo. As the intention of this system is to complete object detection, object tracking, and simple operations like manipulating a screw head, this style of end-effector was chosen for the LWMS. In addition to the gripper, a red 5V laser in line with the centre of the end-effector was added to visualize the relative position of the end-effector on a desired target. The end-effector's assembly is shown in Figure 3.6.

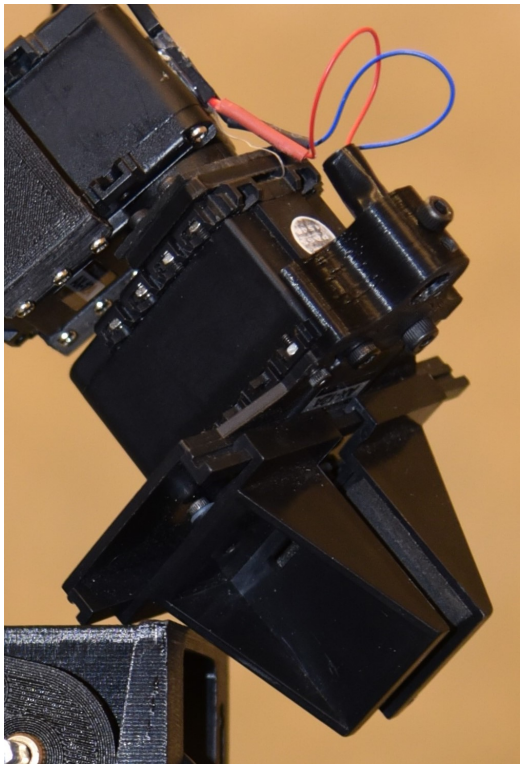

Figure 3.6: End-Effector Assembly of the LWMS

3.2.2 Actuators and Electrical Systems

As the LWMS is to be mounted to the OmniRaptor and controlled by its on-board computer, the electrical system of the LWMS needed to be integrated. The OmniRaptor's computer is a Nvidia Jetson TX2 and was used for its high computing power

and small scale. The computer is utilized to run the real-time kinematic solver for the manipulator as well as the object tracking algorithm later discussed in Section 4.1 of this thesis. To control the servos, an Arbotix-M micro-controller was used. In normal configurations this controller can be used to drive and program the servos for planned servo positions and raw servo values. For the application of this thesis, the micro-controller was used as a motor driver and commands were sent via the OmniRaptor's on-board computer through a USB connection.

The actuators in the LWMS consist of Dynamixel MX-64, AX-12a, AX-18a servos. In general manipulator assemblies, the same actuator is used throughout each joint.In the design of this manipulator, different actuators were used at different joints due to the moments produce at each joint. The strongest of the actuators, the MX-64, was used in the shoulder pitch joint, as this joint had the largest moment produced as well as radial load due to its location. The AX-18a servo was used as the shoulder pan servo, this has similar properties to the AX-12a actuators but offer a greater maximum axial load. To increase the axial load of the shoulder pan servo a thrust bearing was used. The remaining joints of the manipulator contain the AX-12a servos as they produce the needed torque and joint rates required for the LWMS. The properties of the servos used in the LWMS operating at 12V are displayed in Table 3.1.

Table 3.1: Actuator Properties of the LWMS

Actuator	Resolution (degrees)	Stall Torque (Nm)	Radial Load (N)	Axial Load (N)	No Load Speed (rev/min)
MX-64	0.29	6.0	40	20	63
AX-18a	0.29	1.8	30	15	97
AX-12a	0.29	1.5	25	10	59

Dynamixel servos are a system developed to be used exclusively connecting joints of a robot or a mechanical structure. The servos are DC motors that contain a reduction gear train, controller, driver, and network connection that allow an all in one solution to a small scale servo actuator [39]. The servos allow for position feedback

that is transmitted to the on-board computer via the motor driver which allows the monitoring and control of the servo values.

The full electrical system for the LWMS is shown in Figure 3.7. The full electrical system of the OmniRaptor can be found in [5].

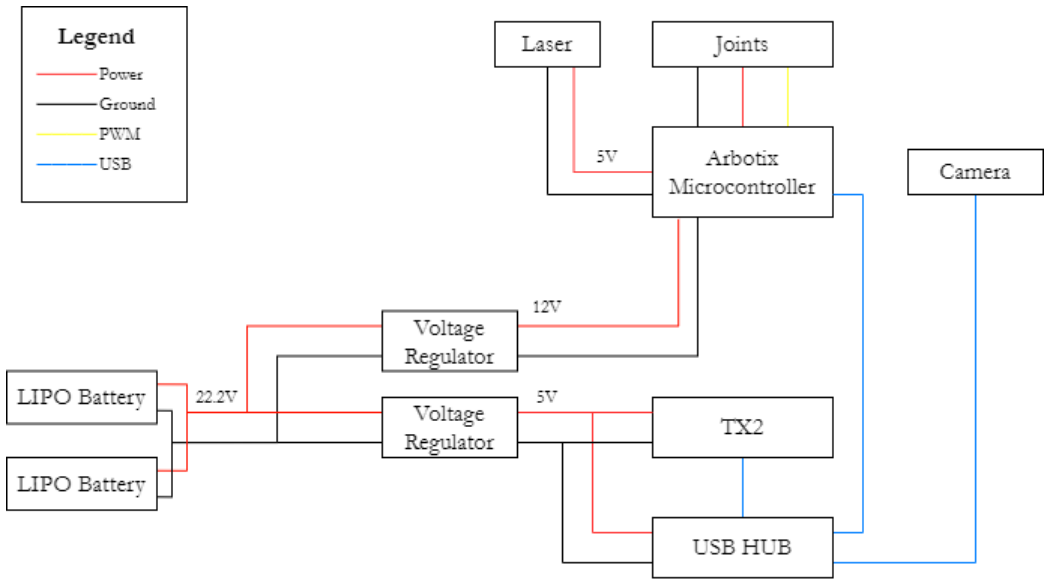

Figure 3.7: The LWMS's Electrical Systems

3.2.3 Vision System

The vision system used in this prototype is a USB Logitech webcam, the specifications of the webcam are listed below:

- 1,920 x 1,080 video resolution.

This camera is rigidly attached to the base of the manipulator and the offset of the camera's sensor to the centre of the shoulder pan joint was recorded. This offset is used in the visual servoing of this system discussed in Section 4.2.2 of this thesis. The webcam is connected to the OmniRaptor's on-'board computer via USB. The layout of the vision sensor is displayed in Figure 3.8.

Figure 3.8: LWMS Vission Sensor: Logitech Webcam

Chapter 4

Control Structure

This chapter outlines the control structure and strategies that govern the operating behavior of the prototype LWMS. This chapter as well outlines the derivation of the governing mathematical equations required to actuate the manipulator with point and velocity commands, the control architecture used for basic point-to-point visual servoing, and a proposed control structure for velocity based visual servoing.

The LWMS is comprised of two main control systems that allow for general movement of the manipulator, object detection, and visual servoing. The general control structure is displayed in Figure 4.1. The OmniRaptor's control structure is simplified in this figure as the focus of this thesis is on visual servoing and control of the LWMS.

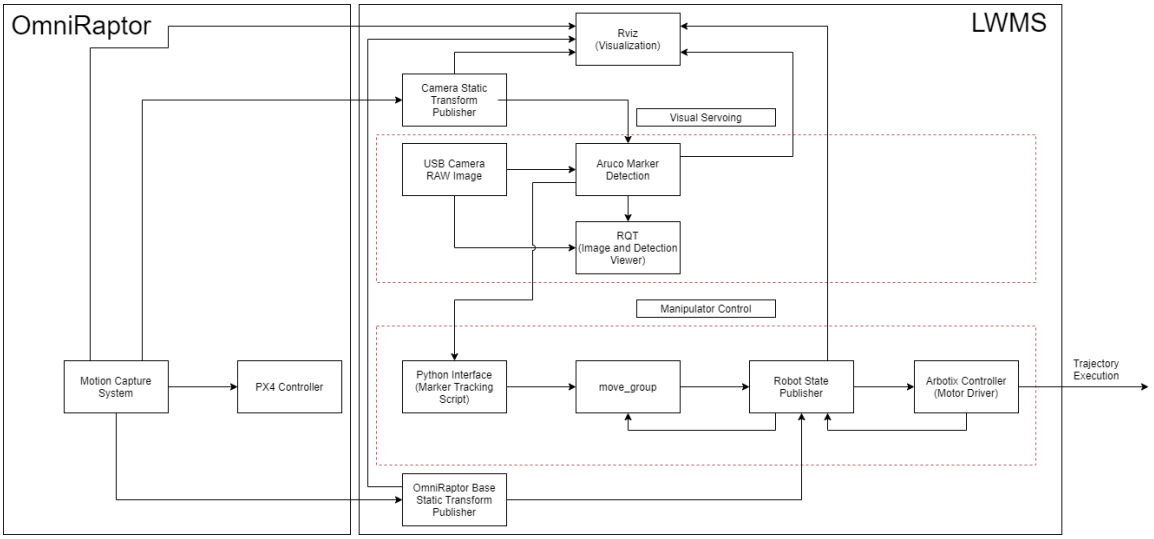

Figure 4.1: Control Structure of the LWMS

The manipulator control is initialized by the Python interface, a script is ran and the marker tracking is initiated. This script samples the position of the aruco marker from the USB camera feed in the visual servoing portion of the control system. The coordinates of the marker frame in addition to a predefined tool orientation of the end-effector are sent to the `move_group` node where the trajectory is planned. The `move_group` node then sends the commands to the robot state publisher which simultaneously sends the visualization of the trajectory and the OmniRaptor's position to the Rviz visualizer. The robot state publisher then executes the trajectory through the use of a motor driver.

4.1 Manipulator Control

In order to control a system of actuators of a manipulator, forward and inverse kinematic equations are needed. To further simplify the LWMS a simple diagram was made to visualize the joint configuration, link lengths, and offsets (a, b, c, and d) seen in Figure 4.2.

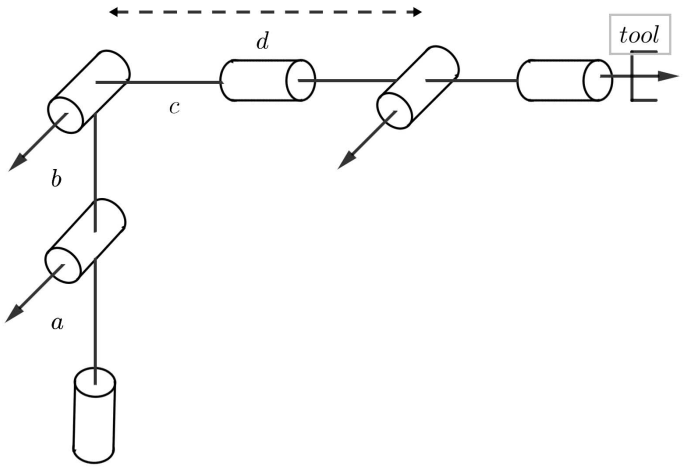

Figure 4.2: General Kinematic Diagram of the LWMS

4.1.1 Forward Kinematics

The kinematic equations of a robot that compute the position of the end-effector from specified values for the joint parameters is known as forward kinematics. To derive an expression that relates each link in a manipulator system, a fixed frame is attached to each link and then a transformation matrix is formed that describes neighboring links. The individual transformation links can be formed to solve for the position and orientation of link n relative to the base frame link. The transformations constructed define the frame i relative to frame $i-1$ and is a function of four link parameters for any given robot. The general transformation matrix for ${}^{i-1}_{i}\mathbf{T}$ is [34]:

$$
{}^{i-1}_{i}\mathbf{T} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & -c\alpha_{i-1} & -c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}
$$

where a is the link length, α is the link twist, d is the link offset, θ is the joint angle, and s and c denote sine and cosine, respectively. These four parameters are known as the Denvait-Hartenberg Parameters (D-H Parameters). To further understand the kinematics of the LWMS the D-H Parameters were identified and the transformation matrix from the base frame to the frame, ${}^{0}_{tool}\mathbf{T}$, was found:

