Development of a Mobile Modular Robotic System, R2TM3, for Enhanced Mobility in Unstructured Environments

by

Sean Phillips

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Limited mobility of mobile ground robots in highly unstructured environments is a problem that inhibits the use of such robots in applications with irregular terrain. Furthermore, applications with hazardous environments are good candidates for the use of robotics to reduce the risk of harm to people. Urban search and rescue (USAR) is an application where the environment is irregular, highly unstructured and hazardous to rescuers and survivors. Consequently, it is of interest to effectively use ground robots in applications such as USAR, by employing mobility enhancement techniques, which stem from the robot's mechanical design. In this case, a robot may go over an obstacle rather than around it.

In this thesis the Reconfigurable Robot Team of Mobile Modules with Manipulators (R2TM3) is proposed as a solution to limited mobility in unstructured terrains, specifically aimed at USAR. In this work the conceptualization, mechatronic development, controls, implementation and testing of the system are given.

The R2TM3 employs a mobile modular system in which each module is highly functional: self mobile and capable of manipulation with a five degree of freedom (5-DOF) serial manipulator. The manipulator configuration, the docking system and cooperative strategy between the manipulators and track drives enable a system that can perform severe obstacle climbing and also remain highly manoeuvrable. By utilizing modularity, the system may emulate that of a larger robot when the modules are docking to climb obstacles, but may also get into smaller confined spaces by using single robot modules. The use of the 5-DOF manipulator as the docking device allows for module docking that can cope with severe misalignments and offsets – a critical first step in cooperative obstacle management in rough terrain.

The system's concept rationale is outlined, which has been formulated based on a literature review of mobility enhanced systems. Based on the concept, the realization of a low cost prototype is described in detail. Single robot and cooperative robot control methods are given and implemented. Finally, a variety of experiments are conducted with the concept prototype which shows that the intended performance of the concept has been met: mobility enhancement and manoeuvrability.

Acknowledgements

First and foremost, I would like to thank my wife for her patience and support through my oddly timed academic pursuits. I have always felt encouraged and supported through the entire process of my education. I would also like to thank all of my other family members and friends who have supported my decisions and expressed interest in my endeavors.

I am very grateful to my supervisors, Dr. Jan Paul Huissoon and Dr. Hamidreza Karbasi for granting me this tremendous opportunity and for providing guidance and encouragement. Additionally, I would like to thank them for giving me a large amount of liberty in the project choice, research direction and decision making. The entire process has been extremely rewarding because I was able to propose, define my own project, and see it through.

I would also like to thank Dr. Dana Kulić and Dr. William W. Melek for agreeing to read my thesis.

Thank you to everyone who has assisted with or supported this project, whether it be the manufacturing of parts, providing feedback or just letting me voice my problems. Without this assistance, the project would not have worked out as it has.

The support and encouragement of others has made this work possible. Thank you to everyone who made it possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The motivation for this work stems from an interest in applying robotics to dangerous and difficult tasks to reduce the risk of harm to people. With my familiarity in working with off-road machinery, the idea of using robotics in rough off-road terrains was also of interest. Once a robot leaves a well structured setting, such as a factory, and becomes mobile, its environment can become highly unstructured. Upon considering the use of mobile ground robots in such unstructured environments the problems and limitations became apparent; wheels or tracks provide a simple, robust and efficient means of locomotion, but may suffer from limited mobility in unstructured terrains [1], [2]. Therefore, the application interest and the research problem of limited mobility motivated this work.

1.2 Intended Applications

Given the motivation for this work, urban search and rescue (USAR) is an obvious and well suited specific application. After a disaster has occurred, a search for survivors needs to be conducted promptly. Rescuers are at high risk of injury due to many potential hazards such as air contaminants, heat and fire, falling debris, structural collapse and possibly nuclear radiation, among others [3]. With little initial information on the severity and nature of these risks, the rescuers are vulnerable to harm

upon entering the disaster site for assessment. Often first responders suffer from injuries or other related health problems from their search and rescue efforts [4]. In the case of a collapse the rescue effort may be halted, due to structural uncertainty, which prevents the rescue of survivors in the viable time frame [5]. When considering past accounts of disasters, it becomes apparent, and has already been recognized, that USAR is a good application for robotics for the purpose of preserving human life. Mobile ground robots may be used via teleoperation to enter a disaster site with unknown risks to provide sensory feedback (visual etc.) to the remote operator. This helps to remotely assess the risks and identify the location of survivors, from the up close view of the robot [6]. In some cases robots could deliver small, yet life saving packages, or provide a means of communication with survivors [1]. Another possibility is to have more autonomy built into robotic systems which may enable greater mission efficiency or ease the operational burden on the remote operator [7], [8]. Aerial robots are also used in USAR and may be used in combination with ground robots as part of a team, providing information from an overall view of the disaster site [9].

The first documented use of robotics in USAR was in the rescue effort after the terrible incident of the world trade tower attacks [1]. Many problems were identified in using mobile ground robots in USAR in this real world situation, such as poor robustness, poor situation awareness and limited mobility [1]. Most robots would simply get stuck on debris, smaller ones would disappear into a crevice and larger ones would be prevented from getting into smaller confined spaces [1]. Since that time it has been recognized that using robots in unstructured terrains, such as that found in most USAR, requires unique approaches originating from the mechanical design of the robot. Many robots have been developed to address this problem and several will be discussed in this chapter.

It is also worth mentioning that limited mobility in unstructured terrains is not exclusive to USAR. The mobility enhancement of mobile robot platforms can be applied to a variety of other applications. Any environment in which the nature of its obstacles cannot be known exactly before entering, or the obstacles are known but severe, is a good candidate for mobility enhancing techniques. It might be necessary to go over an obstacle rather than around it. Examples of additional applications to USAR are planetary exploration, inspection and hazardous site cleanup [10].

1.3 Literature Review

In this section, several different areas of robotics will be reviewed and robotic systems will be described for the purpose examining approaches used to enhance mobility. Wheeled or tracked robots provide a simple, robust and efficient means of locomotion, but tracked mobile robots are preferred in rough terrains because their greater contact area with the ground greatly enhances manoeuvrability and traction [2]. Walking robots, an alternative to mobility enhanced tracked robots, can work well in rough terrain, but they have their own set of research problems. They can suffer from a high level of complexity, can be difficult to stabilize and are inefficient for travelling across large distances in relatively smooth terrain [2]. Walking robots and other alternatives to tracked mobile robots are not reviewed in this work and are left as an alternative research area.

The R2TM3 concept developed in this work stems from several areas of robotics reviewed in the following sections. The review is done within the context of the stated research problem and intended application.

1.3.1 Modular Reconfigurable Robotics

Modular reconfigurable robotics is a field originally inspired from cellular biology [11]; simple robot modules can be assembled in a variety of ways into a larger functional system. With this biological inspiration, researchers have been trying to realize the potential benefits of the approach: versatility, robustness and low cost [11]. Versatility comes from the fact that with the same number of modules, the robot may be configured to perform different tasks such as locomotion, manipulation or simply providing a structure. Robustness is a result of the modules potentially being identical; giving the system redundancy, where a failed module can be discarded or replaced with any other module. Having a high production volume of just one module type can possibly lead to cost savings. In [11] the authors categorize modular reconfigurable robots (MRRs) into lattice, chain, mobile and stochastic (hybrid systems also exist). An example of a hybrid chain and lattice MRR can be seen in Figure 1-1.

Figure 1-1: Atron, a Hybrid and Lattice MRR [11]

In [12] the primary reasons for studying the field are nicely stated as well as the interplay between them, “A strong focus on the basic research of biological principles will uncover new possibilities and open new doors for applications. On the other hand, a focus on applications will force real life constraints onto the otherwise abstract theory and much experience can be transferred to theory by working with real world problems.” [12]. The key problems to be overcome within the field are identified as achieving reliable self-reconfiguration, limited versatile functionality, higher module cost as well as others [12]. The author also discusses the idea of homogeneous systems and heterogeneous systems [12]. Some MRR systems consist of several different modules types that perform a specific task such as actuation [13], making the system heterogeneous.

In [13] the authors discuss the use of heterogeneous modules to reduce the module cost and complexity, where only a specific module will contain a motor, which alleviates the need for redundant cost. Furthermore, they present a modular robotic scheme in which manual reconfiguration is required, called Thor – “a robotic building kit” [13].

Figure 1-2: Thor, a Robotic Building Kit [13]

This approach is the result of the researchers recognizing that directing the modular design to a more specific application, may yield higher functionality, yet still allow for adequate versatility, "The Thor modular robot is partly inspired by the experience from our participation in the ICRA 2008 Planetary Robotic Contingency challenge." [13]. This work signifies a deviation from the original biological inspiration of modular robotics, towards heterogeneous modules, which enables a higher level of functionality in practical applications.

Another modular system, LocoKit, was developed with a layered heterogeneity approach [14]. The authors of this work propose a system of three layers of heterogeneity; mechanics, actuation and electronics [14]. They propose an increase in heterogeneity, a narrower application area and a reduction in module complexity to overcome the primary challenges with modular robotics, such as limited functionality [14]. Specifically, the researchers directed the development toward legged locomotion. Although it is still a common research goal to scale modules down to the micron scale and have a system consisting of millions of modules [12], it is obvious that with today's technology a different approach is needed to realize practical utility, such as heterogeneity.

Heterogeneity is not the only approach to achieve greater MRR functionality. Another approach to enhance the MRR concept is to have self mobile modules [15]. The authors in [15] propose that having modules capable of meaningful tasks themselves can help to overcome the challenge of self-reconfiguration. Instead of heterogeneity, this approach increases the individual functionality of each homogeneous module, leading to an overall more functional system that is still versatile. Although module complexity (and therefore cost) is higher than the heterogeneity approach, the greatly increased functionality justifies this complexity. This could also be considered a cooperation

approach of mobile robotics. From this perspective each module is viewed as a somewhat independent robot that can achieve enhanced functionality through cooperation.

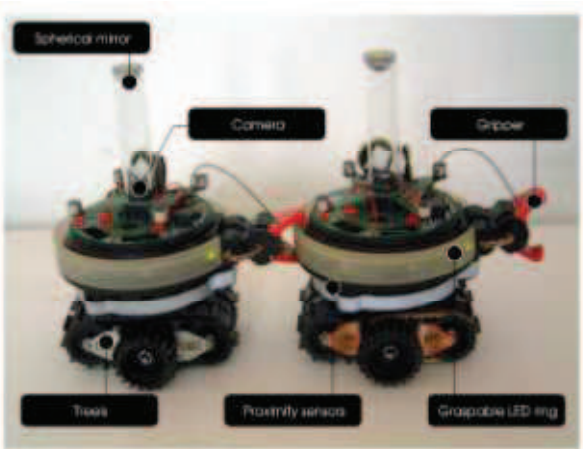

Figure 1-3: Swarm-bot, a Mobile Modular System [15]

This review of modular reconfigurable robotics has led me to the conclusion that if this technology is to be presently utilized in practical applications a deviation from the original scientific inspiration needs to occur. In particular, the idea of mobile modules provides a basis for the concept developed in this thesis.

1.3.2 Mobile Robotics

Mobile robotics is a large and diverse field, and areas of this field that pertain to this thesis will be focused on. A large portion of mobile robotics research is focused on robot autonomy and fully autonomous systems have been successfully developed. Fully autonomous robots are an exceptional technological achievement with exciting applications. However, full autonomy may not be appropriate for practical implementation in some applications with present day technology. In the case of USAR, most ground systems in use are teleoperated [1], [6], [10], [16], where the robot consists of a wheeled or tracked robot base, teleoperated by a remote operator. Reliable full autonomy is difficult to achieve in applications like USAR since the environment is highly unstructured and may require the robot(s) to go over obstacles rather than avoid them [10], [17], [18], [19], [20]. Evaluating the situation and the difficult decision making, in this application, is usually left to the remote operator. In addition, the USAR mission is high risk and mistakes can be devastating not only

to the search personnel (or robot) but to the survivors themselves. Therefore, with present technology considered and the risks involved, teleoperation is the norm in applications like USAR.

This is not to say that some autonomy in this type of application does not have any benefit. In [7] the authors present work done in simulation showing that an operator controlling a group of robots with autonomous path planning was more successful in locating survivors than an operator controlling the same number of robots with no autonomy. In [8] the authors present work in which a fully autonomous system is tested to complete specific tasks in USAR such as navigation, stair climbing and manipulation. Ultimately, semi-autonomous systems are an ideal approach to the application of robotics to USAR, where the remote operator has control over the robots via teleoperation, but they possess enough autonomy to either ease the operational burden on the operator or to improve the mission efficiency [2], [7]. For example, the operator could give overall motion commands, but the low level joint control of the robot or robot modules would be solved automatically [2], [19], [21].