Table 4.1: D-H Parameters of LWMS

$Frame_{i-1}$	α_{i-1}	a_{i-1}	d_i	θ_i	$Frame_i$
0	0	0	0	θ_1	1
1	$\pi/2$	0	a	θ_2	2
2	0	b	0	θ_3	3
3	-$\pi/2$	0	0	θ_4	4
4	$\pi/2$	0	d	θ_5	5
5	-$\pi/2$	0	0	θ_6	6
6	0	l_{tool}	0	0	tool

$$
{}^{0}_{tool}\mathbf{T} = {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T}\,{}^{2}_{3}\mathbf{T}\,T\,{}^{3}_{4}\mathbf{T}\,{}^{4}_{5}\mathbf{T}\,{}^{5}_{6}\mathbf{T}\,{}^{6}_{tool}\mathbf{T}
$$

$$
{}^{0}_{tool}\mathbf{T} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}
$$

where:

$$a_{11} = s_6(c_4s_1 - s_4c_1c_{23}) - c_6((s_5c_1s_{23}) - c_5(s_1s_4 + c_4c_1c_{23}))$$
$$a_{12} = s_6(s_5c_1s_{12} - c_5(s_1s_4 - c_1c_{23})) = c_6(c_4s + 1 - s_4c_1c_{23})$$
$$a_{13} = -c_5(c_1s_{23} - s_5(s_1s_4 + c_4(c_1c_{23})))$$
$$a_{14} = bc_1c_2 - l_{tool}(c_6(s_5c_1s_{23} - c_5(s_1s_4 - c_4c_1c_{23})) - s_6(c_4s_1 - s_4(c_1c_{23})))$$
$$a_{21} = -c_6(s_5(s_1s_{23} + c_5(c_1s_4 - c_4s_1c_{23}))) - s_6(c_1c_4 + s_4s_1c_{23})$$
$$a_{22} = s_6(s_5s_1s_{23} + c_5(c_1s_4 - c_4s_1c_{23})) - s_6(c_1c_4 + s_4s_1c_{23})$$
$$a_{23} = s_5(c_1s_4 - c_4s_1c_{23}) - c_5s_1s_{23}$$
$$a_{24} = ac_1 - l_{tool}(c_6(s_5(s_1s_{23}) + c_5(c_1s_4 - c_4s_1c_{23})) + s_6(c_1c_4 + s_4s_1c_{23})) + d(c_1c_4 + s_4s_1c_{23}) + bc_2s_1$$
$$a_{31} = c_6(s_5c23 + c_4c_5s_{12}) - s_4 - s_6 - s_{23}$$
$$a_{32} = -s_6(s_5c_{23} + c_4c_5s_{23}) - c_6s_4s_{23}$$
$$a_{33} = c_5c_{23} - c_4c_5s_{23}$$
$$a_{34} = bs_2 + l_{tool}(c_6(s_5c_{23} + c_4c_5(s_{23}) - s_4s_6s_{23}) + ds_4s_{23}$$

4.1.2 Inverse Kinematics

The problem that inverse kinematics solves is the needed joint angles which achieve a desired tool orientation and position. The problem of finding the required joint angles to place the tool frame is solved in two parts. Like in forward kinematics, the frame transformation from the base link to the tool is computed and then inverse kinematics is applied to solve for the joint angles [34].

As there are many ways to solve the inverse kinematic equations, in this thesis Robot Operating System (ROS) and *MoveIt!*'s *IKFast* plugin were utilized to generate solutions for the complex equations involved with the inverse kinematics. The structure of *MoveIt!* and ROS is further explained in Section 4.1.3 of this thesis.

4.1.3 Inverse Kinematics and *MoveIt!*

ROS is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms [40]. For this thesis ROS's

MoveIt! library was utilized, which is specifically designed for robotic manipulator control.

The *MoveIt!* Library

MoveIt! incorporates the latest advances in motion planning, manipulation, 3D perception, and kinematics. The application of *MoveIt* to this thesis was the motion planning library. This allows the generation of high-degree of freedom trajectories through cluttered environments and avoiding singular configurations. *MoveIt!* uses an analytical inverse kinematics generator to produce a solution for a desired trajectory. It is not trivial to create hand-optimized inverse kinematics solution for manipulators that can capture all singular configurations, thus having the inverse kinematics in closed-form drastically speeds up the process [41].

Rviz and *MoveIt!*

To interact with the physical manipulator a user interface was needed. Built in to ROS's framework, Rviz is a visualization library that allows the user to control and plan paths for manipulators and other robotic platforms. In order to control the LWMS, a Unified Robot Description Format (URDF) file was created from the CAD model of the LWMS. A URDF files is an XML file format used in ROS to describe the links and joints of the robot and how they interact with each other in a kinematic chain. Using *MoveIt!*'s Setup Assistant a Sematic Robot Description Format (SRDF) file was created to generate a self-collision matrix, planning groups, transformation frames, and other configuration files needed to operate the robot. The generated interactive model of the LWMS is displayed in Figure 4.3.

Visual representation of the LWMS is useful, but the control of the manipulator was needed. Displayed in Figure 4.4 is the system architecture for the primary node provided by *MoveIt!*. This node gathers all the individual components together to

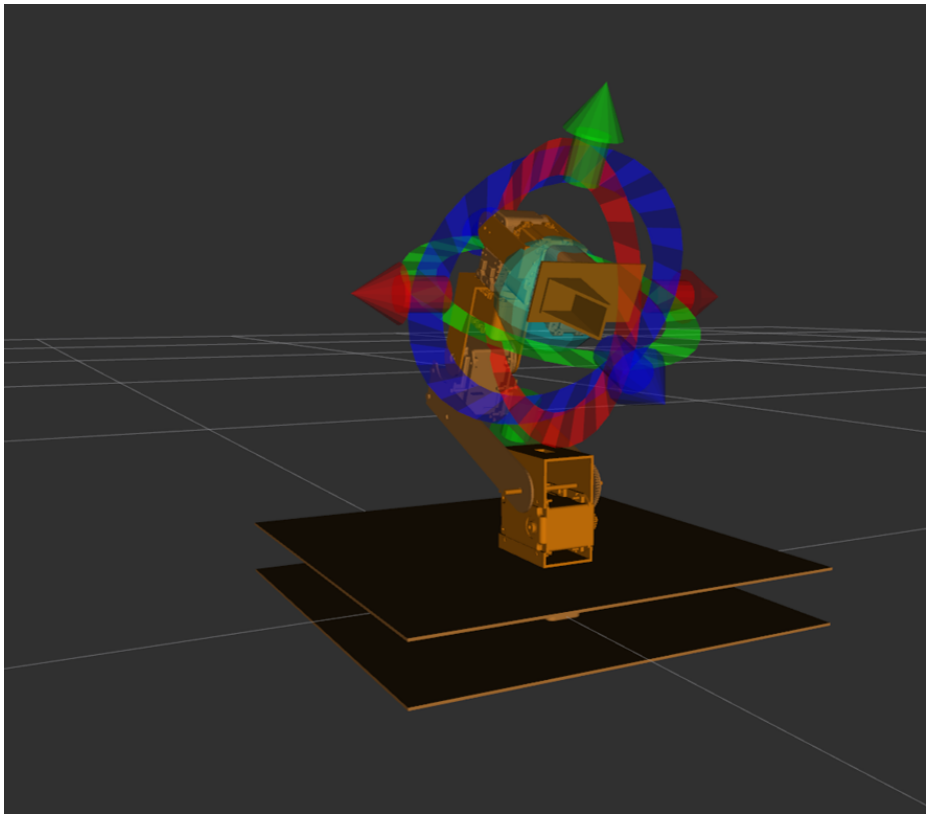

Figure 4.3: Interactive Model of the LWMS

provide a set of ROS services for the user to use [2]. The user can access the actions and services provided by the `move_group` in multiple ways:

- C++: `move_group_interface` where specific commands and functions can be used written in C++
- Python: `moveit_commander` where specific commands and functions can be used written in Python
- GUI: Motion Planning Rviz Pluggin which is a ROS visualizer

In this thesis the Rviz plugin was used for general joint space control. Joint space control is the process of breaking down a desired movement task into discrete motions that satisfy movement constraints and possibly optimize some aspect of the movement. A goal state is specified and the positions, velocities, and accelerations are

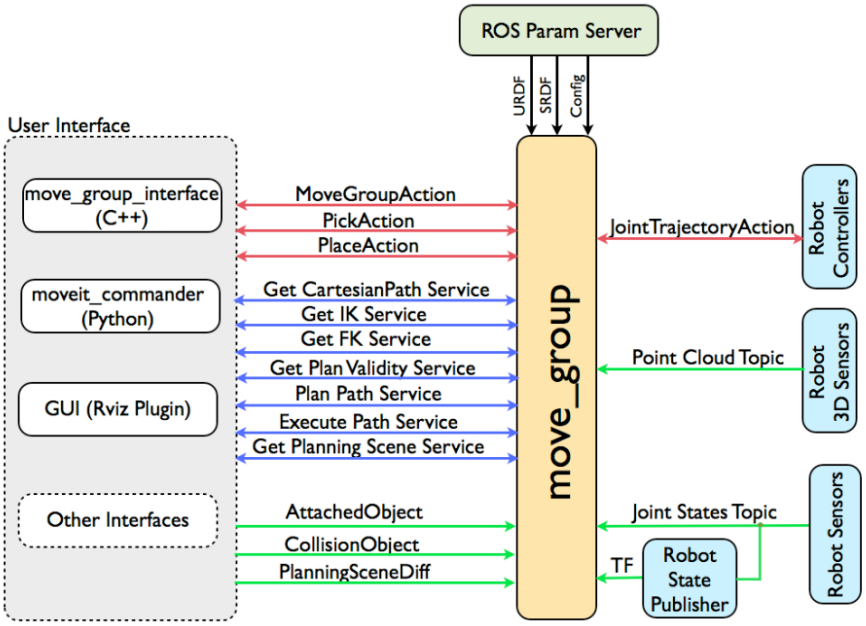

Figure 4.4: `The move_group` Node [2]

calculated with the given constraints in place [34]. By default if the user inputs a goal state and plans and executes the trajectory in Rviz, *MoveIt* executes the trajectory in joint space. An example of a joint space planned path is displayed in Figure 4.5.

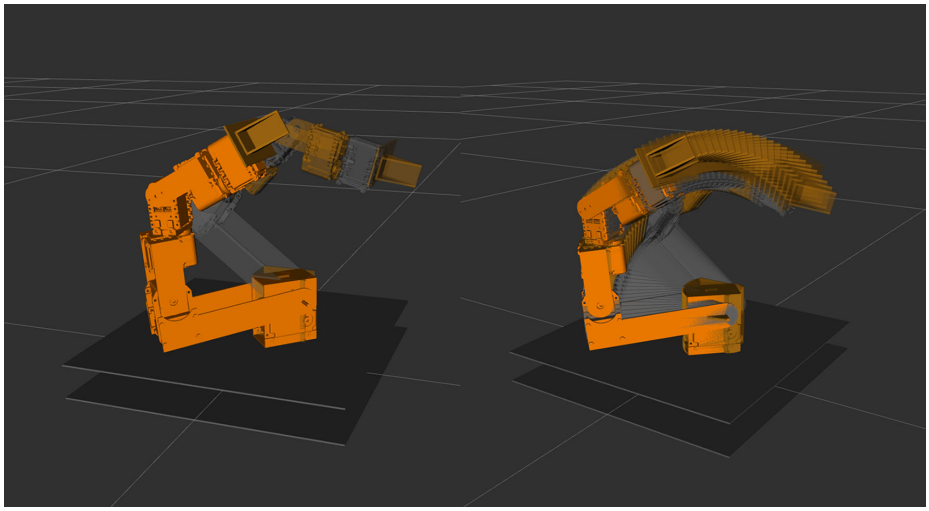

Figure 4.5: LWMS Planned Path: (Left) Start and Goal State, (Right) Path Execution

Although the use of the movement Rviz pluggin is useful for visualization and demonstration purposes, an automated method of control was needed. This was achieved using the Python *Moveit!* Interface also known as `moveit_commander`.

***MoveIt* Python User Interface**

In order to command the robot for the application of visual servoing for this thesis the `moveit_commander` python interface was used. This allows the user to complete pick-and-place operations and execution of Cartesian paths with predefined functions. In this thesis the command line interface was used in ROS to complete the goal of object tracking which is discussed in Section 4.2.2.

4.1.4 Velocity Control of the LWMS

The control structure presented in Section 4.1.3 was used for the results displayed in Sections 5.2 and 5.3. The marker was successfully tracked as displayed in Figure 5.3, but there was a delay from when the desired position is fed into the controller and the end-effector reaches that goal. This delay is due to the fact that the controller is taking the current position of the $marker\_frame$, calculating the joint trajectories, accounting for singular configurations, and then executing the trajectory via the motor driver, also known as point-to-point tracking. To solve the delay in tracking a velocity controller was introduced with the goal of generating smooth continuous tracking of the marker.

The first controller implemented in the LWMS was a joint-space controller that would find the a smooth trajectory to an input given within its work space while avoiding self collisions and singularities. A real-time controller that was needed is known as a Cartesian space controller. The idea behind Cartesian space control is to abstract away from the generalized coordinates of the system and plan a trajectory in a coor-

dinate system that is directly relevant to the task to be performed. In the case of this thesis that is to give linear commands to the end-effector in the x, y, and z Cartesian system of the base of the LWMS.

Velocity Control Using ROS

With the intent of achieving the goal of continuous path planning a velocity controller was developed using the *ros_control* package. This package provides a set of effort, joint state, position, velocity, and joint trajectory controllers for a ROS controlled robot platform. The layout of the control package can be seen in Figure 4.6.

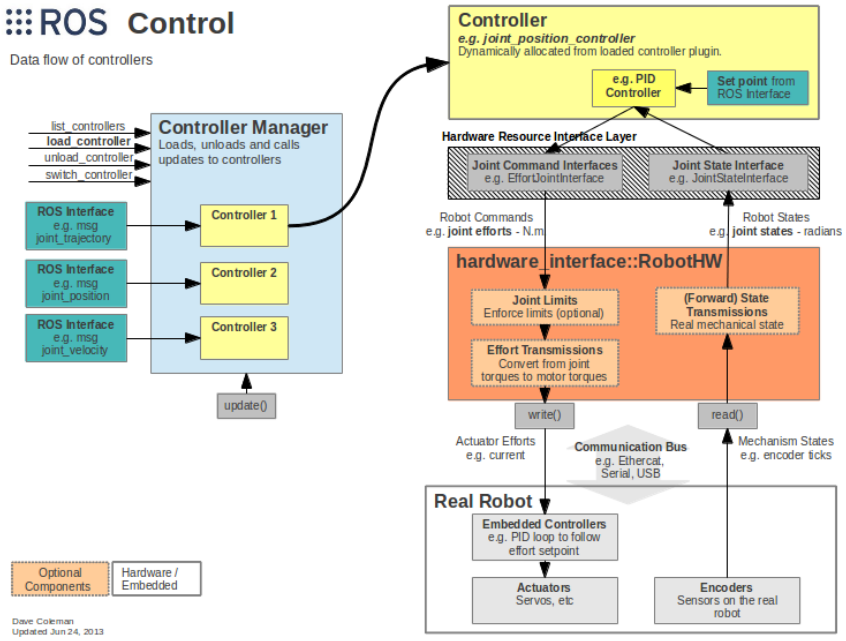

Figure 4.6: *ros_control* Structure [3]

The controllers and interface provided by *Moveit*! are still used but the method of control is changed. The controller manager allows access to a specified controller the package provides, handles resource conflicts between controllers, and can stop or start controllers with a specified run-time. The package uses a generic control loop feedback mechanism (PID) to control the output, in this case velocity, to the

actuators. This is achieved through a similar hardware interface procedure found in Section 4.1.3 where a URDF file was created. This file was modified to provide transmission interfaces between each of the joints, which allows for the creation of effort and flow maps between joints used by *ros_control*. This allows for the use of simple reductions, differential, and linkage transmissions. The transmission of the shoulder pitch joint between the base link and shoulder link with a gear reduction of 1:3 can be seen in Figure 4.7.

```
<!-- SHOULDER PITCH JOINT -->
       <joint name="shoulder_pitch_joint" type="revolute">
              <parent link="shoulder_pan_link" />
              <child link="shoulder_pitch_link"/>
              <origin rpy="0 0 0" xyz="0.0235 -0.0065 0.031"/>
              <axis xyz="0 0 1" />
              <limit effort="300" lower="-1.68" upper="0.25" velocity="5"/>
       </joint>

<!-- Transmission 2 -->
  <transmission name="tran2">
    <type>transmission_interface/SimpleTransmission</type>
    <joint name="shoulder_pitch_joint">
      <hardwareInterface>hardware_interface/PositionJointInterface</hardwareInterface>
    </joint>
    <actuator name="motor2">
      <hardwareInterface>hardware_interface/PositionJointInterface</hardwareInterface>
      <mechanicalReduction>3</mechanicalReduction>
    </actuator>
  </transmission>
```

Figure 4.7: Transmission Between Base Link and Shoulder Pitch Link

The modification of the URDF as well as the *ros_control* package running allows the choice of a set of controllers governed by the controller manager. The *Moveit*! interface is used to visualize the manipulator and the the velocity controller is selected using a series of commands. This allows access to topics that can be published to the manipulator to control the x, y, and z linear and rotational velocity of a desired frame. For the LWMS, the end-effector frame, also known as the *wrist_roll_link*, was the frame of interest. In addition to the generated topics, a Graphical User Interface (GUI) was developed using RVIZ GUI tools to control the manipulator using velocity commands. The GUI can be seen in Figure 4.8.

In addition to the added features, *Moveit*!'s interactive marker and planning interface is still active and can be utilized. The functions of the implemented GUI include:

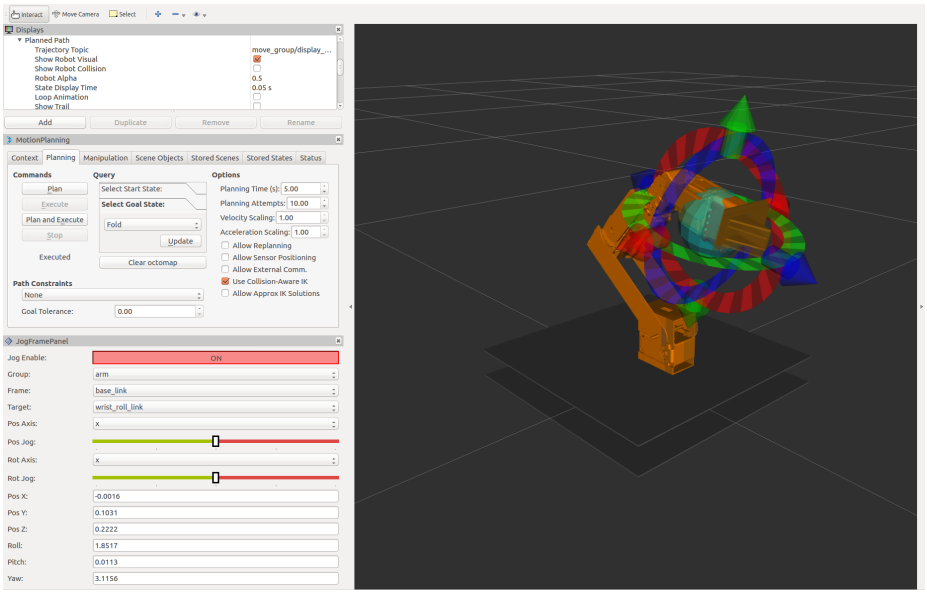

Figure 4.8: RViz Velocity GUI

- ON/OFF Button
- Specified planning frame that allows the user to input commands relative to the selected frame
- Specified command frame that the velcoity commands are given to
- User is able to choose between a rotational or linear velocity
- User is able to choose between a x, y, or z input
- User inputs velocity via a slider bar

4.2 Object Detection and Visual Servoing

It is the intent of the OmniRaptor project to detect features suitable for landing as well as preforming inspection and maintenance with the LWMS. A final goal for the LWMS is to recognize features and feed the relative position to the manipulator for object tracking and disturbance rejection.

4.2.1 Object Detection

As the LWMS is a proof-of-concept system, an object detection algorithm was not used. Instead, to mimic the pose of a detected object or feature a fiducial marker detection package was used. A fiducial marker is an object placed in the field of view of an imaging system for use as a point of reference or a measure. The fiducial marker package used in this thesis is known as the Aruco Library. This library consists of square markers composed of a wide black border and an inner binary matrix which determines its identifier (ID). The black border facilitates its fast detection in the image and the binary codification allows its identification and the application of error detection and correction techniques. An example of several Aruco markers are displayed in Figure 4.9.

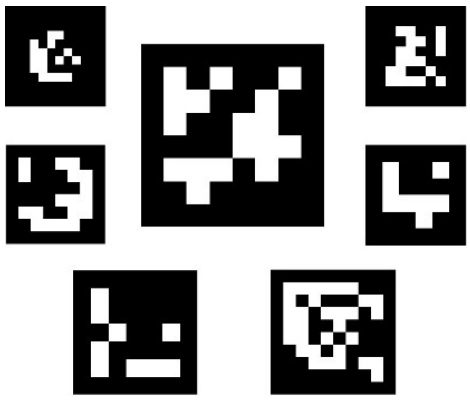

Figure 4.9: Aruco Library Markers

In order to accurately detect and extract the pose from a marker presented to the camera the vision sensor needs to be calibrated. Camera calibration consists in obtaining the camera's intrinsic parameters and distortion coefficients. As the webcam used in this thesis has a fixed focal length these parameters remain constant, thus camera calibration only needs to be done once. The calibration is done using OpenCV in combination with a calibration checker board. The distance between the tiles on the board are known and the board is put in front of the camera and rotated and

moved along each axis in the camera's view. Once the program has gathered enough information, the values to account for the distortion of the camera are recorded and used to accurately detect a Aruco marker. The camera calibration of the webcam used in the LWMS can be seen in Figure 4.10

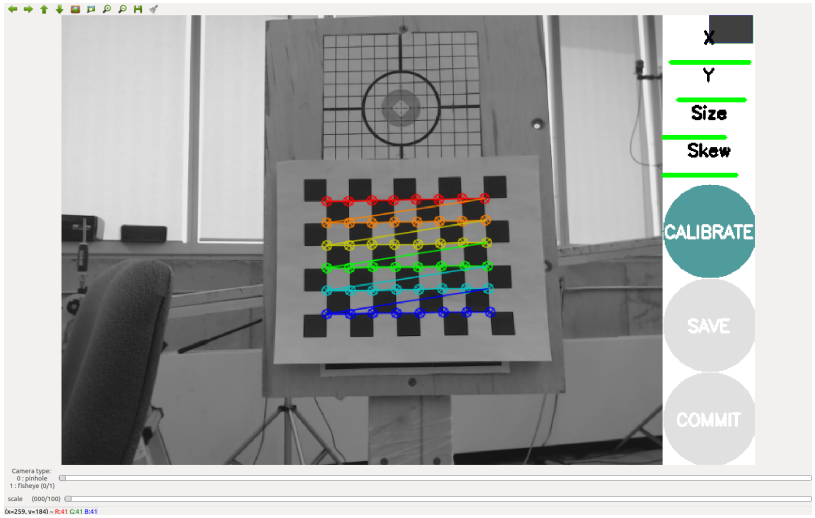

Figure 4.10: Logitech Webcam Camera Calibration

The Aruco library allows for single or multi marker detection as each generated marker has a specific ID. For this thesis a single marker was used with the ID tag of 701. Once the camera was successfully calibrated and the size and ID of the marker were set as parameters in the detection algorithm, the x, y, and z position and orientation (pose) relative to the face of the camera can be extracted. An example of a detected marker in addition to the extracted pose can be seen in Figure 4.11.

4.2.2 Visual Servoing

As the intended goal for the LWMS is to follow the marker, a method of feeding the relative pose of the marker to the command frame of the manipulator was needed. In order to achieve this a set of static and dynamic transformations were used by utilizing ROS's tf package. This package keeps track of mutiple coordinate frames over time. For the LWMS the frames of interest are the marker, camera, and the frames of the

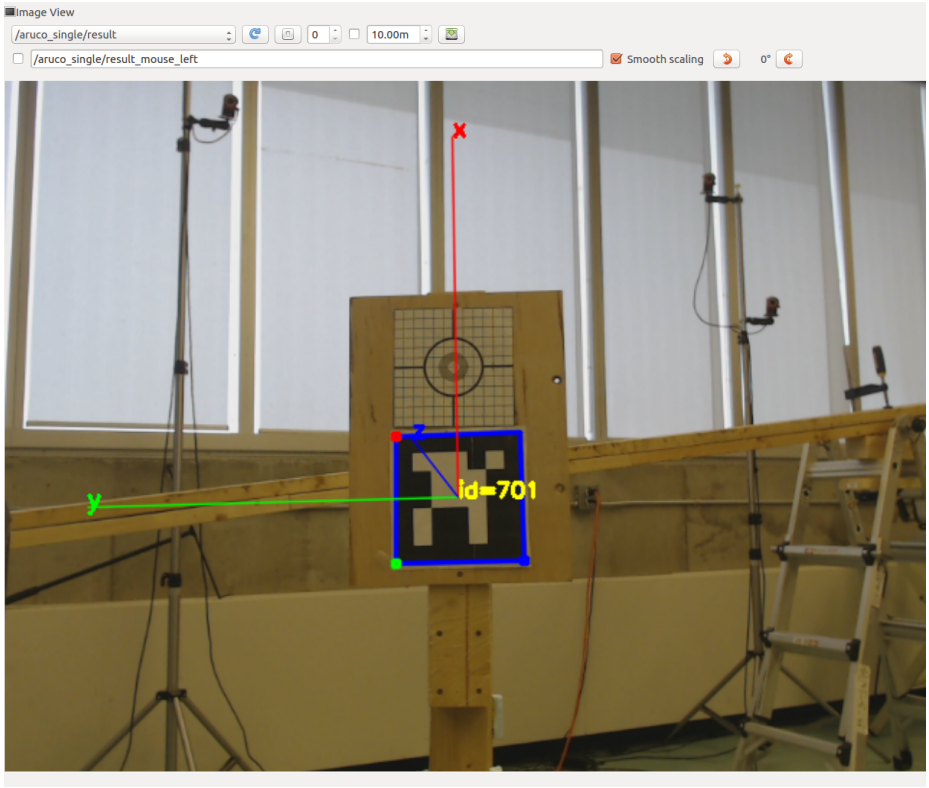

Figure 4.11: Aruco Marker Detection and Extracted Pose

manipulator. The frames of the manipulator were produced using *Moveit*'s setup assistant discussed in Section 4.1.3 of this thesis.

As the vision system is physically connected to the manipulator, a static transform was used to connect the frames of the camera and the base of the manipulator. This was done by using the x, y, and z offsets from the centre of the *base_link* actuator to the face and centre of the camera, also known as the *cam_base_link* frame. As the *base_link* frame is linked to the frames of the manipulator and the camera frame, the pose of a detected Aruco marker ($marker\_frame$) relative to any of the joint or link frames of the manipulator can be extracted. The linked frames can be visualized in RVIZ as seen in Figure 4.12.