The field of mobile robotics as applied to USAR is advancing and there are organizations dedicated to furthering this research. The RoboCup Rescue League is an international organization that aims to advance the field by holding competitions for virtual systems and real robotic systems, as well to standardize the assessment of rescue robots [22]. The Center for Robot-Assisted Search and Rescue (CRASAR) is an organization that serves to utilize robotic developments to assist in actual search and rescue missions [23]. The potential technological advances from such organizations are an exciting prospect and the application of mobile robots in USAR is of growing interest.

Several mobile robots that have been developed for, or used in, USAR or related fields will now be described. A very common tracked mobile robot used in USAR and similar applications is the Packbot [24], which is a standard platform used by many researchers. It has a robust construction and uses its front flippers to help it climb stairs.

Figure 1-4: Packbot [24] Used as Base Platform for X1 Robot [8]

In [6] the authors proposed a system with a hyper-redundant robot (HRR) mounted on a mobile base, with a camera mounted on the end of the HRR. This concept made several advancements such as enhanced sensing ability (situation awareness) via the HRR arm being manoeuvred into many different configurations and the added robustness of having redundant links in the arm.

Figure 1-5: A Hyper Redundant Robot Mounted on a Mobile Platform [6]

Although this work made a significant contribution to USAR robotics, the system lacks any kind of mobility enhancement method.

Another robot developed for USAR is the CAESAR robot [25]. This robot has a symmetrical design, which enables it to continue working when flipped over. This robot has been designed for

field use and has had many improvements made to it to enhance its overall robustness. It uses actuated elevating arms on the front and rear to help enhance mobility.

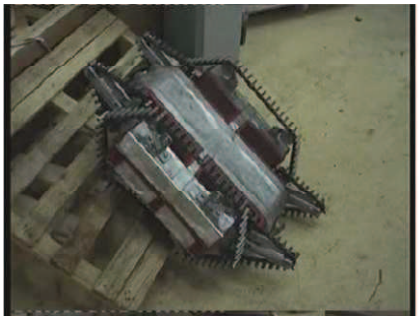

Figure 1-6: CAESAR a Mobile USAR Robot [25]

In the previous description of mobile robots used in USAR the type of mobility enhancement method used, if any, was indicated. This topic was not discussed in detail, since the next section is entirely devoted to categorizing different robotic systems by mobility enhancement methods, whether the robot system is specifically intended for USAR or not. More novel mobility enhancement techniques will be discussed there.

1.3.3 Mobility Enhancement Methods

Many robots used in unstructured environments use some method of enhancing their mobility. Mobility enhancement can be for obstacle climbing, fitting into tight spaces or enhancing traction and stability in rough terrain. There are several methods used to enhance mobility of interest and they are categorized as follows.

Traditional Flippers or Rotating Tracks:

Many robots utilize flipper or elevating arms to help them climb up obstacles such as stairs and ramps, as seen on the Packbot [24] and CAESAR [25] in Figure 1-4 and Figure 1-6 respectively. These simple and robust mechanisms work well for many obstacles, but are limited in the size of an obstacle they may enable the robot to climb. Additionally, employing this technique results in a fixed robot size, which may inhibit the robot from getting into tight spaces.

Reconfiguration and Hybrid Methods:

A number of other robotic systems have been developed in response to the limited mobility problem, that employ more advanced mobility enhancement techniques such as reconfiguration [18], [26], [27] or hybrid techniques [10], [19].

A novel robotic system that has been developed for mobility enhancement in USAR using self-reconfiguration, is the AMOEBA-I [18].

Figure 1-7: AMOEBA-I, a Shape Shifting Robot [18]

Each single tracked module is connected by an actuated link that has a pitch and yaw joint [18]. The modules can then be repositioned relative to one another to assume different configurations that have different advantages, such as stair climbing versus side stability. An interesting technique in this work is the use of cooperative shape shifting [18]. The authors develop a scheme whereby the torque requirements for a given joint can be reduced by simultaneously moving another joint that repositions the centre of gravity of the module being actuated. This type of cooperation can lead to smaller actuators and lighter designs. AMOEBA-I is waterproof and hollow, allowing it to float and move in water [18]. A major shortcoming of the AMOEBA-I system is that it cannot perform manipulation.

In [26] a climbing crawler robot, DIR-2, is shown that can achieve high ledge climbing. It uses a triangular shaped crawler unit with a 2-DOF linkage that can fold into the crawler. Although this crawler can climb heights up to 1.5 times its crawler length, it also has no manipulation capability and could be stranded if flipped on its side, since its linkage cannot be actuated in a direction that would push it upright.

Figure 1-8: DIR-2, a Reconfigurable Crawler [26]

In [27] a system with an onboard manipulator is used in which the track profile of the vehicle base can be changed. The yielded system functionality resembles that of a system with flipper mechanisms or rotating track segments. It is categorized here since its approach utilizes track reconfiguration rather than external flipper mechanisms.

An example of a well developed robot that has both enhanced mobility capabilities and manipulation capabilities is HELIOS–IX [19]. The HELIOS robot has been designed specifically for USAR and exhibits some shape shifting qualities as well as some docking abilities, but its main feature is the hybrid use of its manipulator for mobility enhancement [19].

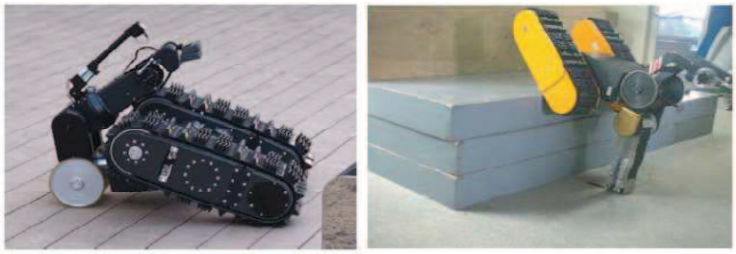

Figure 1-9: a) HELIOS IX, b) HELIOS VIII [19]

Figure 1-10: HELIOS IX, Manipulation [19]

The HELIOS robot uses its manipulator via a passive wheel to save energy on flat terrain, as a lifting device for ledges and stairs, as a centre of gravity adjustment, and finally as an actual

manipulator [19]. The gripper is torque controlled and this feedback is sent back to the remote operator, which causes vibration on their controls [19].

Another novel hybrid mobile robot (HMR) developed is described in [10]. The author of this work harnesses a symbiosis between locomotion and manipulation, somewhat similar to the HELIOS robot. The robot's manipulator and track drive system are integrated, which allows the manipulator to fold completely into the track base [10]. This approach differs from the traditional mobile manipulator approach of having the manipulator mounted on top of the track vehicle base [27] and enables a symmetrical design that functions the same when flipped over. Furthermore, continuous rotation of the joint links is enabled through a wireless network between components [10].

Figure 1-11: HMR, Hybrid Approach to Mobility Enhancement [10]

Through the use its novel integrated design the HMR can perform a variety of manipulation and climbing tasks that allows it to adapt to different situations by using the manipulator and track base in different configurations [10]. The individual DOF of this system are controlled remotely by two joysticks, where some simultaneous inputs are possible as well. Operator practice is required to perform some of the simultaneous motions [10].

Marsupial and Modularity Methods:

In the previously reviewed systems it is clear that great strides have been made in enhancing mobility in unstructured terrains. One thing that the reviewed literature thus far does not show is the ability of the systems to vary their overall size to potentially get into tight spaces, yet manage larger obstacles. Several methods that enable this are examined next.

In [28] researchers investigate the use of a marsupial robotic system for USAR. In their experiments the larger track robot with a manipulator lifts tiny drivable robots into confined spaces and drops them so they can explore and provide information on the confined spaces. The findings are that the cooperation of robots yields functionality not otherwise achievable by single robots [28].

However, in the implementation in [28], there is the potential that the smaller robots may not be able to get back from the confined space, since they have limited functionality. The marsupial approach is similar to a heterogeneous modular approach and is distinguished by there being one large robot acting as a carrier or mover of smaller robots.

Homogeneous modularity can be used as a mobility enhancement method for managing obstacles. It can serve as a technique for scaling the robotic system; enabling it to get into tight spaces as well as overcoming larger obstacles. Since the R2TM3 concept is of this type, the specific systems that employ this type of modularity are described in the next section.

1.3.4 Modular Mobile Robotics

The authors of [29] developed Millibot Trains for enhanced mobility. This system consisted of small mobile modules that are able to connect or disconnect using a pin connector concept. Once the robots connect they are able to lift one another enabling the entire train to overcome more severe obstacles by climbing. Each Millibot is self sufficient with track drives, onboard power, RF communication and control [29]. They were intended to be small to enhance their own manoeuvrability.

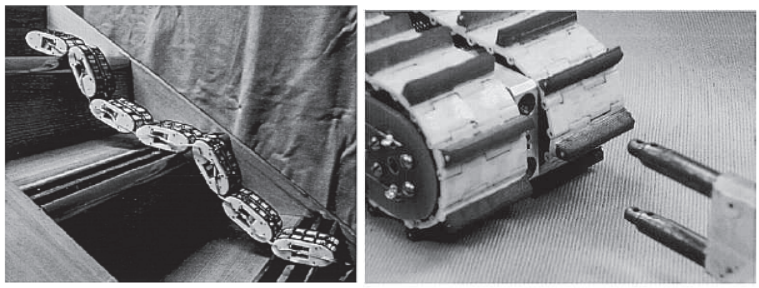

Figure 1-12: Millibot Trains, Robots May Connect for Enhanced Mobility [29]

Although this concept was novel, it still lacked some important features such as the ability to align the docking connector in rough terrain and manipulation. The authors of [29], however, did indicate docking alignment as a future research challenge.

An enhancement of the ideas developed in [29] was presented in [30]. In this work, the researchers developed a system where the modules had 2-DOF at their docking connector. They also used a two stage docking method where permanent magnets were used to connect the modules then rotation of the connector would occur to lock the modules in the roll direction [30]. The male end of the connector was designed with a generous cone shape to tolerate some misalignment. In addition, the

researchers developed a preliminary autonomous docking method using ultrasonic transducers and control states [30], for docking on a flat level surface.

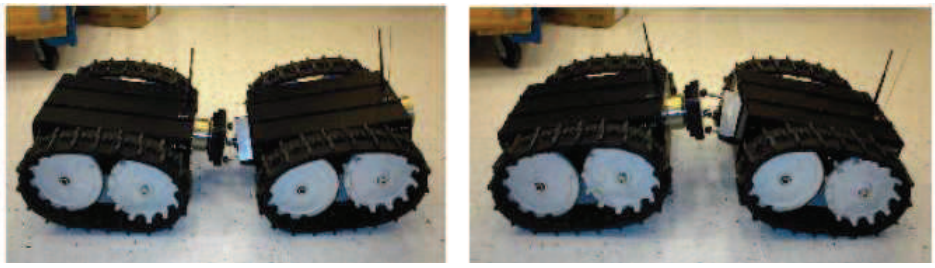

Figure 1-13: Extended Gemini Platform [30]

Although the work in [30] addresses the issue of misalignment, there is still a rather low limitation as to how much docking misalignment can be tolerated. In an unstructured environment like USAR, misalignment could be severe and in any direction. In addition, the rotation range of the connector is significantly less than the lifting range of the Millibot (180°) [29].

As was shown previously in Figure 1-3, the Swarm-bot [15] also resembles a system of mobile modules capable of connecting and lifting one another. However, the approach to the Swarm-bot system is different from the work in [29] and [30]. The Swarm-bot is inspired more by the original modular scientific approach. In the Swarm-bot approach only enough functionality has been added to each s-bot for it to cooperate and communicate locally with its neighbors [15]. The ultimate goal of this approach is a system consisting of a large number of s-bots resulting in a globally useful behavior, such as hole crossing [15]. The aim of the Swarm-bot system is less to do with specific applications and more to do with scientific experimentation of decentralized algorithms [15]. Regardless, it does mechanically resemble that of the systems in [29] and [30], and is worth noting. It could be argued that the difference between mobile MRRs and mobile robot cooperation is a matter of perspective in some cases.

Another modular mobile system developed is JL-1[20]. Similar to the platform in [30] JL-1 has a cone shaped male connector on the front of each module. The main enhancements of this system are the significantly higher ranges of motion of the connector, the added rotational degree of freedom and better misalignment tolerance. JL-1 uses a parallel manipulator for its connector and can achieve +/- 45° posture adjustment in the pitch and yaw directions and can achieve 0° to 360° rotation in the roll direction [20]. The manipulator is capable of arbitrarily changing the posture of its connected module,

due to its high force output. The actuated female coupler on the back of the modules enables module offset tolerance.

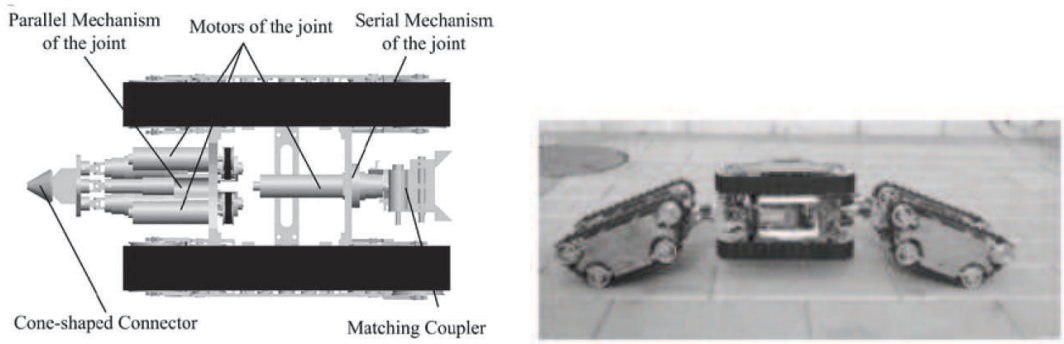

Figure 1-14: JL-1, a Modular Mobile Robotic System [20]

Even with the advancements made in the JL-1 system, it was still realized that in rough terrain, common to applications like USAR, docking may not be possible due to severe misalignments. In addition, it was recognized that JL-1 cannot perform any manipulation [21]. These identified shortcomings led to the development of JL-2 [21]. JL-2 deviates from JL-1 in that its connector is now a dual purpose gripper mounted on the front of the module and on the rear of the module there is a sophisticated and actuated docking disk [21].

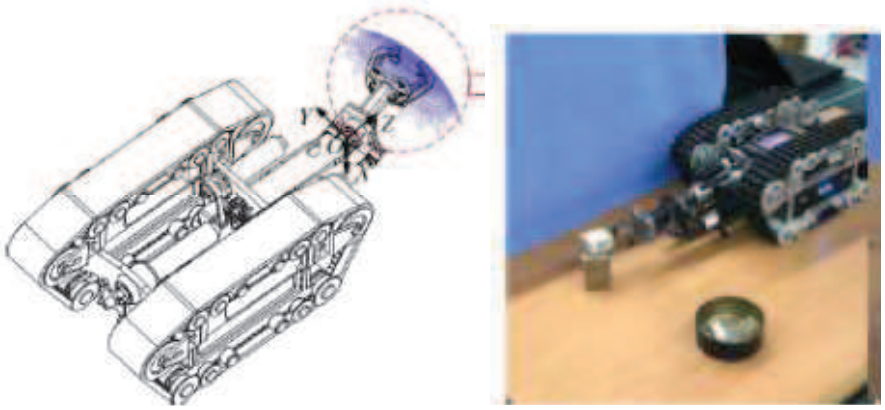

Figure 1-15: JL-2, with Docking Manipulator [21]

The gripper is actuated with a cam system in order to allow for gradual alignment [21]. The docking disk is equipped with a main docking cone and sub-docking cones [21]. The main concept is that the gripper can nip the docking disk then gradually align as it closes. In addition, the gripper cam mechanism effectively locks closed, which removes the need to use energy to keep the connector

closed. This new design leads to similar posturing capabilities, but with the addition of tolerating three dimensional misalignment, as well as basic manipulation [21]. JL-2 employs a distributed control scheme in which each module contains the same control hardware and they are operated remotely by joystick [21]. Some autonomy has been added to JL-2 for some autonomous docking capabilities using sonar sensors [21].

Upon evaluation of the JL-2 design, it can be proposed that a serial manipulator with more degrees of freedom could be used to dock several robots, where the JL-2 uses a parallel manipulator for the powerful posture adjusting capabilities [21]. In [31], [17] the authors propose the idea of using a serial manipulator as a viable docking device for a variety of terrain situations. In [31] the authors propose the GR system and suggest that docking several robots would result in greater efficiency when travelling through undulated terrain. They also suggest, that if the manipulator was strong enough, it could enhance gap crossing [31]. Finally, they conducted experiments with a passive manipulator mechanism connecting two robots for climbing several stairs [31]. They propose cooperation between two robot modules as a way of easing the required force a given robot needs to overcome on a slope or stairs, by creating a "virtual angle of inclination" [31]. Furthermore, their experiments show that a single robot could not climb the stairs without tumbling over, which is due to the stabilizing effect of the robot connection rather than active robot cooperation.

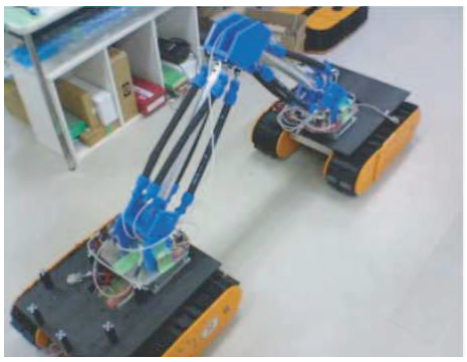

Figure 1-16: Two HELIOS Carriers Connected by a Pneumatic Arm [17]

In [32] the authors use a simple passive link between mobile bases and demonstrate that small gap crossing can be achieved cooperatively this way, showing that connecting the robots in this manner can have an inherent tendency to enhance mobility. In [17] and [33] the docked manipulator idea is applied to the HELIOS carriers (which normally do not have their own manipulators), where HELIOS VIII was intended as the robot with the docking manipulator. In [17], an experiment is done where

two carriers are connected with an arm powered by pneumatic artificial rubber muscles (PARMs) on several of the joints and they attach with an air gripper, seen in Figure 1-16. Each connected side has two active DOF and a passive joint in the middle of the manipulator. A similar experiment as done in [31] is performed, but the PARMs are manually activated by a remote operator to assist in the stair climbing. As in [31] the approach is to cooperate to reduce the "virtual angle of inclination" and stabilize the robots. The method aims to continually keep the robots in contact with the ground or slope/stairs.

The method of docking robots via an onboard serial manipulator is potentially an excellent method of robot or module cooperation and is the method employed in the R2TM3 system. Several observations of the work done in [31], [17] and [33] are as follows:

- The cooperative method is limited to moderate obstacles such as stairs, slopes or small gaps. With two robots, the limited connected range of motion and the given cooperation methods, a robot could not be pitched up off the ground to a severe angle. The system's ability to move a module for obstacle management is much less than that of the parallel mechanism used in [21], in regard to force output.
- The HELIOS VIII manipulator, or any of the other tested manipulators, does not have a base joint that allows rotation in the ground plane relative to the vehicle base. The docking location is asymmetrical about the vehicle's geometrical centre. There is no roll joint to cope with misalignments for docking.

These specific observations are made since they pertain to the formulation of the R2TM3 concept discussed subsequently. Given that the concept will employ an onboard manipulator, the next section provides some background on robotic manipulators.

1.4 Robotic Manipulators

Some background on robotic manipulators, within the context of this thesis work, is provided to highlight some key aspects. A robotic manipulator can be defined as, "… a reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks." [34]. Implied in the definition is the flexibility of robotic manipulators, due in part to their re-programmability. They have

been in use in factory settings for many years and their technology is well developed. In this thesis a manipulator is considered as a serial robotic manipulator with revolute joints.

As mobile robots become more common, so do mobile manipulators [27]. The use of manipulators in mobile robotics provides additional functionality such as enabling better sensing by having a camera or other sensory on the end of the arm and enabling mobile manipulation. Specifically for USAR this allows for enhanced search capabilities and situation awareness with a manipulator camera, the ability to deliver or gather packages and to possibly open doors to otherwise inaccessible rooms [19]. Having an onboard manipulator on a mobile base poses several new challenges in the control of manipulators such as maintaining robot stability and manipulator-vehicle interaction [27].

In USAR, controlling a manipulator in the unstructured environment is handled primarily by teleoperation with a well developed operator interface [10], [19], [17]. Having low level joint feedback controllers with sent motion commands via teleoperation is a good and typical approach [19], [21].

As mentioned in the previous literature review several systems utilize symmetry and/or a completely integrated manipulator design to negate the robot stability problem [10], [19]. In other systems, an active stability method is used to avoid tip-over by modifying the manipulator configuration [27]. In the case of the robot tipping over, the manipulator would have to be used to push the vehicle upright, if possible. An active tip-over avoidance approach may also serve to enhance vehicle steering, since a stable state implies a state in which the vehicle-ground forces are as evenly distributed as possible. The manipulator may also be teleoperated to adjust the centre of mass (COM) when climbing a slope [19]. In any case, good design practice is to keep the manipulator as light as possible and keep it in a stable configuration whenever possible; to keep the system's overall center of mass (COM) close to the ground.

Utilizing the mobile vehicle base DOF and manipulator DOF in coordination, to handle manipulator-vehicle interaction, depends on the specific system and level of integration. In [10] the operator is intended to coordinate the manipulator's DOF with the base's in order to enable the hybrid approach in locomotion and manipulation. For these types of systems in general, it may be adequate to operate the manipulator independently, while the base remains still and manipulator loading is assumed to be adequately reacted by the base and ground.

For any robotic manipulator a general control approach needs to be considered. In a general sense, this can be looked at as nonlinear multi-variable dynamic control or linear single input/single output (SISO) independent joint control [34]. In the first case the dynamic coupling between manipulator links is modeled in the control scheme and the system is linearized by a nonlinear feedback control law, which is referred to as inverse dynamics [34]. This allows for higher speed and more precise control, possibly with direct drive actuators. In the second case, each joint has its own independent SISO linear feedback controller and the dynamic coupling between the links or payload is considered as a disturbance to the control loop [34]. This approach is appropriate for applications that have relatively low acceleration and velocity, which reduces the dynamic effects [34]. Furthermore, the gear ratios at the manipulator joints serve to decouple the link dynamics; the greater the gear ratios the more diminished the dynamic coupling is [34].

In the case where teleoperation is used, such as USAR, and the manipulator is not used at high speeds or for extremely precise movements, the independent joint control approach is likely sufficient and is the approach used for the R2TM3 concept prototype. An approach for finding manipulator joint set-point positions may be to command the manipulator wrist position, then to use inverse kinematics.

1.5 Discussion and Introduction to R2TM3

In this introductory chapter, the motivation for the work and the primary intended application has been stated, focused on the research problem of limited mobility in unstructured environments. The areas of modular reconfigurable robotics and mobile robotics have been reviewed within the framework of the research direction, leading to a closer examination of modularity and mobility combined. Considering existing mobility enhancement techniques, including modular mobile systems, it can be concluded that utilizing the modular mobile approach and using a serial manipulator as the docking device has great potential; specifically if the docking is rigid, the manipulator has many DOF and is active and if the robot tracks cooperate with the manipulator. As a result of these findings, the Reconfigurable Robot Team of Mobile Modules with Manipulators (R2TM3) has been developed and will be described in detail in Chapter 2.

Figure 1-17 and Figure 1-18 introduce the R2TM3 concept prototype. It is intended to handle severe obstacles and to be highly manoeuvrable.

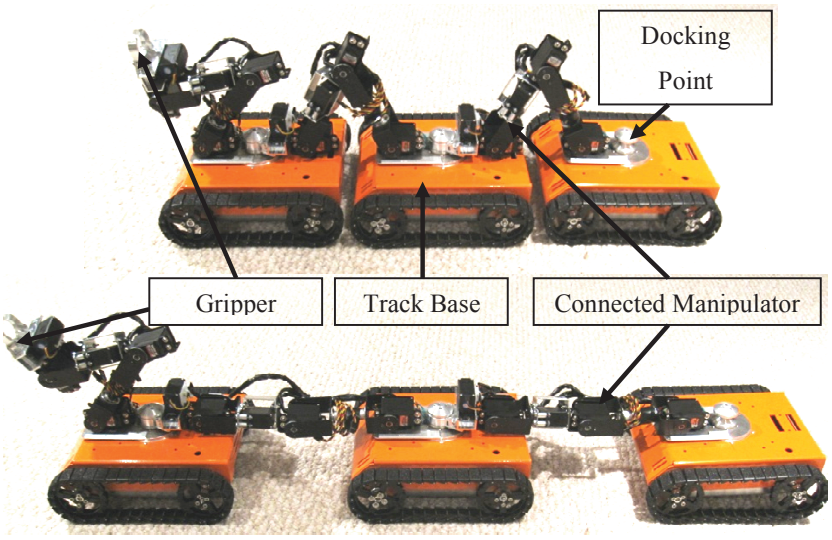

Figure 1-17: The R2TM3 Prototype, Connected Configuration

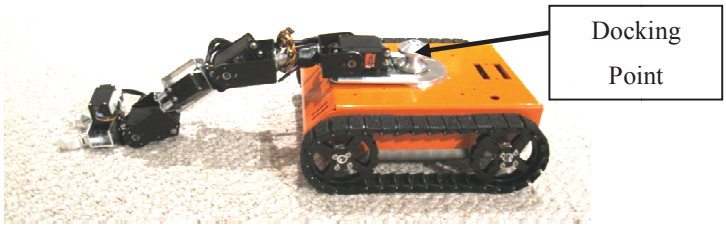

Figure 1-18: Single R2TM3 Robot Module

Three robot modules have been constructed for the prototype, since this is the minimum number required for the severe obstacle climbing approach; for example, when the first one is pitched up by the second one, the third one ensures the second one stays on the ground. This will be described in detail in Chapter 2. In this scheme more than three robot modules could be used.