The manipulator in the LWMS takes commands relative to the *base_link* frame using the Python interface mentioned in Section 4.1.3. As the *base_link* is the frame of

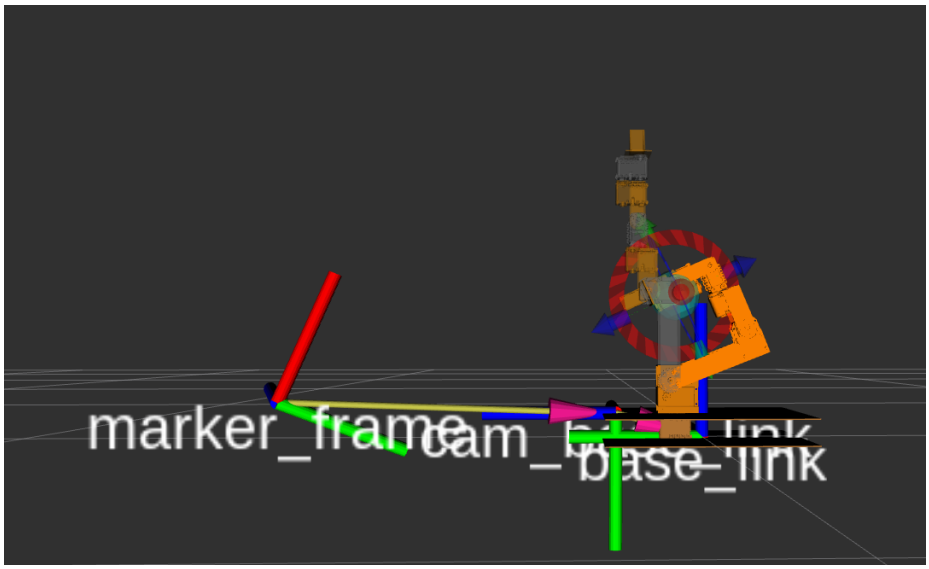

Figure 4.12: LWMS Resultant Frames

interest the pose the marker was extracted to command the manipulator. ROS's framework uses nodes that publish data that can be accessed by the user. The pose of the *marker_frame* was published to the ROS server at a rate of 100 Hz. The data of the extracted pose was then 'subscribed' to and then stored as a set of variables. The data was stored so that the user can add predefined offsets to the pose command of the end-effector. The stored variables allow flexibility of the user in the case of physical contact of a detected marker for pick-and-place operations or marker tracking when a pose offset is used. With the pose offsets applied, the variables are then passed to the controller at a rate of 10 Hz as the goal pose of the manipulator.

4.3 Linking The LWMS to The OmniRaptor

As the LWMS is to be mounted on the OmniRaptor a set of transformations were needed to link the OmniRaptor's frames to the frames of the LWMS. The OmniRaptor's on-board computer (Jetson tx2) was used to control the OmniRaptor as well as the LWMS. In addition to the computer, a flight controller and multiple micro-controllers are used that receive commands from the computer to drive propulsion

systems and the perching mechanism of the platform. To control the platform in an indoor laboratory environment an OptiTrack motion capture system was used. The motion capture system uses an array of 14 infrared (IR) cameras located in various locations about the testing area. The cameras used can be seen in Figure 4.13. Combining the IR cameras, a set of IR beacons on the OmniRaptor, and the OptiTrack software the 6-DOF pose of the OmniRaptor can be extracted and a frame attached for visualization and positioning for autonomous control. This pose is measured relative to a set ground plane and position known as the *map* frame. In addition, to improve the accuracy of the pose, the motion capture data is combined with the accelormeter data from the flight controller to produce even more accurate tracking of the OmniRaptor in 3D space. The frame linked to the OmniRaptor as the command frame was the centre of the flight controller of the platform and is known as *omni*. The entire testing area used in this thesis can be seen in Figure 5.1 in Section 5.1 of this thesis.

Figure 4.13: Opti-Track Camera

The x, y, and z offsets from the centre of the flight controller to the centre of the shoulder actuator of the LWMS (*base_link*) were measured and used as the transformation data to link the LWMS to the OmniRaptor. This allowed the pose of any of the frames of the LWMS or OmniRaptor to be extracted relative to the *map* frame. In

addition, the real-time visualization of the LWMS combined with the Omniraptor was achieved. The resultant frames of the entire system working together with a detected marker can be seen in Figure 4.14. Another example of the real-time visualization of the frames of the LWMS and OmniRaptor can be seen in Figure 4.15, where the system is tilted on its y axis.

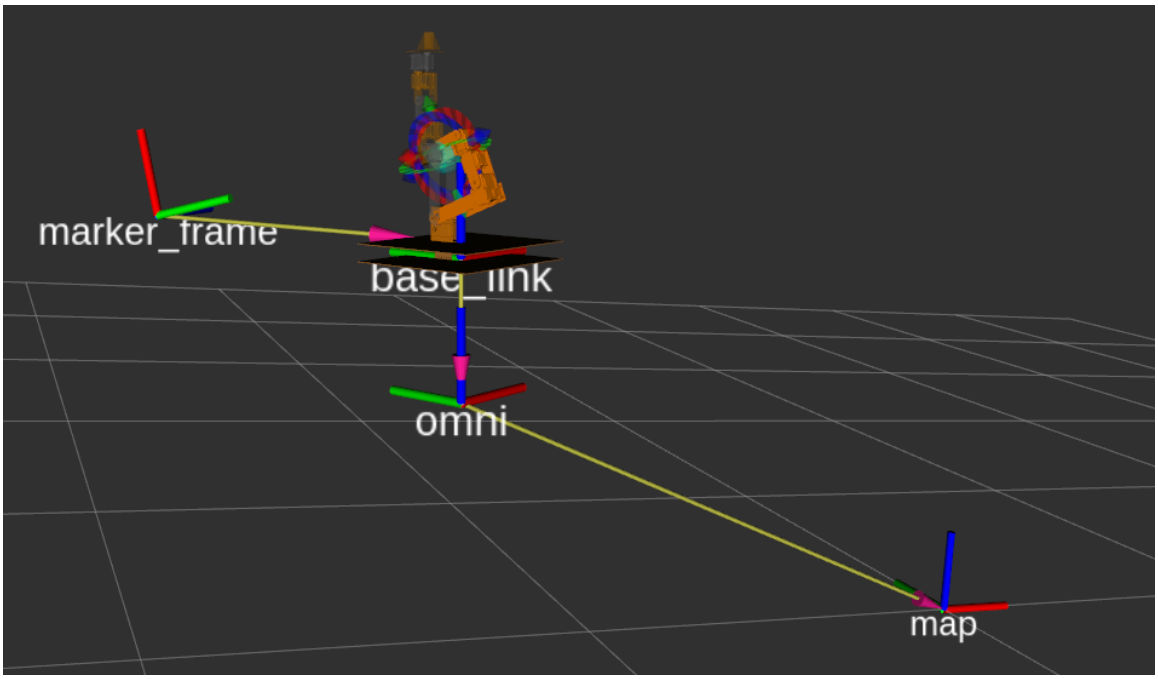

Figure 4.14: Frames of the OmniRaptor and the LWMS

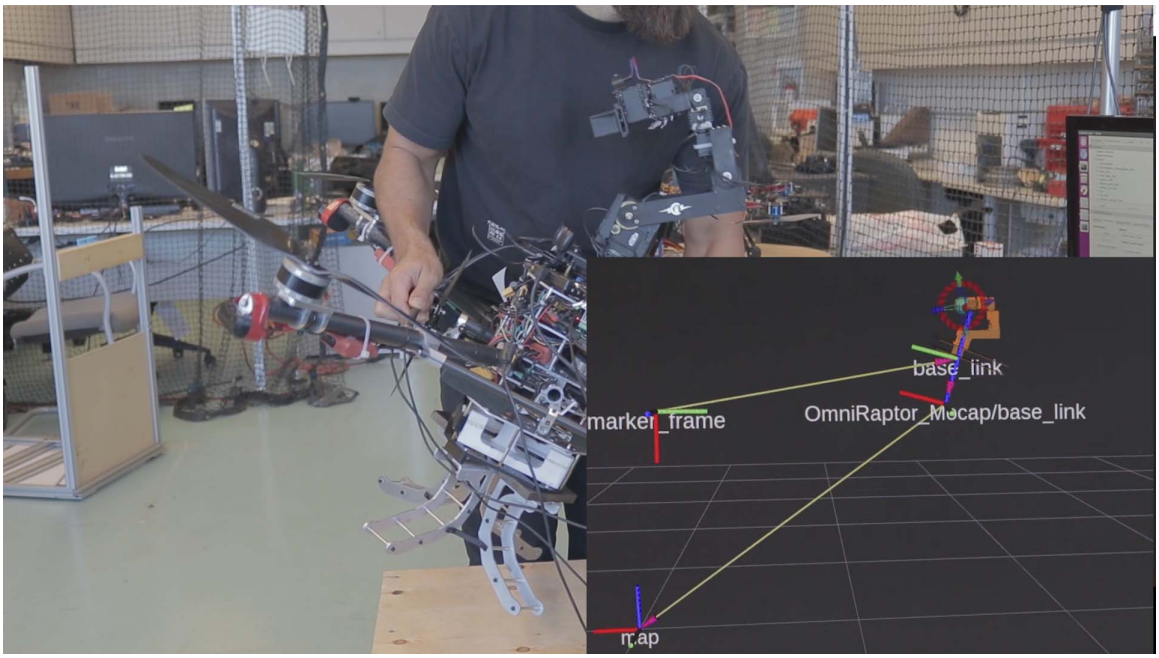

Figure 4.15: Comparison of RViz Visualization to the Real OmniRaptor and LWMS

Chapter 5

Testing, Results, and Discussion

In order to validate the effectiveness of the LWMS developed in this thesis, several tests were conducted to evaluate the systems ability to identify a marker, track the identified marker, and to operate the LWMS with the OmniRaptor while in flight. Testing of the system was done independently from the OmniRaptor as well as with the OmniRaptor in several flight tests. This was done so that the performance results of the LWMS could be compared both with and without the use of the OmniRaptor platform.

5.1 Experimental Setup

This section discusses the details of the experimental setup of the testing of the LWMS in combination with the OmniRaptor. The testing was done in a laboratory setting in a designated area created for flight capable robotic platforms. The test area is netted for safety and includes the motion capture system mentioned in Section 4.3 of this thesis. The test area can be seen in Figure 5.1.

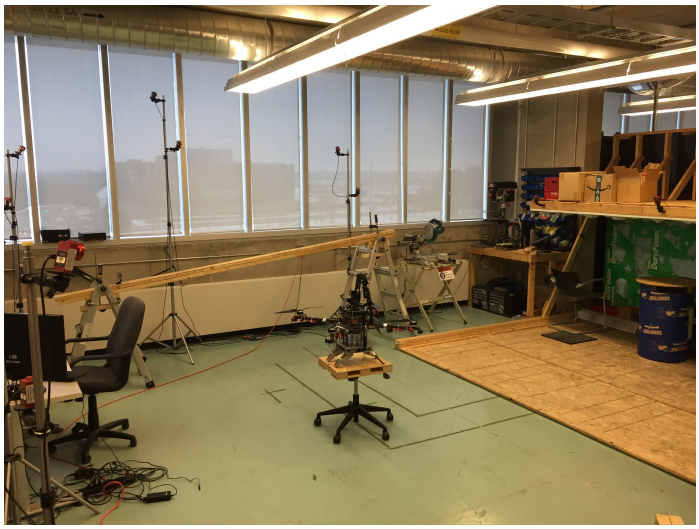

Figure 5.1: Test Area of the OmniRaptor and LWMS

As the goal of the LWMS is to track a designated marker, a cross hair target was attached to a mobile base for stationary tracking and in-flight tracking as discussed in Sections 5.2 and 5.3, respectively. The target mounted to the mobile stand can be seen in Figure 5.2.

As the end-effector is intended to track the marker on the stand, a target was used to visualize the end-effector's relative position via the laser that was mounted to the end-effector with a z offset. The x, y, and z offset from the marker's centre was measured and used in conjunction to the marker's relative position to feed into the controller of the end-effector's position when the marker tracking script was executed.

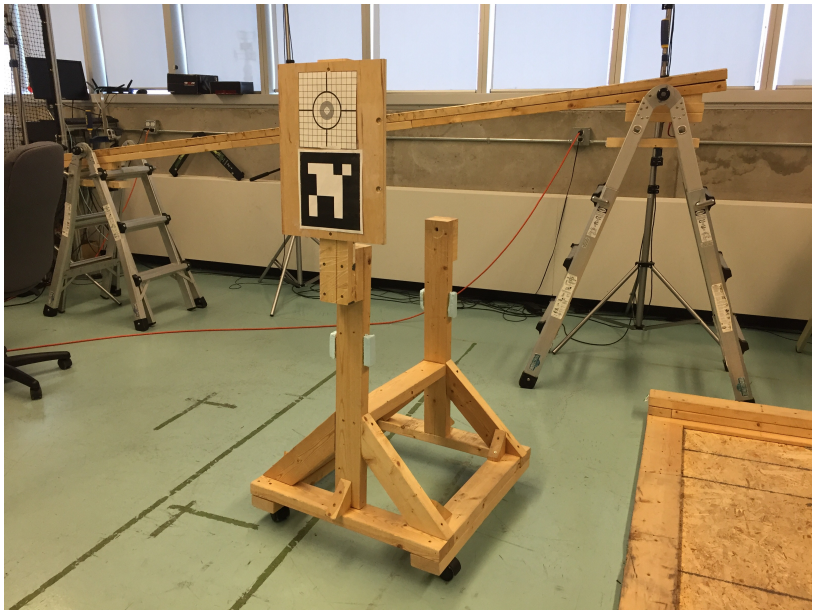

Figure 5.2: Mobile and Stationary Target

The visualization of the end-effector's position when tracking the marker can be seen in Figure 5.3.