During the development and construction of R2TM3, its concept was introduced and described in [35] where a simulation was done based on its configuration. The simulation was done for three connected modules using a genetic algorithm for manipulator pose optimization. In a prior work [36], by the author, the general idea of enhancing mobility through modularity was utilized. In this work, reconfiguration path planning using a wavefront algorithm was simulated as well as a cooperative climbing algorithm that optimizes the module ground contact while traversing a known ground profile. The early work from [36] helped to inspire further research in this area that has led to the work presented in this thesis.

1.6 Contributions

In this thesis work, a fully realized mobile modular robotic system (R2TM3) is conceptualized, developed, constructed and tested, to enhance robot mobility. In prior work similar concepts have been proposed, but the concept of the R2TM3 differs or aims to expand on prior ideas. The specific contributions of this thesis work are:

Concept Development:

- The concept uses the manipulator as a docking device; it docks rigidly and symmetrically with a high range of motion. A passive docking joint is also used to allow for shifting manoeuvres. The concept approach allows for extreme obstacle climbing via track and manipulator cooperation. The concept development is described in detail in Chapter 2.

Proof of Concept Mechatronic Prototype Design:

- A fully realized mechatronic design, including mechanical design, control hardware architecture and component integration. More details are in Chapter 2.

Controls, Implementation and Testing:

- Implementation of robot controls and testing of robots; single robot and cooperative robots, including extreme obstacle climbing. See Chapter 3 and Chapter 4 for details.

1.7 Thesis Organization

This thesis consists of five chapters including the introduction. The details on the mechatronic design are described in Chapter 2 which includes; concept development, mechanical design and analysis, control hardware architecture and the electrical power system. These topics are under one chapter to emphasize the mechatronic approach. The robot controls, both for a single robot and for robot cooperation, and teleoperation are described in Chapter 3. The implementation, test methods and test results are given in Chapter 4. Finally, conclusions and suggestions for future work are given in Chapter 5.

Chapter 2

Mechatronic Design

2.1 Concept Development

The developed concept aims to improve or expand on existing mobility enhancement solutions, through a unique mechanical design, cooperative climbing strategy and manoeuvrability methods. This chapter section will outline the concept development and rationale in detail.

2.1.1 Concept Objectives

From the prior literature review and an account of the primary intended application, several desirable key attributes for these types of robotic systems can be extracted and are described in Table 2-1.

Table 2-1: Primary Desired Attributes for Mobile Ground Robots in USAR

Primary Attributes	Description
Enhanced Mobility	Enables severe obstacle climbing, fitting into confined spaces, enhanced traction and enhanced stability
Situation Awareness[1]	Sensing ability for knowledge of robot location and surroundings
Robustness	Ability to resist or withstand damage through overall construction or redundancy
Manipulation	For delivering of packages, retrieving packages and opening doors

Given the description of the desired attributes, several design features are compared. The assessment uses a 0 rank for neutral and 1 for a positive correlation.

Table 2-2: Design Configuration Selection Matrix for Mobile Ground Robots

		Design Features				
Attributes	Weight	Modularity	Track Flippers	On Board Manipulator	Reconfiguration or Hybrid	Symmetry
Enhanced Mobility	1	1	1	0	1	1
Situation Awareness	1	0	0	1	0	0
Robustness	1	1	0	0	0	0
Manipulation	1	0	0	1	0	0
Total	4	2	1	2	1	1

Table 2-2 shows that there are a number of configurations that can effectively satisfy the primary attributes. For the R2TM3 it was decided to combine modularity with an on board manipulator yielding a total score of 4. In the case of HELIOS IX [19] and the HMR [10] the systems primarily use a hybrid approach, with a manipulator and the HMR employs symmetry yielding a total score of 4 as well. In the case of modularity there is an opportunity to have the effective size of the system change, in addition to the module cooperation it enables, which is not accounted for in Table 2-2 due to the simplicity of the assessment.

[1] This definition is limited to this robot type. Other types of robots would also likely be used as part of a larger heterogeneous team for better situation awareness.

Based on the decided configuration, the general concept rationale is defined. In Table 2-3 higher resolution concept objectives, specific to the R2TM3, are shown along with the concept solutions.

Table 2-3: Concept Objectives for the R2TM3 Concept

Concept Objectives	Solution
Docking in a Variety of Terrain Situations	5-DOF Manipulator with base and roll joint
Climbing Severe Obstacles	Three or more modules utilizing a cooperative strategy between tracks and manipulators
Connected Manoeuvrability	Base joint on manipulator, passive docking joint and sufficient clearance between tracks
Low Power to Remain Docked	Toggle docking mechanism
Simple and Compact Docking Mechanism	Dual gripper and docking mechanism
Stability in Rough Terrain	Light weight manipulator

With the primary concept objectives defined, based on the selected configuration, the developed concept can be described in detail. Further development and implementation of the concept focuses on mobility enhancement. However, the inherent concept configuration lends itself to all of the attributes listed in Table 2-1.

2.1.2 Concept Description and Notable Features

The R2TM3 concept prototype possesses a number of notable features as listed in Table 2-4. The listed features closely resemble the solutions from Table 2-3, since they were derived from concept development.

Table 2-4: Notable R2TM3 Concept Features

Feature	Description
Docking system contains a passive joint.	Enables connected manoeuvrability; steering and shifting.
Docking location at geometric center of mobile base.	Allows for connected manoeuvrability and symmetrical force interaction between modules. Manipulator link lengths designed to leave clearance for rotation.
5-DOF manipulator.*	Enables docking in severe terrain with high misalignments. Also provides connected manoeuvrability with base joint.
Cooperative climbing strategy.	Enables climbing of severe obstacles (ledges) without having large, powerful and heavy manipulator or a low DOF parallel manipulator.
Gripper serves as docking mechanism.	Inner profile of gripper suitable for grasping and docking.
Toggle lock docking.	Linkage that moves gripper locks at input toggle position, requiring minimal power to stay docked.
Homogeneous modular system.	Enables robustness through redundancy; each robot has its own control hardware and power supply
Each module fully functional.	*Self mobile and capable of manipulation

The R2TM3 concept uses several operational modes to achieve different configurations:

- Single Robot Drive (0): Vehicle speed and steering.
- Single Robot Manipulation (1): Forward/back, up/down, base joint, wrist rotation, gripper and manipulator on/off.
- Cooperative Climbing (2): Robots in connected configuration; manipulators and tracks cooperate to climb a severe obstacle such as a ledge.
- Cooperative Steering (3): Robots in connected configuration steer cooperatively.
- Cooperative Shifting (4): Robots in connected configuration shift for convenient lateral movement.

Modes 0 and 1 are used to control a single robot in the group, or all of the robots identically and simultaneously. Track and manipulator control could have been more integrated for single robot control, but this is left as future work since it is not the focus of this research.

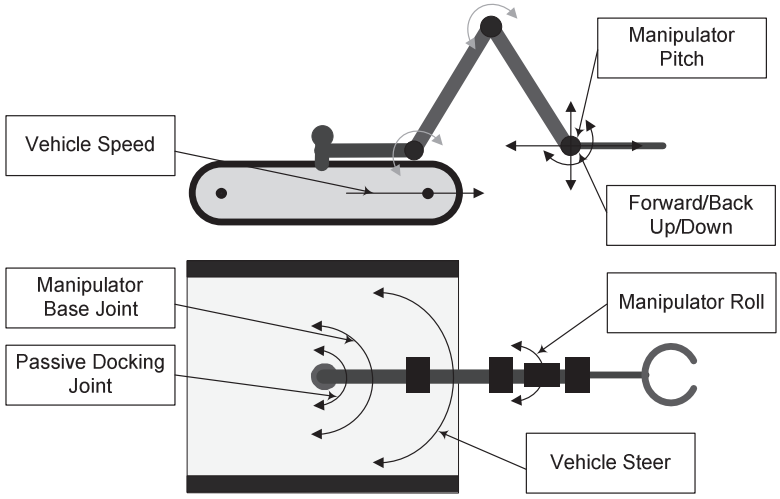

Figure 2-1: Robot Drive and Manipulator

The selected manipulator configuration in the concept is an articulated manipulator (RRR) type known as an elbow manipulator [34]. The base revolute joint axis is perpendicular to the other revolute axes. In addition it is equipped with two wrist axes for roll and pitch. This 5-DOF arrangement allows for a relatively large workspace, connected manoeuvrability and, due to the base and roll joint, can enable docking in severe terrain misalignments.

Mode 2 is used for cooperative climbing. It can be used for climbing a variety of obstacles and is intended for severe obstacle climbing. Figure 2-2 depicts the climbing strategy, where the first module climbs the obstacle. The procedure would continue until all modules have overcome the obstacle.

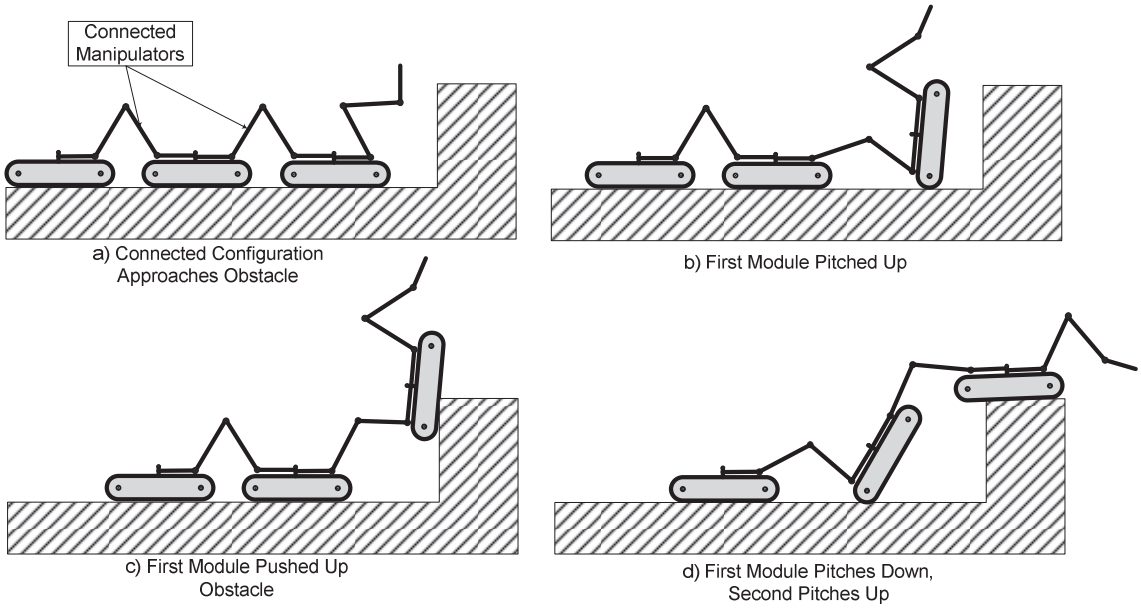

Figure 2-2: Cooperative Climbing Concept

The active cooperative climbing strategy allows for the use of a light 5-DOF serial manipulator to perform climbing that would otherwise be impossible with the manipulator alone or with a passive linkage relying only on the tracks.

Mode 3 is used to keep the robots in formation while traveling together over rough terrain. In this mode the manipulator is turned off to ease the loading on it. In this mode the robots cooperatively form an arch to steer around obstacles. The manipulator link lengths are designed to allow clearance for vehicle base rotation in this mode and Mode 4. The connected manipulators serve to also passively stabilize the modules. From this mode it is easy to quickly switch into Modes 2 or 4 as needed.