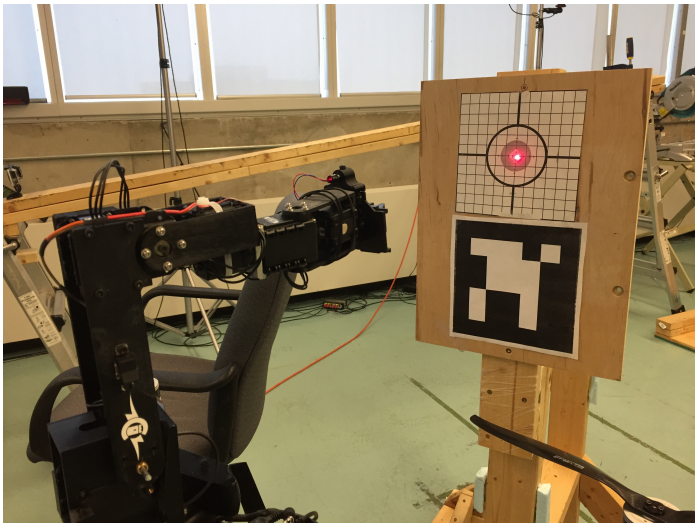

Figure 5.3: End-Effector's Relative Position Visualization

5.2 Stationary Testing

5.2.1 Single Axis Tracking

A set of stationary tests were completed to evaluate the ability of the LWMS to detect and track a target. In this set of tests the LWMS was mounted to the OmniRaptor and linked to the motion capture system of the test area. The OmniRaptor was then held in a stationary position and then the marker was brought into the view of the LWMS via a mobile base. The position commanded to the LWMS's controller was compared to the end-effector's actual position in each of the tests. The first set of tests completed involved isolating each axis to test the functionality of the visual servoing on a single axis. As an example in the testing of the x axis, the marker's relative x position was fed into the controller while constant y and z values were fed into the controller. This test was then repeated for the y and z axes.

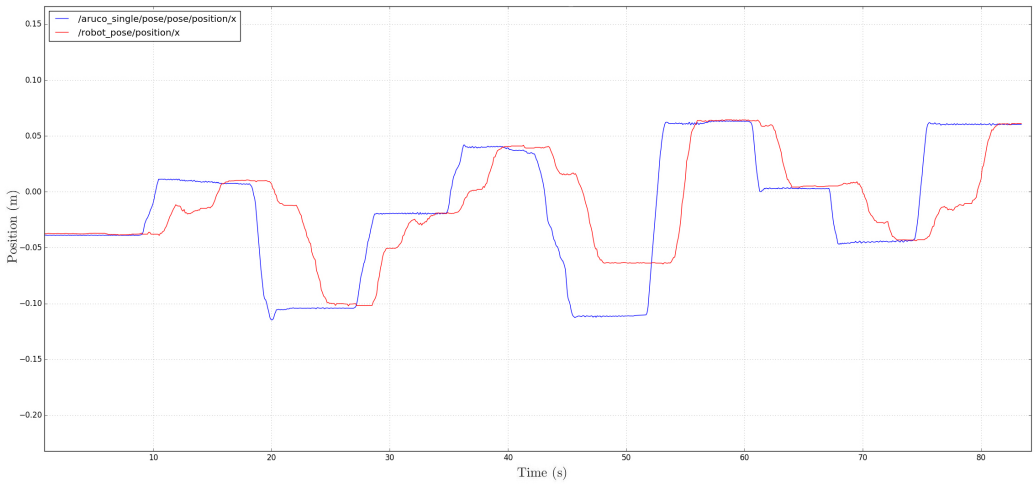

Figure 5.4: x Position Tracking of Marker

The results presented in Figures 5.4, 5.5, and 5.6 show that the marker was successfully tracked in the individual x, y, and z axes. It was also observed that there was a considerable amount of delay from when the position of the marker was fed into the

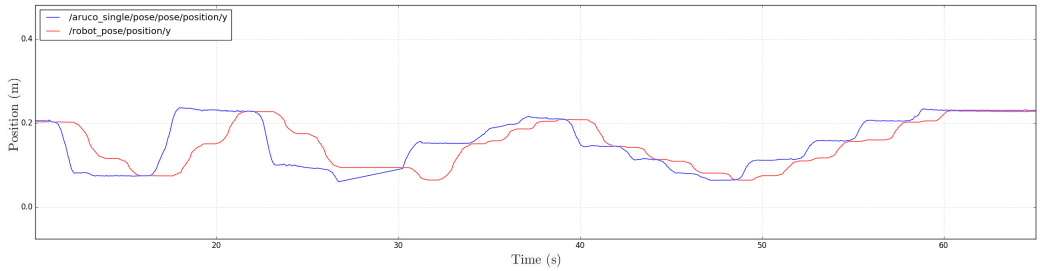

Figure 5.5: y Position Tracking of Marker

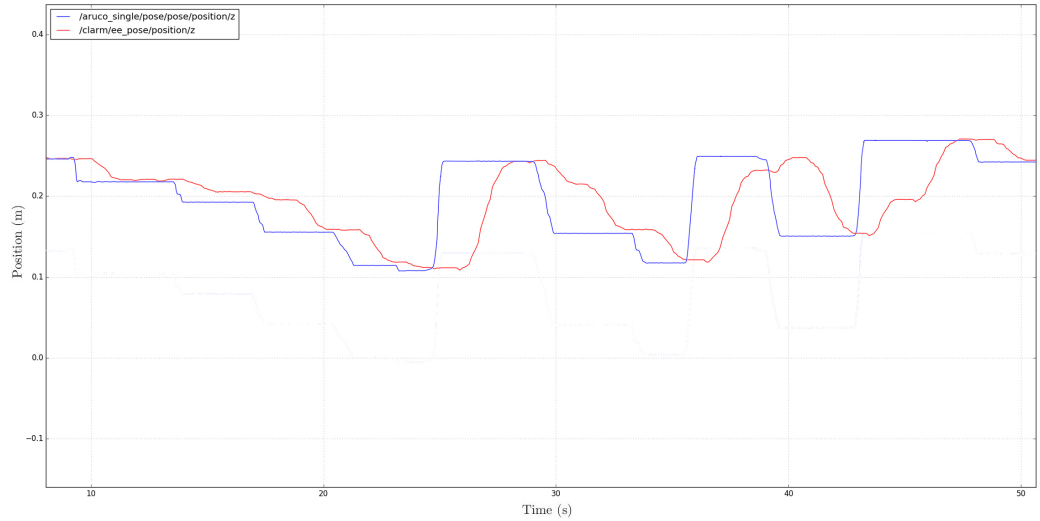

Figure 5.6: z Position Tracking of Marker

controller, to when the end-effector achieved that goal position. The delay differs in different areas of the graphs and this is due to the fact that the controller calculates a trajectory to the desired position while avoiding singular configurations. The execution of a trajectory varies in length as the end-effector is not moving in a straight line to the goal pose but in a optimized path due to the joint space control method used. The stationary tracking of the x axis as well as the tracking of the marker is shown in Figure 5.7.

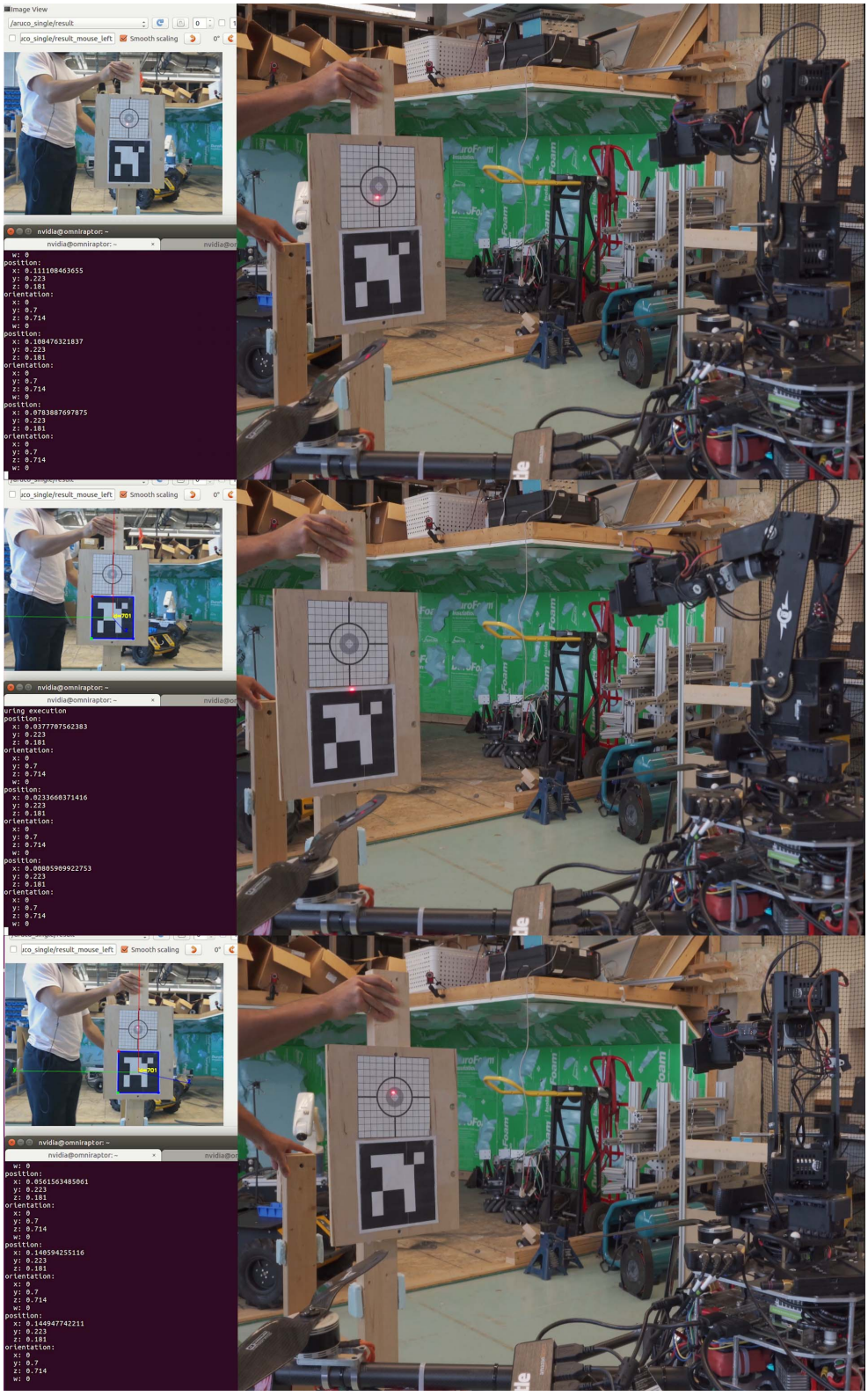

Figure 5.7: Poses of LWMS During x Position Tracking of Marker

5.2.2 Multi-Axis Tracking

The next set of tests were to evaluate the tracking of two axes as well as all three axes combined when a marker was detected. The results of tracking the x and y axes as well as the x, y, and z axes can be seen in Figures 5.8 and 5.9.

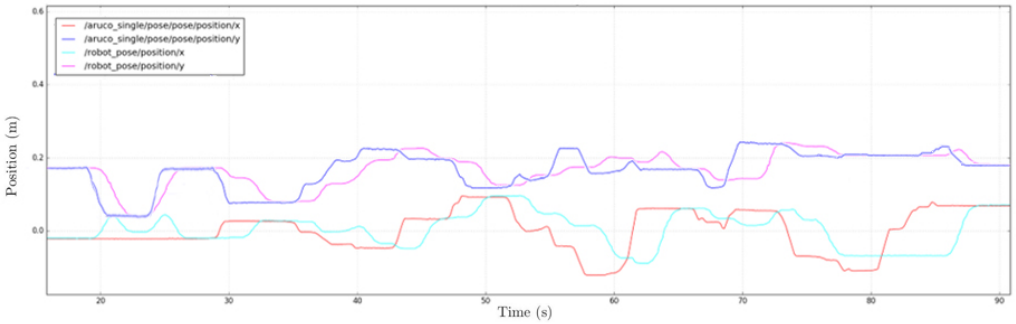

Figure 5.8: x and y Position Tracking of Marker

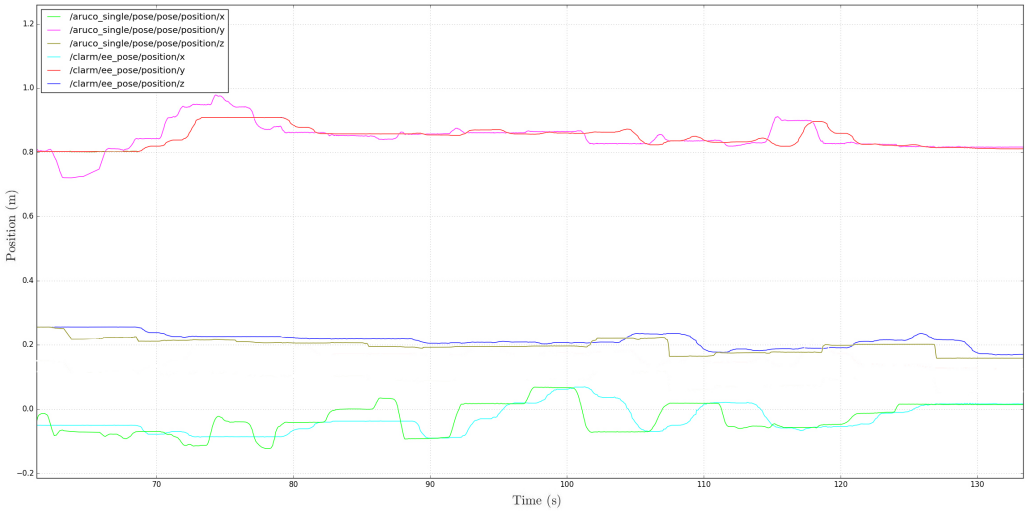

Figure 5.9: x, y, and z Position Tracking of Marker

The results presented in both single and multi-axis tracking produce a similar delay when tracking the marker. In both cases, if a position that was given to the controller was not valid, unreachable, or produced a singularity, the controller would not execute the trajectory and remain in its current position. This resulted in regions of the graphs where the end-effector's position remained constant for a period of time. In

other regions it was observed that the end-effector's position would take a step like pattern to reach the goal. This was due to the method of control. In each of the tests the position of the marker was sampled, the controller executed the planned trajectory to that sample position, and the re-sample for the next trajectory after execution. Finally, as the manipulator was moving point-to-point it was observed that in some cases that the manipulator would take an unconventional path or reconfigure its actuators to reach the marker's position as the controller was using joint-space control. This added a significant delay as the trajectory time was greater than just a straight line path. The results found are due to the method of control which yields point-to-point tracking. This level of tracking error would not be acceptable for the goal of the system to position a sensor in windy conditions. The stationary tracking of the x and y position of the marker is shown in Figure 5.10

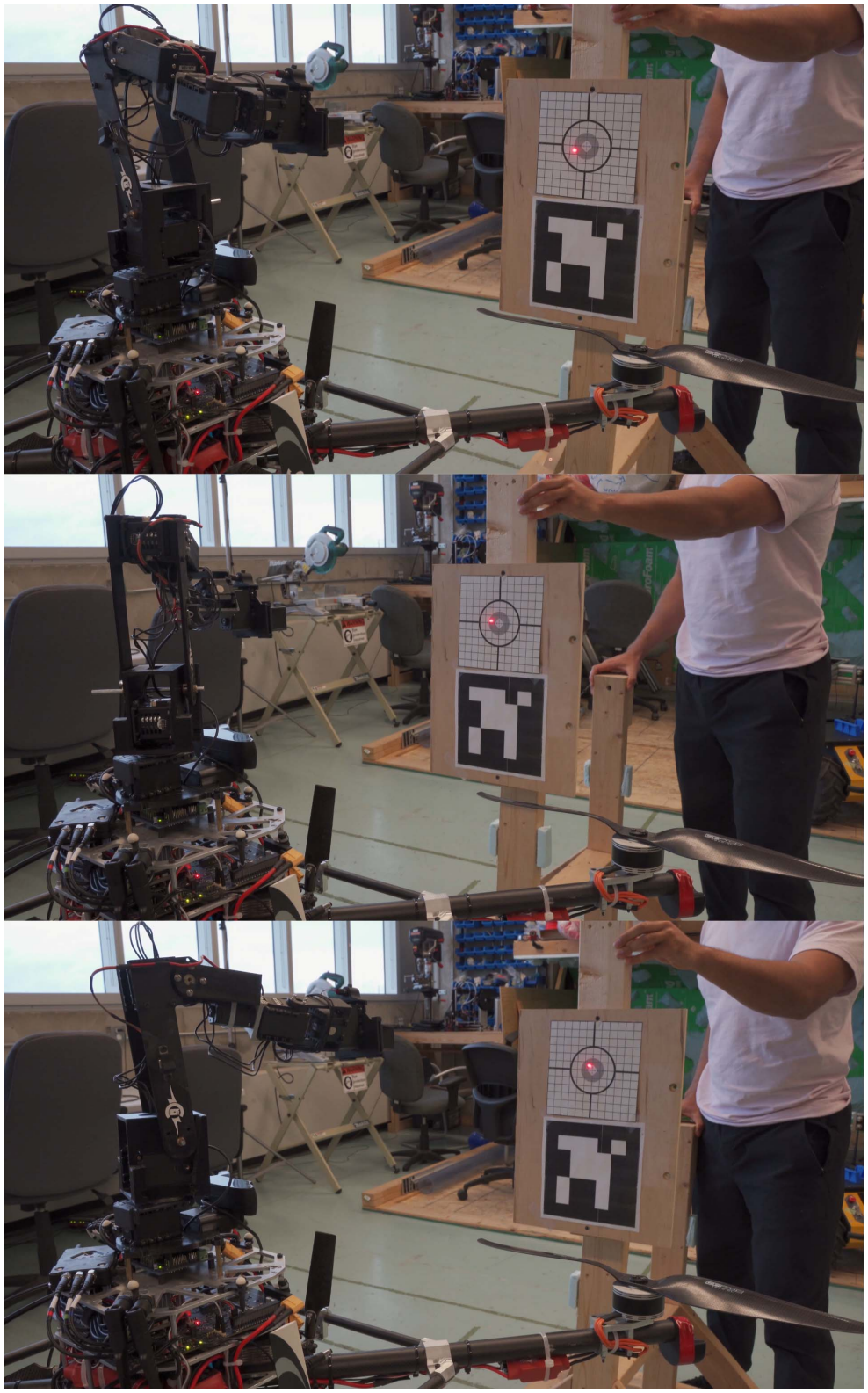

Figure 5.10: Poses of LWMS During x and y Position Tracking of Marker

5.3 Flight Testing

5.3.1 Joint Space Control

To test that the motion of the manipulator would not affect the dynamics of the OmniRaptor a flight test was performed where the manipulator was actuated to the limits of its work space while the OmniRaptor was commanded to hold a position. In this test the manipulator was operated in joint space control where the motion from one point to the next was a smooth continuous motion. The LWMS operating with the OmniRaptor can be seen in Figure 5.11.

In this flight test the OmniRaptor took off in manual control by the operator and then put into autonomous mode. In this mode the OmniRaptor was programmed to hold its current position. The manipulator was then actuated to random extremes in its work space and the data from the flight controller was recorded. The operator switched back to manual control and then landed the OmniRaptor. The x, y, and z local positions of the OmniRaptor can be seen in Figures 5.12, 5.13, and 5.14, respectively. The black dotted lines represent the region of time that the OmniRaptor was in autonomous control with the goal of holding position. In the data displayed in Figures 5.12, 5.13, and 5.14 a maximum deviation was found in the x axis of 3 cm. Minimal deviation was found in both the y and z axes. The deviation of the x axis found was due to the initial calibration of the motion capture system and other factors such as the disturbance generated in the small test area by the thrust produced by the propellers.