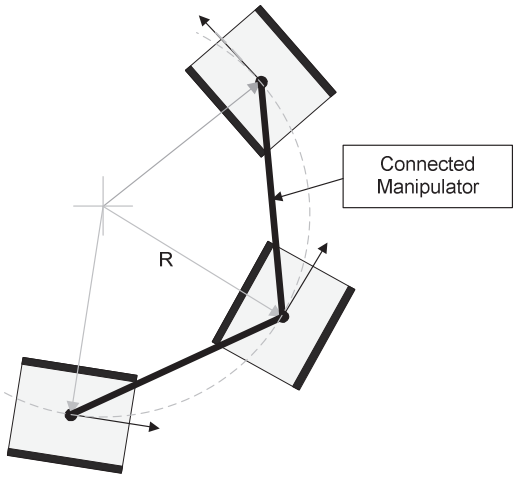

Figure 2-3: Cooperative Steering, Top View

Mode 4 is used as an additional manoeuvring method to Mode 3. It gives a more direct means of lateral movement than Mode 3 does. This provides similar locomotion to that of an omnidirectional platform.

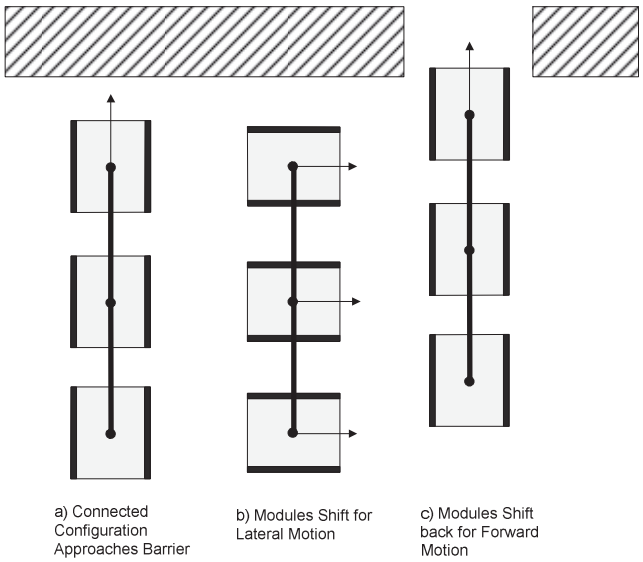

Figure 2-4: Cooperative Shifting, Top View

An additional mode of operation, with four robot modules, is possible and is referred to as the Quad configuration mode. This configuration serves as a means to scale the size of the robot vehicle

in both length and width. The Quad configuration is not implemented at this time since the research focus is on obstacle climbing and manoeuvring, and only three robot modules were built. Figure 2-5 depicts the configuration and how this may emulate a larger robot.

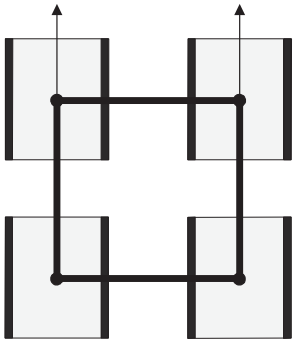

Figure 2-5: Conceptual Quad Configuration

The approximate scale of the system concept is determined by several factors. Firstly, the size of existing similar robotic systems was considered as a heuristic [19], [21], [18]. This was taken to be the equivalent to the overall size of the Quad configuration. Taking into account the needed clearances between the tracks for vehicle base rotation, the overall size of the vehicle base was derived. However, if the individual modules are too small then they could be susceptible to getting stuck on common obstacle types such as stairs. It is best if the track length of the vehicle is long enough to ride over stair ridges without needing to perform special manoeuvres. Consequently, it was decided that the ideal length of the vehicle base is between 305 mm (12 in) and 356 mm (14 in) and the width between 254 mm (10 in) and 305 mm (12 in).

The factors that need to be considered for concept implementation are considered next, given a low cost design principle.

2.1.3 Concept Prototype

For the realization of the concept prototype, several compromises had to be made, in order to reduce cost and time spent. It was decided to select existing hardware for purchase for the various standard components. Components were then modified to suit the design and custom design solutions were added for nonstandard parts.

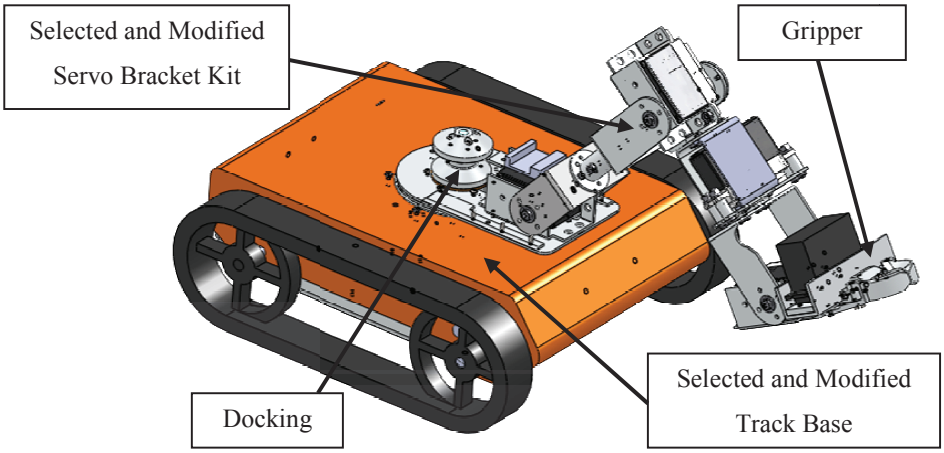

Figure 2-6: Concept Prototype

It is not necessary for the proof of concept prototype to be robust enough for field use. To keep costs down, the constructed prototype is intended for indoor and infrequent lab testing.

The selected track vehicle base is smaller than the specified concept size at 222 mm (8.75 in) long and 203 mm (8 in) wide. It is scaled down by approximately 2/3 to 3/4 of the preferred size with respect to length and width. Consequently, the entire prototype is scaled down. This means that the prototype track length would not be ideal for riding on standard stairs. Also, the gripper is scaled down such that it could not grasp a standard door knob. However, this is just a matter of relative size between the robot prototype and its surroundings. For testing of the concept, performance can be measured in relative terms between the robot size and obstacle size. The selected track vehicle base, track design is also quite simple. Its traction characteristics are not ideal. More details on the tracks will be given in 2.2.1.

Owing to the low cost prototype principle, the manipulator joint motors are of the radio control (RC) servo type, which poses certain limitations as well; angular range of motion, torque and control techniques. More details on the servo motors will be given in 2.2.1. In addition, the manipulator configuration does not allow the manipulator to fold compactly when not in use, due to the simplified design implementation (purchased bracket kit). Again, a low cost solution was sought for the on board robot computation. A 32 bit microcontroller is used in each robot. As stated previously, only three robot modules have been constructed as the minimum required to assess the concept performance. The prototype provides a low cost minimalist solution to enable realization of the concept for testing. Most importantly, the stated limitations do not inhibit the research goals.

2.2 Mechanical Design and Analysis

The design of the R2TM3 utilized CAD software in order to generate 3D data representing the robot components, generate 2D working and assembly drawings, to make sure the designed or selected components fit properly and to perform stress analysis as needed. The CAD software also provided a powerful means to visualize the design outcome. In some cases 3D CAD data was taken directly and used for CNC machining. This CAD/CAM process allowed for better communication of the design concept and allowed for more complex components to be developed without a considerable manufacturing cost increase.

The mechanical design for the robot module was performed so that it would function for its indented purpose, but would not contain additional features that are not critical to the concept objectives. The remainder of this section will outline the details of the mechanical design and analysis.

2.2.1 Prototype Component Integration

The selected track vehicle base is a modified Traxster II robotic platform from Summerour Robotics Corp. After a review of existing platforms, the Traxster II was selected due to its relatively low cost for its size, its larger internal volume for component housing, its flat top surface for mounting and its track configuration[2]. The size of the Traxster II is closest to the desired R2TM3 concept size as stated in 2.1.3. Its larger internal volume is due to it not containing additional actuators or for unnecessary features and its taller track height. The tracks extend beyond the chassis ensuring that it does not interfere when climbing severe obstacles, giving an approach and departure angle of 90°. However, there are several modifications that needed to be made to the Traxster II in order for it to be appropriate for use, as shown in Table 2-5.

[2] It was later discovered that these platforms were on hand and available for use, making them an even more cost effective solution.

Table 2-5: Track Base Modifications

Problem	Solution (Modification)
Existing belly pan passes through the overall height of base, limiting the overall size of components that can be housed. Intended for external battery mounting.	Design custom belly pan that extends internal volume to allow for larger components to be housed.
Adapter needed to integrate manipulator base joint and docking joint.	Cut large centre hole and hole pattern for a custom adapter design.
Tracks are made of smooth plastic with a low coefficient of friction.	Coat track links with spray-on rubber.
Various components need to be mounted where no hole patterns exist.	Drill new holes as needed or use adhesive mounting.

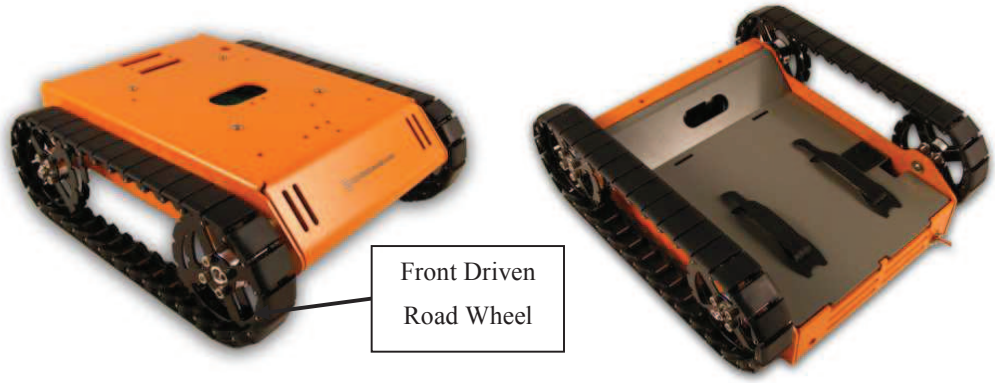

Figure 2-7: Unmodified Traxster II

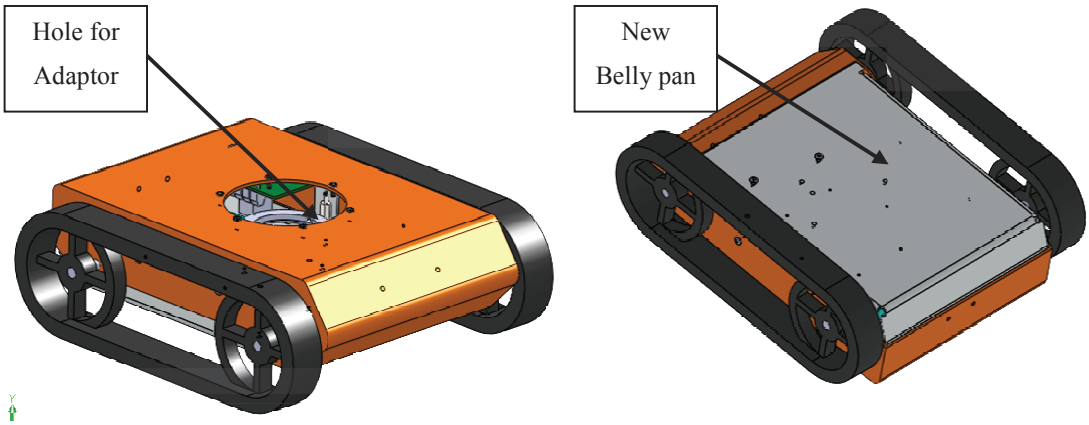

Figure 2-8: Traxster II Modifications

The track vehicle base chassis is made from aluminum with a powder coated surface. It comes with two DC gear motors for the track drive; one for the left and one for the right. This configuration yields a typical skid-steer vehicle type. Each motor is equipped with an optical quadrature encoder that allows for speed and direction measurement. The motors drive the front road wheels which drive the tacks.

The track links are made from molded smooth ABS plastic connected to one another with small press-fit stainless steel pins. This design is simple, yet functional, but the tracks lack tension between the road wheels. This can result in poor traction characteristics in the middle of the tracks. In addition, the track links do not have grousers, protrusion on the surface of the track links, that would normally be used to enhance traction. However, all things considered, the track vehicle base provides what is needed for the concept prototype.

The manipulator is constructed out of a selected bracket set called the Servo Erector Set from Lynxmotion Inc. The bracket set consists of several aluminum brackets, tubes and hubs that can be attached in a variety of ways to achieve a desired configuration. The bracket set is intended to be used with standard hobby RC servos in terms of size, mounting and output drive. The 5-DOF manipulator has been constructed out of this bracket kit along with custom designed portions such as the roll joint, the base joint and the gripper. The selected bracket kit enables a symmetrical design in terms of manipulator link lengths.

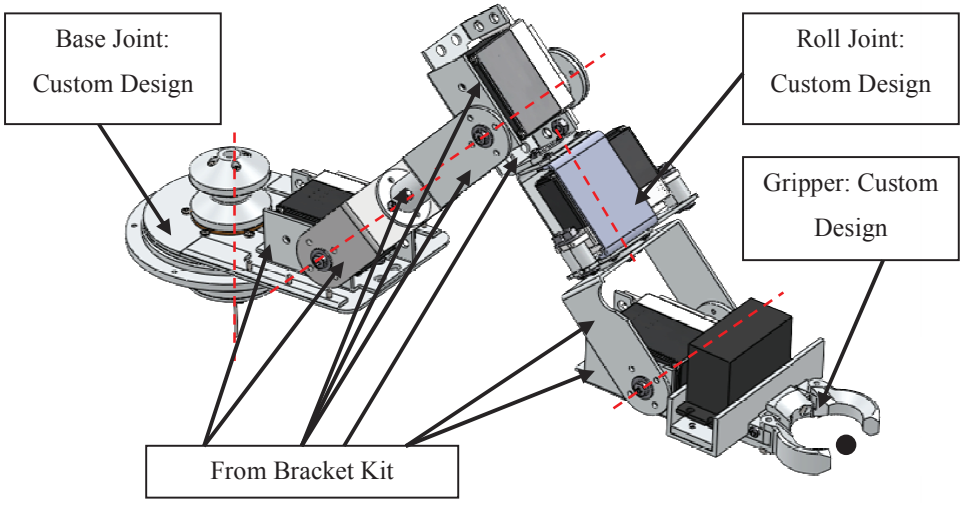

Figure 2-9: Manipulator Bracket Configuration

The selected RC servos for the manipulator are the HiTec HS-645MG high torque servo and the HiTec HS-422 standard servo. These two motors are approximately the same size, mount the same way, have the same output drive hub and are controlled the same way. They differ in internal construction, cost and output performance. The high torque, higher cost, servo is used for all manipulator joints except the gripper, where the standard servo is used. These servos were selected based, in part, on the performance requirements determined from the quasi-static mechanical analysis described in 2.2.3. These servos were also selected based on the need to fit the standard kit and their low cost. Higher performance servos are available, but cost increases drastically, since they utilize a newer digital servo technology. Another specific attribute of position controlled servos is that their range of motion is limited to approximately 180°. In some cases, this can be a hindrance in design performance, but this is actually adequate for the R2TM3 concept prototype. Most of the manipulator joint's range of motion is limited partly by mechanical interference between the links, rather than the servo motor limits. For the base joint a 180° range of motion is enough to perform the intended functions, but a larger range would be better for increased manipulator work space. In order to better utilize the given servo range of motion, the motors were mounted with an angular shift with respect to the connected links, resulting in the joint ranges shown in Figure 2-11[3].

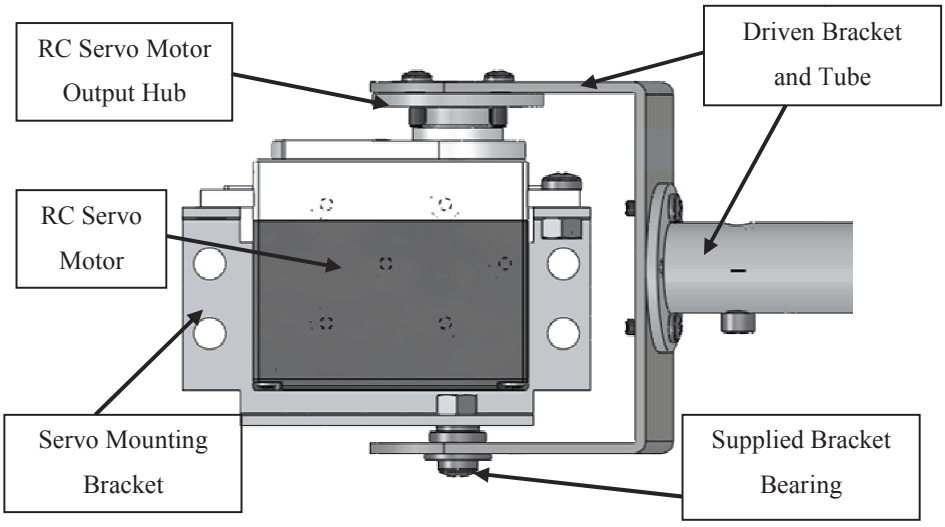

Figure 2-10: Typical Servo Motor Bracket Arrangement

[3] Formal joint frames of reference will be given in sub section 2.2.3 along with D-H parameters

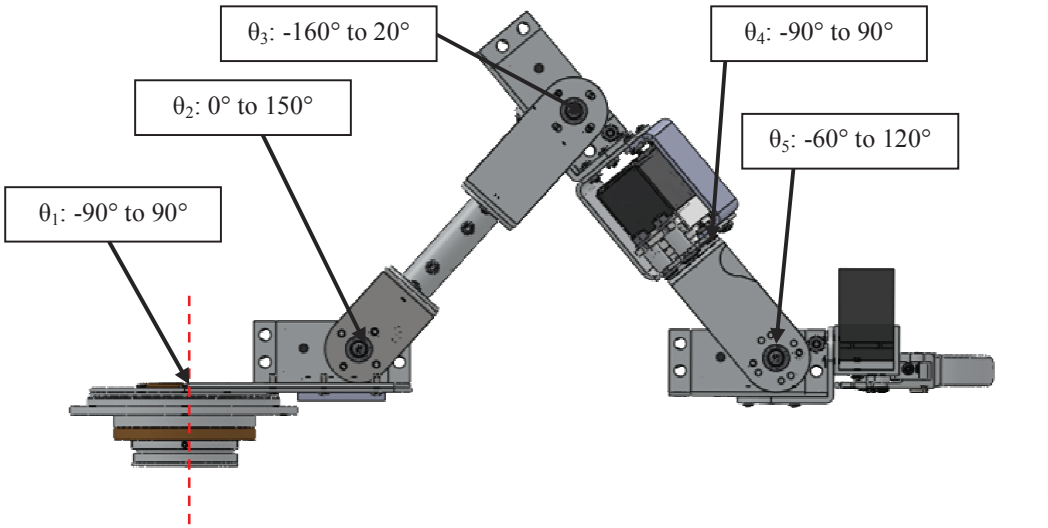

Figure 2-11: Manipulator Joints Range of Motion

The base joint of the manipulator needed to be custom designed and integrated into the vehicle base and, as mentioned, an adaptor plate is used to help achieve this. The manipulator is attached to the base by fastening joint 2's servo bracket to the base joint's output flange. The base joint is diametrically oversized in order to provide space for the coincident passive docking joint, discussed in 2.2.2. The base joint's motor is housed inside the vehicle base and it drives the joint via a gear set with a gear ratio of one. The gears aren't used to change the speed or output torque, but as a way to connect the motor's output with the joint. The large gear diameter is due to the joint being diametrically oversized. A modified bronze bushing and nylon are used to reduce joint friction between moving parts. Additional hardware and structural components are added to secure and stiffen the joint. The driven gear has been modified to fit around the base joint cylinder and to have two keyways. Two small screws are used to transmit torque between the gear and joint.

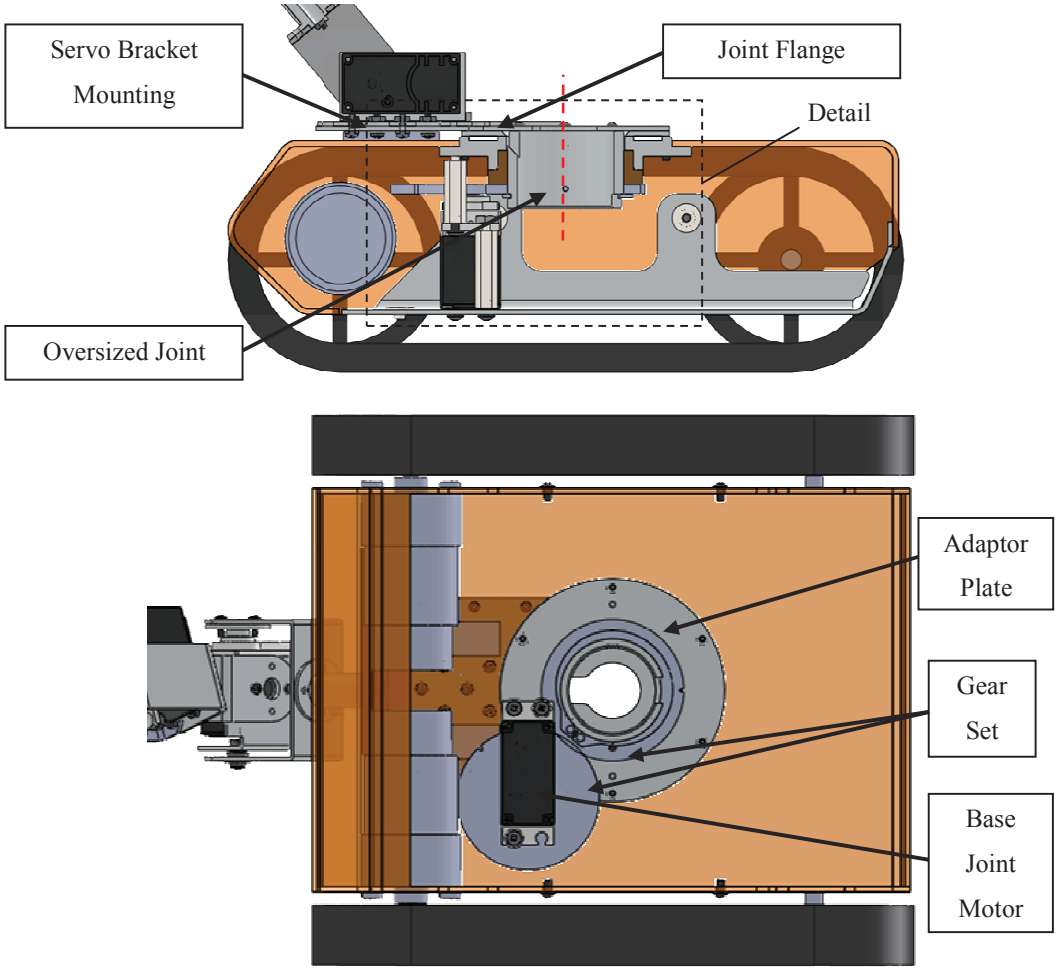

Figure 2-12: Base Joint Design

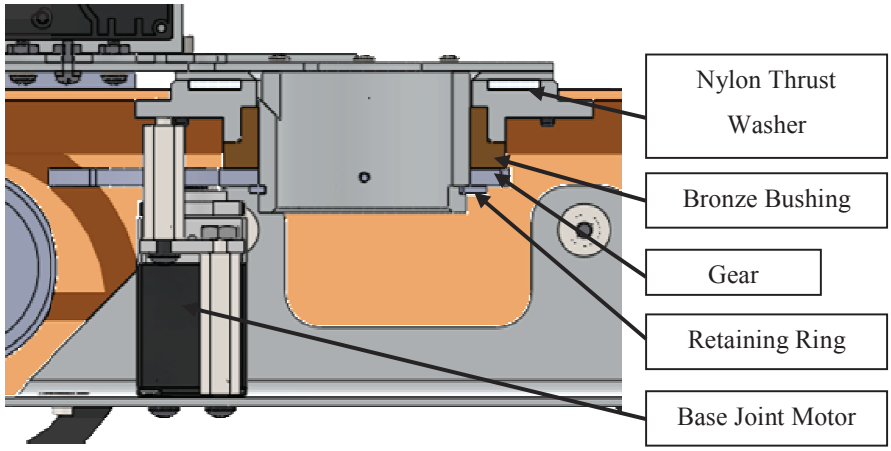

Figure 2-13: Base Joint Detail

The roll joint was placed between manipulator joints 3 and 5, since the manipulator link length was long enough to fit a joint motor in this location, but no existing bracket in the selected kit could provide this configuration. The body of this style of motor is subject to the loading experienced on the motor's output hub. Therefore, a concern for the roll joint is the shear and bending placed on this motor's hub. With this concern in mind, the mounting bracket was designed to fit tightly around the output hub's small diameter, so that bending and shear loads would be shared by the bracket as well as the motor. Furthermore, an additional top bracket was designed to greatly enhance the stiffness on the entire link with respect to bending. For the connection to the next bracket a small piece of aluminum tubing was used in the center of the fastener hole pattern to take shear loading, rather than the fasteners themselves.