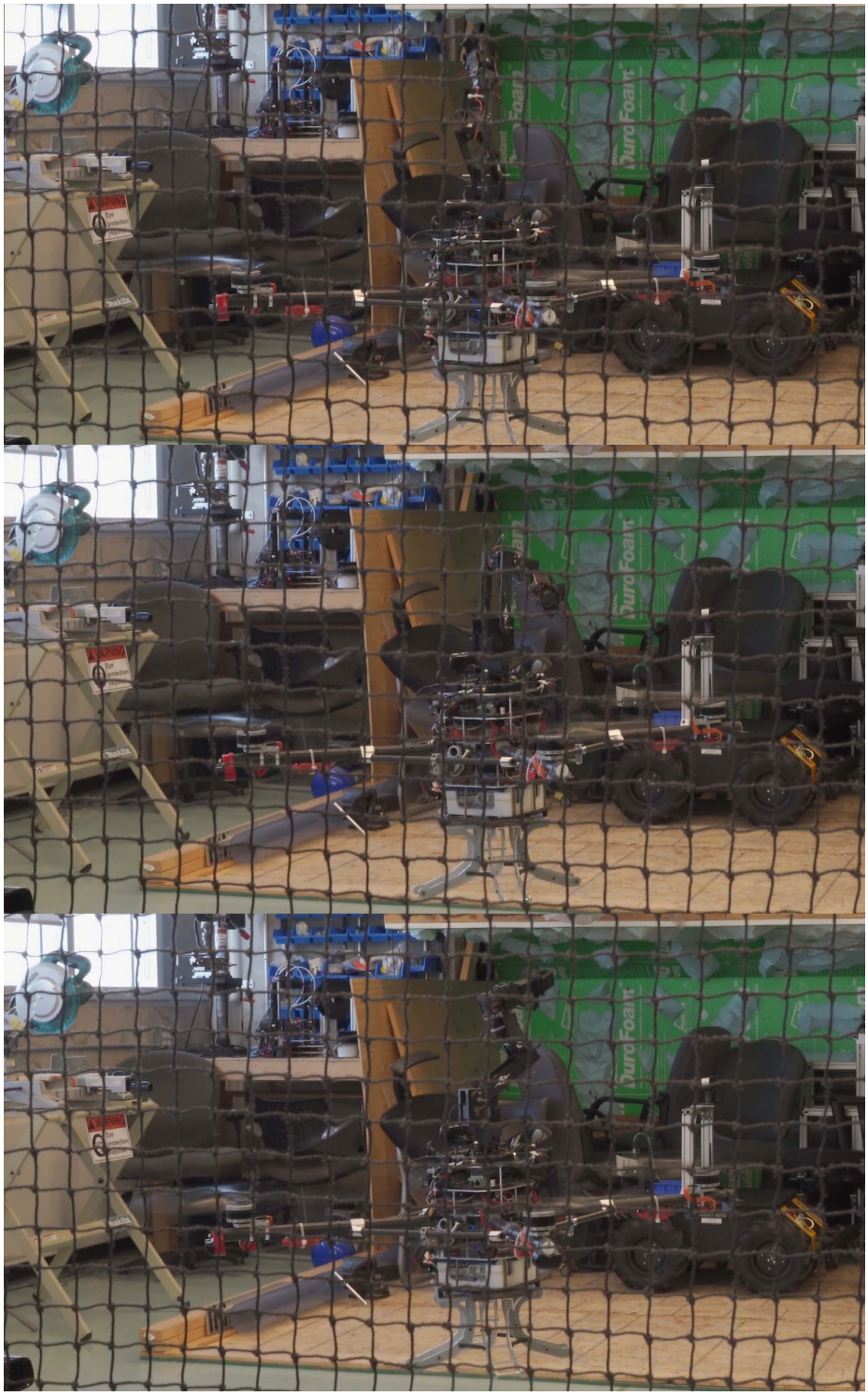

Figure 5.11: LWMS Joint Space Actuation During OmniRaptor Autonomous Operation

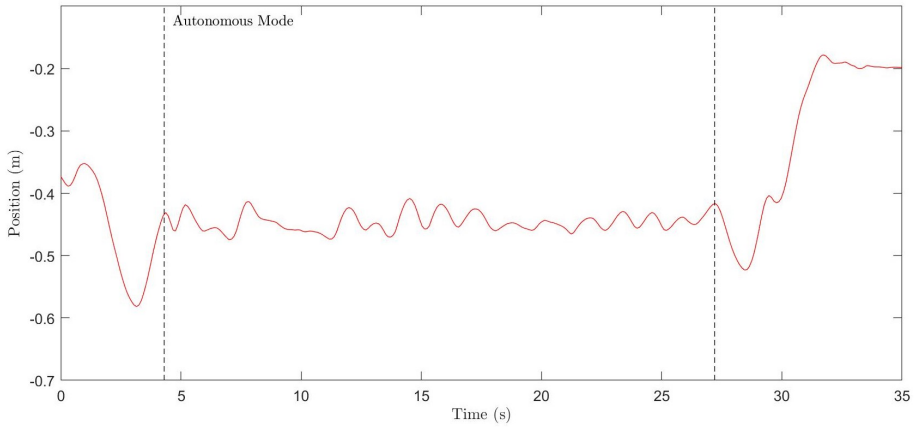

Figure 5.12: Local x Position Of OmniRaptor

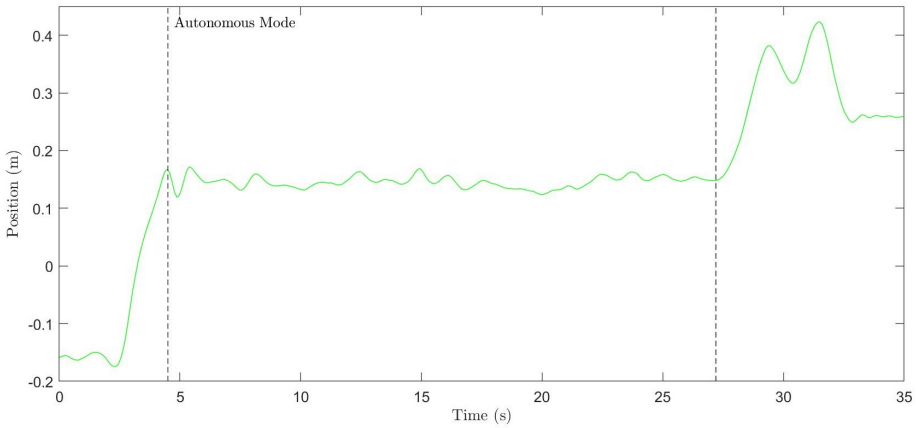

Figure 5.13: Local y Position of OmniRaptor

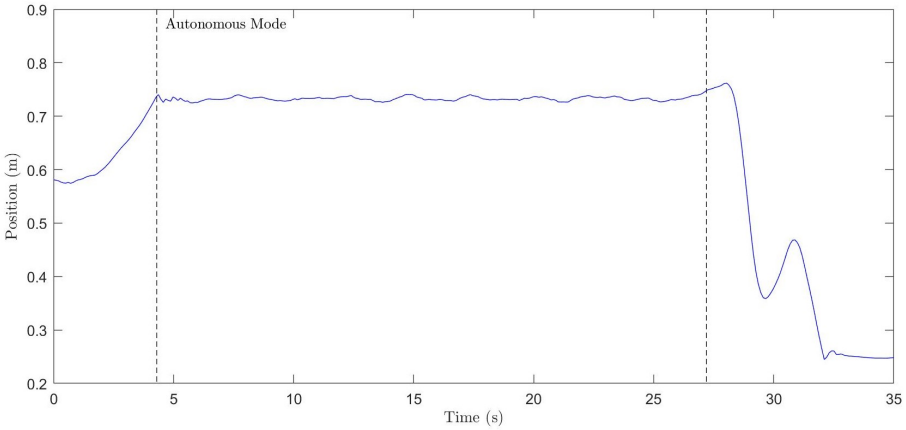

Figure 5.14: Local z Position Of OmniRaptor

5.3.2 In-Flight Marker Tracking

To test the LWMS tracking ability while the OmniRaptor was in operation, the visual servoing of the LWMS was executed while in flight. In this test the OmniRaptor took off in manual mode by an operator and then was commanded to a point where the OmniRaptor was to hold its position and altitude. While in this position the OmniRaptor was disturbed via the control of the operator. This was done through the roll and pitch stick of the remote control. The roll (left or right) command of the stick was mapped to the y position of the OmniRaptor and the pitch (forward or backward) was mapped to the x position of the OmniRaptor. When the control stick was moved to its extremes the OmniRapter would move ± 5 cm. As the control stickres itself upon release the OmniRaptor would return to its original position designated by the autopilot. In this test the operator moved the control stick in both axess individually and then in a random manner to simulate disturbance. The layout of the mapped control stick can be seen in Figure 5.15.

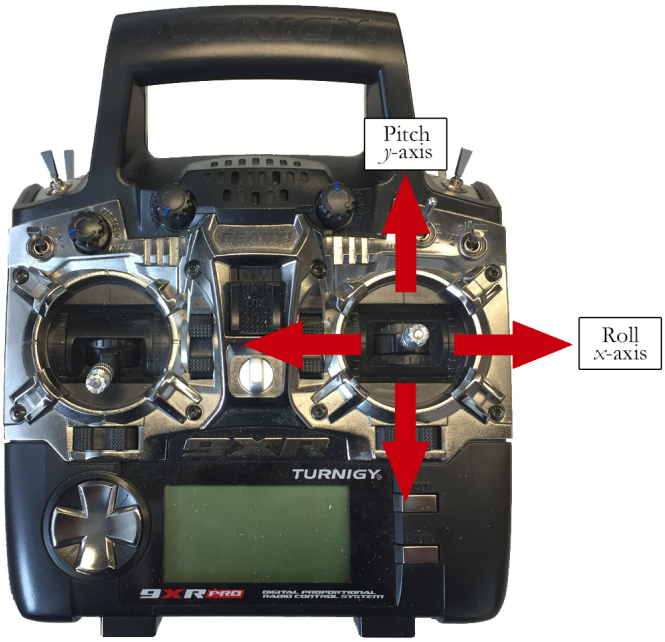

Figure 5.15: Control Stick Layout of Simulated Disturbance Rejection

In this test the x, y, and z positions of the marker value was commanded to the manipulator. The marker was successfully tracked during the flight and while being disturbed by the operator with a maximum delay of 3 seconds. The marker position versus the end-effector's position can be seen in Figure 5.16. A similar delay was found to the data presented in Section 5.2 of this thesis. The local position of the OmniRaptor during the test flight is displayed in Figures 5.17, 5.18, and 5.19. The in-flight marker tracking is also displayed in Figure 5.20.