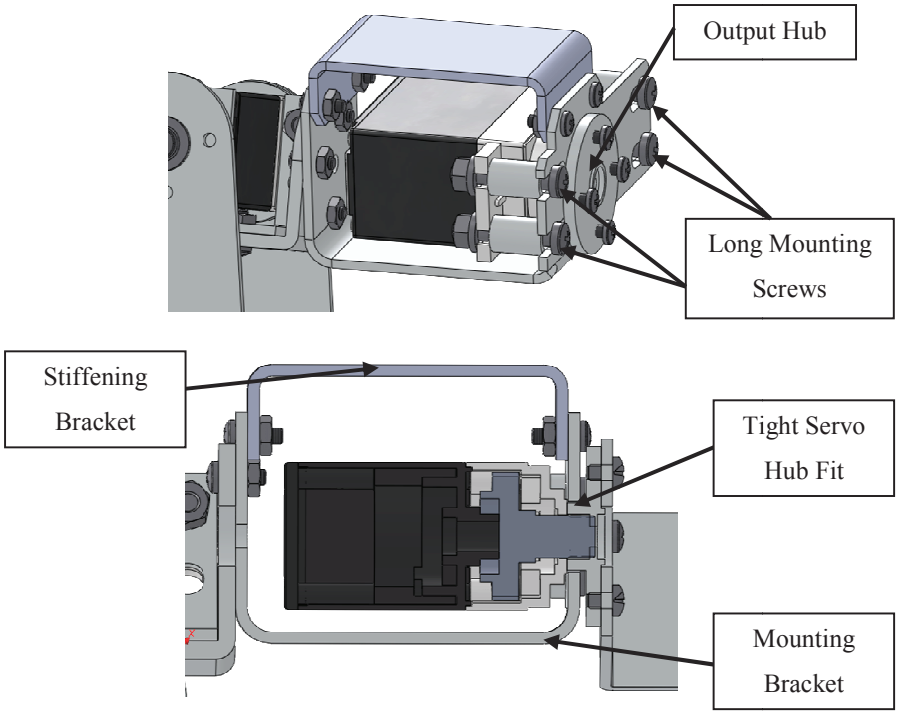

Figure 2-14: Roll Joint design

Other components related to the control hardware and electrical power system needed to be integrated into the prototype design. Figure 2-15 shows some additional component integration features. The docking system (joint and gripper) will be described in detail in 2.2.2.

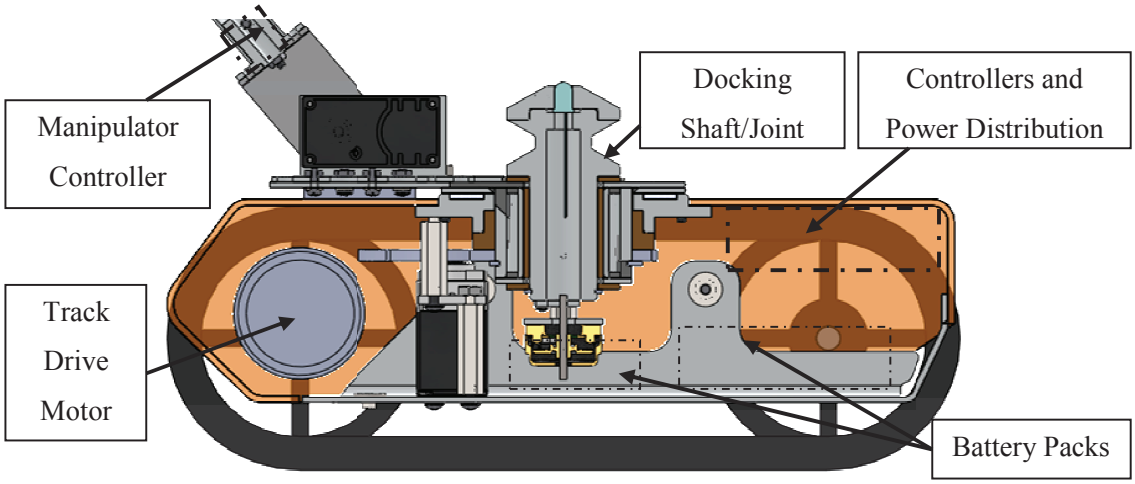

Figure 2-15: Additional Component Integration Features

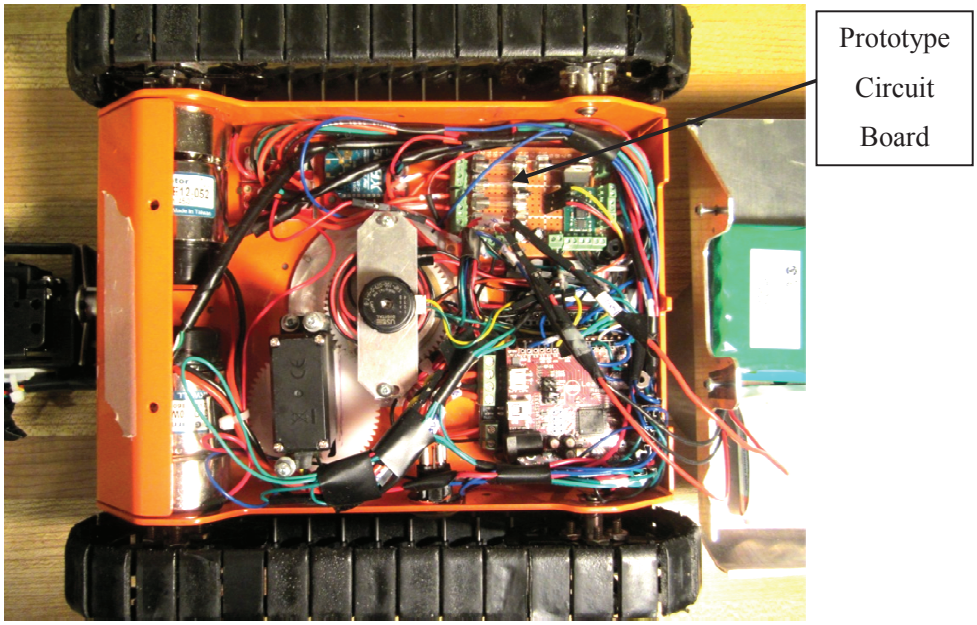

Figure 2-16: Complete Component Integration with Wiring

The selected track vehicle base and manipulator arrangement from the bracket kit are a well suited match, since their size and weights are in the appropriate ranges relative to one another. The link lengths fit well with the track vehicle base dimensions, giving ample connected clearance for the tracks to rotate and the manipulator is light in comparison to the track vehicle base. Fortunately, the selected motors are light weight, which enables the manipulator to lift its own mass with the motors mounted at the joints. Having the manipulator motors mounted at the joints leaves the top surface of the vehicle base clear for base joint rotation and clearance for a docked manipulator to rotate about the docking point. The resultant prototype specifications due to the component integration are shown in Table 2-6. Mechanical motor specifications are shown in Table 2-7.

Table 2-6: Prototype Geometrical and Weight Specifications

Specification	Value (units)	Description
Track Vehicle Base Length (l_{base})	222 mm (8.75 in)	-
Track Vehicle Base Width	203 mm (8.00 in)	-
Track Vehicle Base Height	76 mm (3.0 in)	To top surface of vehicle base chassis.
Track Vehicle Base Weight	20.0 N (4.5 lb)	Includes batteries, other internal components, docking joint and base joint.
Manipulator Link Length (l_m)	114 mm (4.50 in)	-
Manipulator Offset Link Length (l_{mo})	64 mm (2.50 in)	From docking joint to joint 2, or joint 5 to docking point when docked.
Manipulator Weight	5.3 N (1.2 lb)	Includes motors and gripper.
Max. Docking Length (l_d)	356 mm (14 in)	2 ($l_m + l_{mo}$)
Max. Track Gap	133 mm (5.25 in)	$l_d - l_{base}$

Table 2-7: Mechanical RC Servo and Drive Motor Specifications

Specification	Value (units)	Description
High Torque Servo Motor[4]	-	Used for manipulator joints
Stall Torque	0.755 Nm (6.68 lb-in) to 0.932 Nm (8.33 lb-in)	Varies with supplied voltage between 4.8 V and 6 V
Range of Motion	180°	-
Standard Servo Motor	-	Used for gripper/ docking
Stall Torque	0.324 Nm(2.86 lb-in) to 0.402 Nm (3.55 lb-in)	Varies with supplied voltage between 4.8 V and 6 V
Range of Motion	180°	-
Track Drive Motor	-	-
Gear Reduction	1:52	-
Stall Torque	1.77 Nm (15.63 lb-in)	-
No Load Speed	132 RPM	-
Range of Motion	360°	Continuous

2.2.2 Docking System Design

The R2TM3 docking system consists of a docking gripper and docking shaft with a tapered groove. The gripper's inner profile matches that of the tapered groove in the docking shaft. The taper serves to self-align the gripper links as the gripper closes and serves as the contact surface for load transfer.

[4] Speed is not a concern with the RC servos. They respond very quickly.

As described in 2.2.3 the output force of gripper increases severely, for a given motor torque, as the gripper approaches it closed position. Therefore, during docking, the contact forces between the gripper and shaft are much greater the closer to being docked they are, which serves to help alignment; lower forces when misaligned will prevent edges from gouging, while the higher forces when close to the docked position will generate high friction to keep the joint locked.

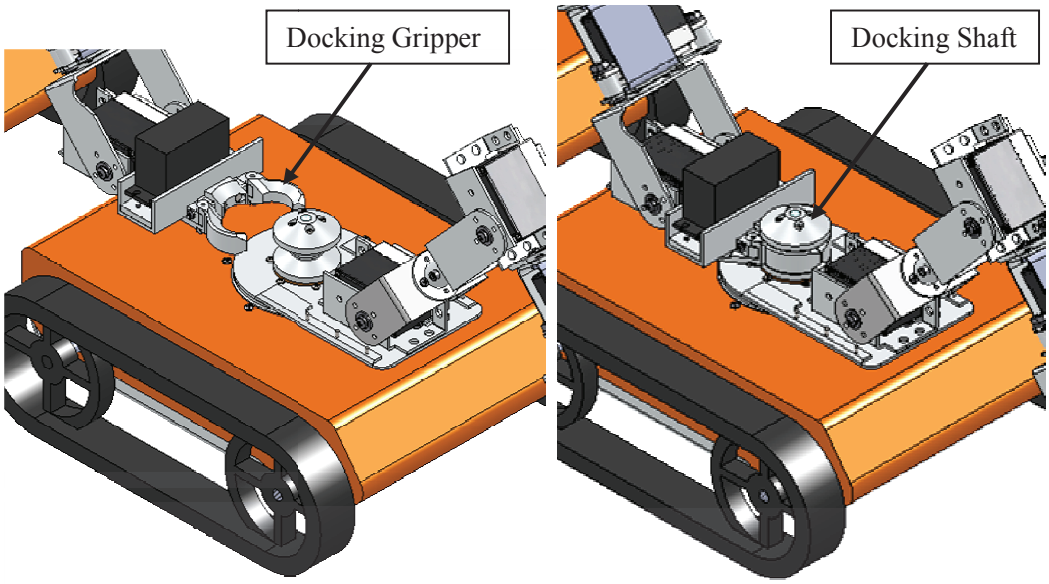

Figure 2-17: Docking System

The use of the gripper for docking allows for a simplified design where an additional docking mechanism and space for it is not needed. Furthermore, the gripper is designed such that it has two flats for other grasping purposes. The gripper is driven by a dual crank-rocker four-bar linkage, which is driven by a standard servo motor. The decision to use a servo motor to drive the gripper and a linkage is that these motors are a common component in the rest of the R2TM3 manipulator design and that a linkage can be designed to give specific performance characteristics. In this case the linkage has been designed so that its crank and coupler links approach a toggle position when the gripper is closed, creating an extremely high mechanical advantage. See 2.2.3 for linkage analysis.