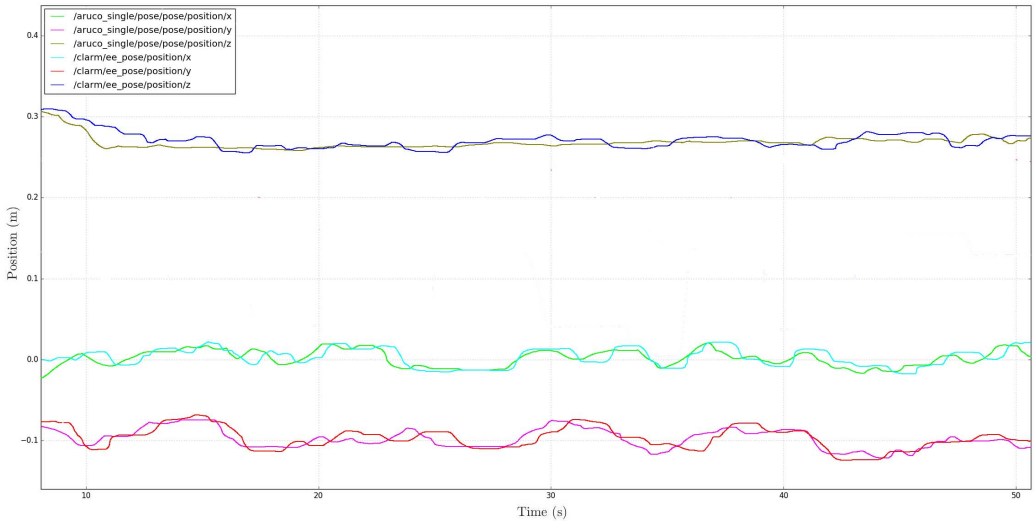

Figure 5.16: x, $\mathbf{y}$, and z Position Tracking of Marker During Flight

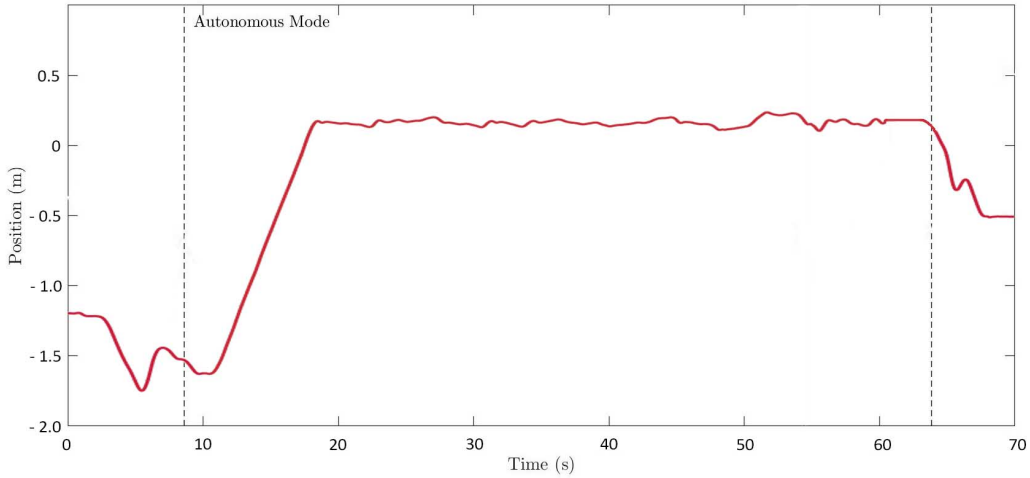

Figure 5.17: Local X Position of the OmniRaptor

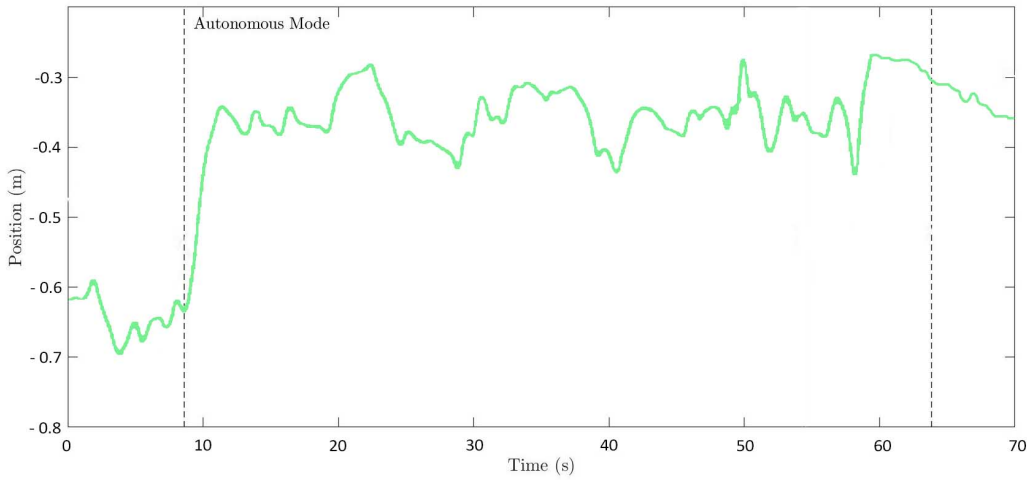

Figure 5.18: Local Y Position of the OmniRaptor

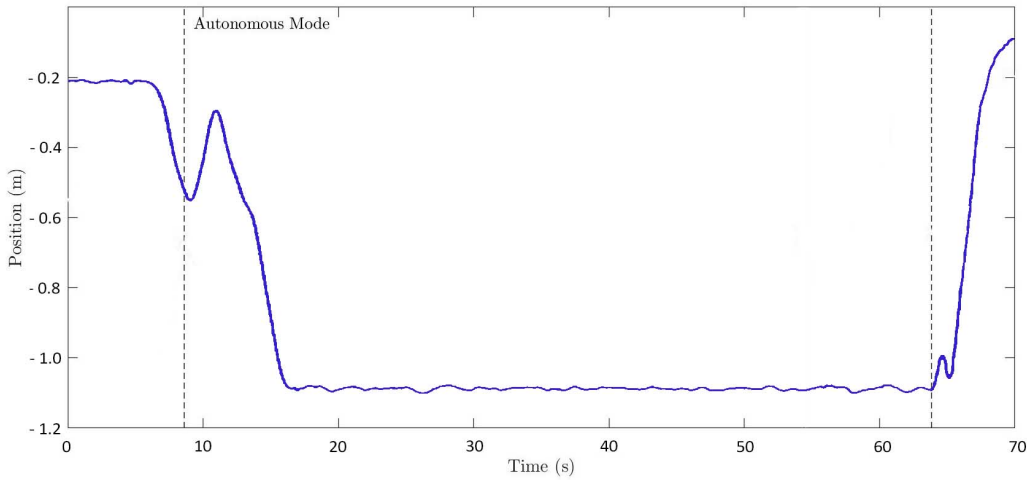

Figure 5.19: Local Z Position of the OmniRaptor

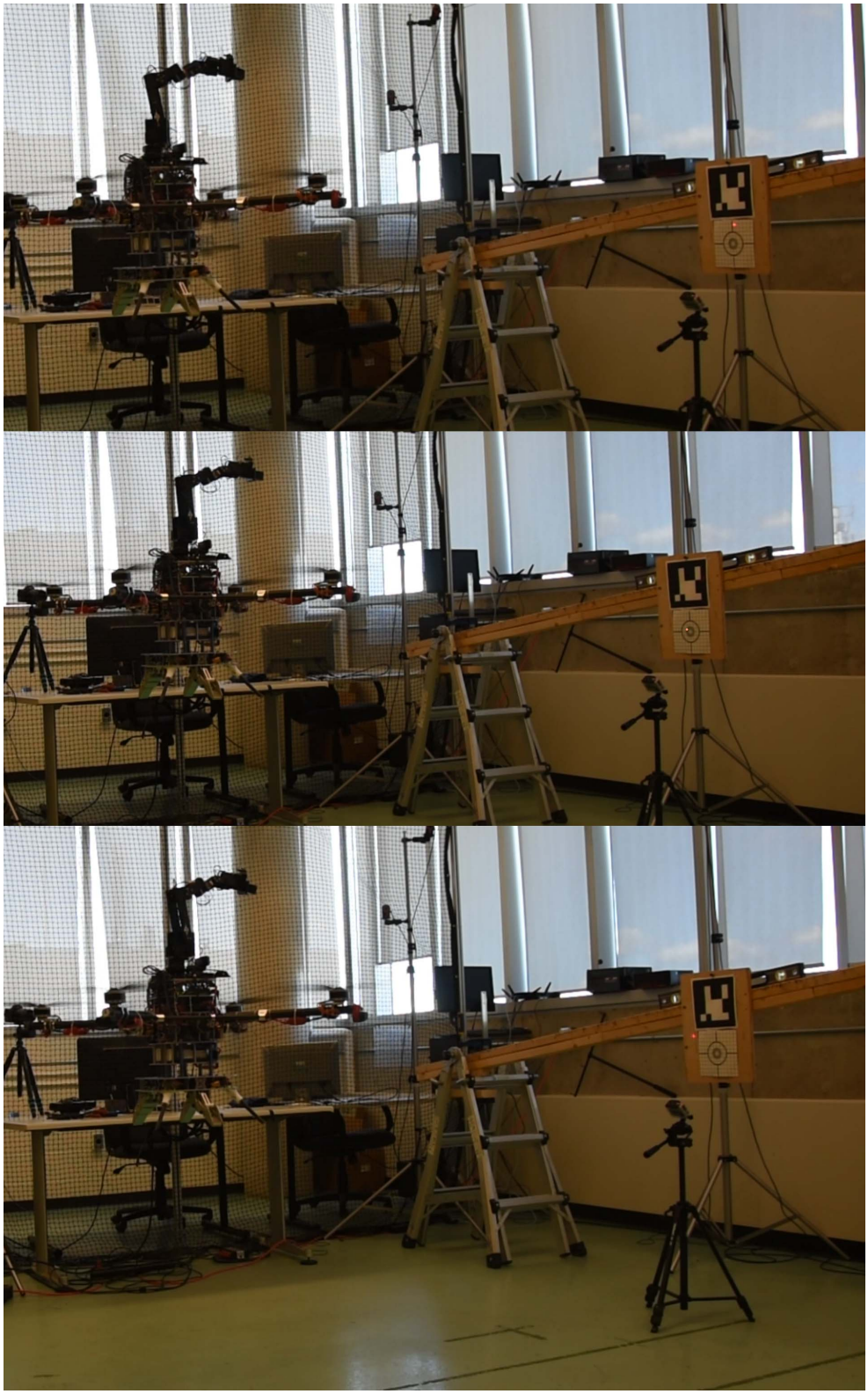

Figure 5.20: In Flight Testing of Marker Tracking

The marker was successfully tracked during the in-flight test where the OmniRaptor was commanded to hold a position and disturbed in the x and y axes by the operator. Although the marker was tracked it was observed that the laser on the target was moving in random behavior as it was seen going on and off the target while being disturbed. The first and main reason for this was due to the point-to-point tracking of the marker. The position changes due to the generated disturbance were too erratic for the controller to keep the sensor on the tracked target. In some sections of the graph shown in Figure 5.16 the end-effector's position would trail behind the position of the marker which was similar to the results found in Section 5.2, but the results found in this test showed a larger delay. In addition, the marker's position relative to the OmniRaptor in the y axis was 2.5 m. This was a greater distance from the testing done in the stationary configuration which ranged from 0.5 m to 1.0 m. Although the tracking is designed to compensate for the y axis distance, the further the marker is away from the LWMS a greater error of tracking is present. Finally, The OmniRaptor was disturbed ± 5 cm in the x and y planes during the flight test, but in the stationary testing the marker was displaced in a range of 0.1 m to 1.0 m at a time. Comparing the results found in the stationary and in-flight tracking of the marker, the LWMS is able to track more accurately if there are larger movements of the relative position of the marker to the OmniRator. This is due to the method of control used.

Chapter 6

Conclusions and Recommendations for Future Work

6.1 Conclusions

The completion of this thesis presents the design, implementation, and testing of a proof-of-concept prototype manipulator system for the OmniRaptor. The work done in this thesis represents a single part of the progression of the OmniRaptor project to its goal of developing a full-autonomous omni-directional multi-rotor system capable of perching, object tracking, feature recognition, and object manipulation.

The proof-of-concept prototype of the LWMS and its control strategy demonstrated the detection of predefined objects, its ability to position the end-effector to track the detected object, and integration into the OmniRaptor's control structure. In addition, a ROS based visualization of the LWMS coupled with the OmniRaptor was achieved. This work gives the conceptual design and control strategies that can be used towards the design and development for a more refined and possibly larger scale

manipulator system for the OmniRaptor.

During testing of the LWMS the control structure provided the system with a basic point-to-point tracking of a fiducial marker. Although the marker was tracked there was some delay from the end-effector reaching its targeted goal due to the joint-space control method used. To resolve this issue, a velocity based control structure was introduced to the manipulator which allowed for velocity commands in Cartesian-space. With an implementation of a secondary feedback controller comparing the position of the tracked object to the end-effector's current position would allow the manipulator to continuously track the detected object rather than in a point-to-point basis. As this was a proof-of-concept, the control structure of the secondary controller was designed but not implemented.

Although this thesis concentrated on the concept of sensor placement and disturbance rejection, the application potential of a manipulator system coupled with an omni-directional multi-rotor are limitless. This system is originally targeted for inspection and light maintenance of structures such as electrical towers but the system could potentially find application in any industry that requires sensor placement, force exertion, or object manipulation in hard to reach or hazardous areas.

6.2 Recomndations for Future Work

As indicated, the prototype LWMS and control structure developed in this thesis posses some limitations and there are multiple areas which further development under taken.

Suggestions for future development work related to the prototype and control system developed in this thesis include:

- Replacement of the PLA parts used in the design to a stronger light-weight material such as aluminum.
- Replacement of the shoulder pitch assembly to a stronger actuator to remove the use of a gear box to reduce weight.
- Design of a second generation LWMS prototype that uses stronger DC servo motors.
- Implementation of a stereo camera that would allow the recognition of features to be tracked.
- Implementation of an object tracking algorithm that detects a set of multiple objects rather than fiducial markers.
- Implementation of a feedback controller for the application of real-time velocity based tracking.
- Implementation of a controller that allows the user to exert a known force on an object or surface.

Future development work extending beyond the scope of the current OmniRaptor prototype includes:

- Redesign of the OmniRaptor's layout for the ability to mount the LWMS in different configurations.
- Addition to the controller that allows the user to designate the command frame of the system, i.e. the end-effector frame.

A proof-of-concept LWMS was presented in this thesis that demonstrated the basic application of visual servoing and combining an omni-directional platform with a

manipulator system. This thesis is intended to pave the way for further development of a manipulator system for the OmniRaptor platform.

References

[1] T. Baltovski, S. Nokleby, and R. Pop-Iliev, "Towards performing remote manipulation using an autonomous aerial vehicle," in *CCToMM Symposium on Mechanisms, Machines, and Mechatronics*, 2015.

[2] MoveIt, "Moveit concepts move_group," 2019. [Online]. Available: https://moveit.ros.org/documentation/concepts/

[3] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. Rodríguez Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. Fernández Perdomo, "ros_control: A generic and simple control framework for ros," *The Journal of Open Source Software*, 2017. [Online]. Available: http://www.theoj.org/joss-papers/joss.00456/10.21105.joss.00456.pdf

[4] T. Baltovski, S. Nokleby, and R. Pop-Iloev, "Design and development of an swinging arm mechanism for an aerial manipulator system," in *International Conference on Mechanical Engineering and Mechatronics*, 2013.

[5] C. Baird, "Design and development of an omni-directional unmanned aerial vehicle and a universal gripper mechanism for autonomous perching," Master's thesis, Ontario Tech University, 2020.

[6] A. Ollero, G. Heredia, A. Franchi, G. Antonelli, K. Kondak, A. Sanfeliu, A. Viguria, J. R. M. de Dios, F. Pierri, J. Cortés, A. Santamaria-Navarro, M. A. Trujillo, R. Balachandran, J. Andrade-Cetto, and A. Rodriguez, "The aeroarms project," 2018.

[7] M. Ryll, D. Bicego, and A. Franchi, "A truly redundant aerial manipulator exploiting a multi-directional thrust base," in *IFAC (International Federation of Automatic Control)*, 2018, pp. 138–143.

[8] I. Palunko and R. Fierro, "Autonomous lift of a cable-suspended load by an unmanned aerial robot," in *IEEE Conference on Control Applications*, 2014, pp. 802–807.

[9] P. Cruz and R. Fierro, "Cable-suspended load lifting by a quadrotor uav: Hybrid model, trajectory generation, and control," in *Autonomous Robot*, 2017, pp. 1629–1643.

[10] Y. S. Sarkisov, M. Jun Kim, D. Bicego, D. Tsetserukou, C. Ott, A. Franchi, and K. Kondak, "Development of sam: cable-suspended aerial manipulator," in *IEEE Int. Conf. on Robotics and Automation*, 2019.

[11] M. Tognon, A. Testa, E. Rossi, and A. Franchi, "Takeoff and landing on slopes via inclined hovering with a tethered aerial robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 1702–1707.

[12] S. Tognon, M. Dash, and A. Franchi, "Observer-based control of position and tension for an aerial robot tethered to a moving platform," in *IEEE Robotics and Autonomous Letters*, 2016, p. 732–737.

[13] M. Tognon and A. Franchi, "Dynamics, control, and estimation for aerial robots tethered by cables and bars," in *IEEE Transactions on Robotics*, vol. 33, 2017, p. 834–845.

[14] I. Palunko, P. Cruz, and R. Fierro, *Load transportation using aerial Robots*. Springer International Publishing, 2018.

[15] M. Bernard, K. Kondak, I. Maza, and A. Ollero, "Autonomous transportation and deployment with aerial robots for search and rescue mission," in *Journal of Field Robotics*, vol. 28, 2011, pp. 914–931.

[16] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and trasportation with aerial robots," in *Autonomous robots. Special issue: Robotics: Science and Systems, R. Tedrake and Y. Matsuoka, Eds.*, vol. 30, 2011, pp. 73–86.

[17] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," in *International Journal of Robotics Research*, vol. 30, 2011, pp. 324–334.

[18] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robot," in *Robotics: Science and Systems*, 2013.

[19] G. yu Zhang, Y. He, B. Dai, F. Gu, L. Yang, J. Han, and G. Liu, "Aerial grasping of an object in the strong wind: Robust control of an aerial manipulator," *Applied Sciences*, vol. 9, p. 2230, 2019.

[20] F. Ruggiero, M. A. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Peréz, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor uavs equipped with a servo robot arm," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4014–4020.

[21] F. Ruggierro, "Impedence control of vtol uavs with a momentum-based forces estimator," 2014.

[22] ——, "Image-based visual-impedaence control," 2018.

[23] K. Kondak, M. Schwarzbach, kai Kriegger, and M. Laiacker, "First analysis and experiements in aerial manipulation using fully actuated redundant robot arm," 2018.

[24] C. Chen, F. Quan, L. Fang, and S. Zhang, "Aerial grasping with a lightweight manipulator based on multi-objective optimization and visual compensation," *Sensors*, 2019.

[25] G. Zhang, Y. He, B. Dai, F. Gu, L. Yang, J. Han, and G. Liu, "Aerial grasping of an object in the strong wind: Robust control of an aerial manipulator," *Applied Sciences*, 2019.

[26] S. C. H. J. K. Suseong Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," 2013.

[27] B. Yang, Y. He, J. Han, and G. Liu, "Rotor-flying manipulator: Modeling, analysis and control," 2014.

[28] A. O. Guillermo Heredia, A.E. Jimenez-Cano, "Control of a multirotor outdoor aerial manipulator," 2014.

[29] H. V. Somasundar Kannan, Miguel A. Olivares-Mendez, "Modeling and control of aerial manipulation cheicle with visual sensor," 2016.

[30] V. K. Justin R. Thomas, "Grasping, perching, and visual servoing for micro aerial vehicles," Ph.D. dissertation, 2017.

[31] R. V. Guangying, "Hexarotor uav platform enaling dextrous aerial mobile manipulation," Ph.D. dissertation, 2017.

[32] V. Ghadiok, "Autonomous aerial manipulation using a quadrotor," Master's thesis, 2011.

[33] X. Meng, Y. He, and J. Han, "Survey on aerial manipulator: System, modeling, and control," *Robotica*, p. 1–30, 2019.

[34] J. J. Craig, *Introduction To Robotics*. Pearson, 2005.

[35] W. Khalil, "Dynamic modeling of robots using recursive newton-euler techniques," *ICINCO*, 2010.

[36] F. von Frankenberg, "Disturbance rejection in multi-rotor unmanned aerial vehicles using a novel rotor geometry," Master's thesis, Univesity of Ontario Institute of Technology, 2017.

[37] F. von Frankenberg and S. Nokleby, "Inclined landing testing of an omnidirectional unmanned aerial vehicle," *Transactions of the Canadian Society for Mechanical Engineering*, pp. 61–70, 2018.

[38] P. B. M. ltd., "A394 tower bolts," 2020.

[39] ROBOTIS, "Dynamixel," 2019. [Online]. Available: http://www.robotis.us/dynamixel/

[40] ROS, "About ros," 2019. [Online]. Available: http://www.ros.org/about-ros/

[41] O. RAVE, "Ikfast," 2019. [Online]. Available: http://openrave.org/docs/0.8.2/openravepy/ikfast/