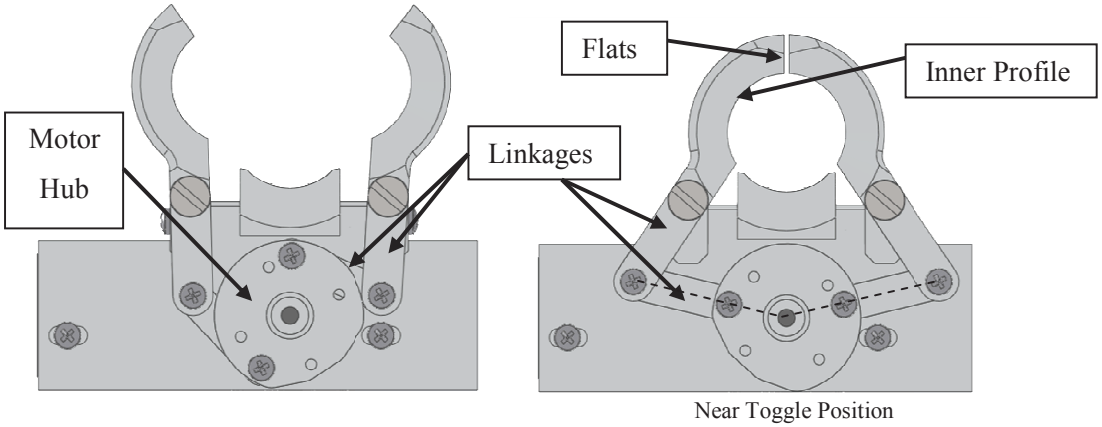

Figure 2-18: Gripper, Open/Close

In order for docking to occur a simple process must be followed. Self-alignment cannot happen if the manipulator's motors are constraining the manipulator as the gripper closes to pull into the docking position. Docking is enabled by the fact that the manipulator motors can be turned off and are back-drivable. The process is as follows:

1. Roughly locate gripper on docking shaft at groove.
2. Close gripper so that links are within groove.
3. Turn off manipulator motors to allow the gripper to seat in groove.
4. Once gripper is seated, turn manipulator motors back on.
5. Repeat process if gripper is not fully seated.

This docking process allows for a rough estimation of the correct gripper pose for docking, when the operator is controlling the manipulator via teleoperation. The self-alignment, due to the tapers and manipulator on/off control can then eliminate any pose errors, allowing the gripper to seat in the groove.

The docking system has been designed to be compact and to reduce protrusions, edges and points that the gripper or manipulator could catch on when docking. Furthermore, the gripper has been designed to open far enough, 32 mm (1.25 in), to fit over the top of the docking shaft in the case where the manipulator cannot approach the groove from the side.

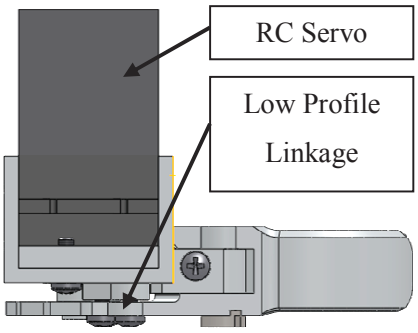

Figure 2-19: Gripper Side View

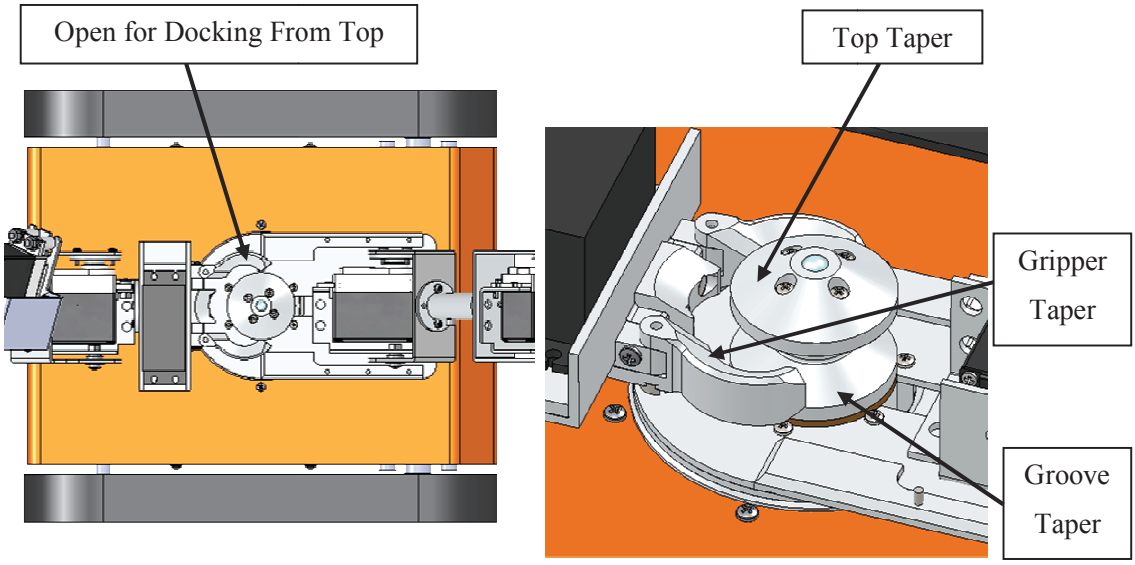

Figure 2-20: Docking, Gripper Approaching

Docking constrains the gripper to the next robot in 5-DOF, allowing only rotation in the next robot's X-Y plane. This rotation is due to the passive joint located inside the track vehicle base seen in Figure 2-21 while friction prevents the gripper from rotating inside the tapered groove. The passive joint allows the robot modules to rotate relative to one another so they may be manoeuvrable when connected, yet can cooperatively climb with the rigid connection. The forces and moments of the connection act on the tapered surfaces of the groove. Owing to the symmetrical concept, the docking joint is coincident with the base joint. Therefore, it has been integrated into the center of the base joint as shown in Figure 2-21.

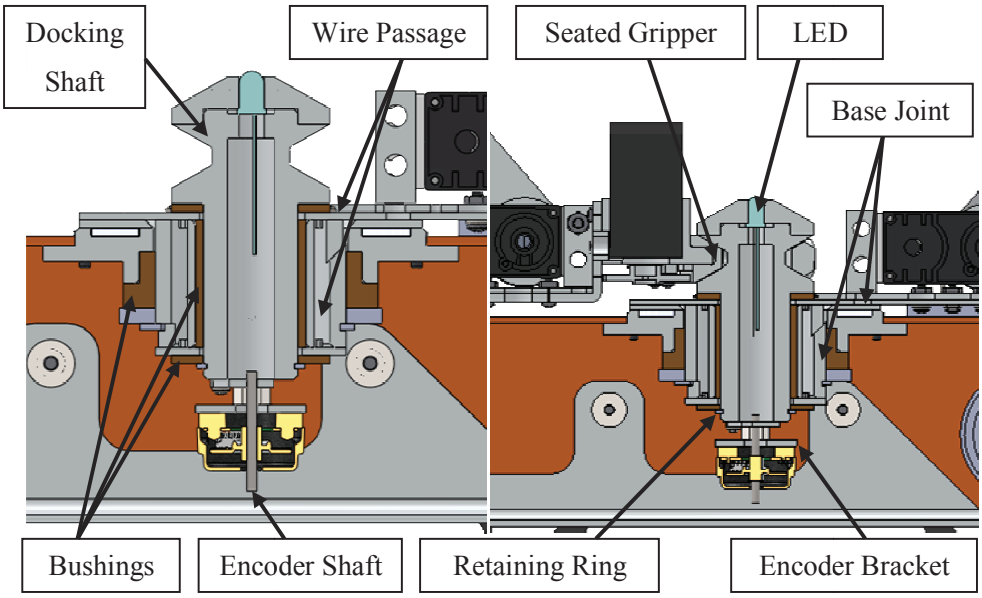

Figure 2-21: Docking Joint Integration

The docking shaft has bronze bushings to reduce friction on surfaces with relative motion, has been designed to encapsulate an LED and contains a small end shaft for docking angle measurement. Furthermore, a cavity has been designed between the base joint and docking joint so that wires may be passed between the manipulator and track base. The docking encoder is housed at the bottom of the track base and mounted with its own bracket.

The docking system, with the use of the dual gripper docking mechanism and an integrated passive docking joint coincident to the base joint, allow for flexible docking with self-aligning characteristics and enables the desired cooperative functionality. Moreover, it clamps rigidly and securely enabling significant force transfer between robots without the risk of the docking mechanism opening or the robots un-docking. This design along with the docking process makes docking possible in all types of terrain situations.

2.2.3 Modeling and Analysis

In order for the R2TM3 design to function as intended, the system needed to be modeled and analyzed. The modeling of a cooperative mobile multi-body robot system such as this is no easy task, "The development of methods to generate motion models of robotic systems is a vast area of research. Still none of the existing approaches address the peculiar issues pertaining to modeling and analysis of physically cooperating mobile robots... " [32]. In [32] the authors propose a P-robot method toward

a framework for the dynamics of such systems, in response to the lack of an existing method. In this thesis work a simplified and intuitive approach, unique to the R2TM3 configuration, is used. Considering the R2TM3's intended use, cooperation method and proposed mechanical components, a quasi-static assumption is made regarding the mathematical models used to describe the system for analysis. In USAR the robot will not likely be used at high speed since it will be in a risky environment with potentially unknown hazards. Models used for analysis and design synthesis either employ a static force model or a kinematic model depending on what is needed. The purpose of the models presented here is to describe the system behavior so that it may be developed to work as intended. Models used for control are included in Chapter 3[5].

Manipulator Kinematics:

As stated, the R2TM3 modules use a 5-DOF manipulator and in order to describe the positional relationship between the joint angles and the wrist pose, the problem of forward kinematics needs to be solved [34], [37]. Figure 2-22 shows the kinematic model; the coordinate frames attached to each link and the revolute joints in a schematic representation. Frame {0} is considered as the inertial frame, since in this model, the mobile base's DOF are ignored. The model is accompanied by a table showing the D-H parameters, following the standard convention. Finally a homogeneous transform, a [4 × 4] matrix in (2.1), is given that expresses the wrist frame pose in terms of the joint variables with respect to the manipulator's base frame {0}.

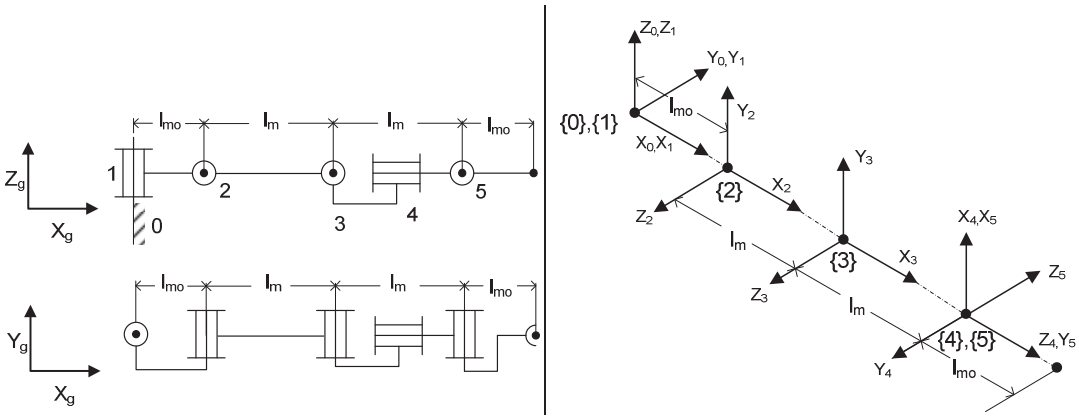

Figure 2-22: 5-DOF Manipulator Kinematic Schematic

[5] In some cases the same models are used.

Table 2-8: Manipulator D-H Parameters

D-H	Parameters			
j	α_{j-1}	a_{j-1}	d_j	θ_j
1	0°	0	0	θ_1
2	90°	l_{mo}	0	θ_2
3	0°	l_m	0	θ_3
4	90°	0	l_m	θ_4
5	90°	0	0	θ_5

$$ {}^0_5T = \begin{bmatrix} {}^0_5t_{11} & {}^0_5t_{12} & {}^0_5t_{13} & {}^0_5t_{14} \\ {}^0_5t_{21} & {}^0_5t_{22} & {}^0_5t_{23} & {}^0_5t_{24} \\ {}^0_5t_{31} & {}^0_5t_{32} & {}^0_5t_{33} & {}^0_5t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1} $$

where

$$ {}^0_5t_{11} = c_1c_2c_3c_4c_5 - c_1s_2s_3c_4c_5 + s_1s_4c_5 + c_1c_2s_3s_5 + c_1s_2c_3s_5 \tag{2.2} $$

$$ {}^0_5t_{12} = -c_1c_2c_3c_4s_5 + c_1s_2s_3c_4s_5 - s_1s_4s_5 + c_1c_2s_3c_5 + c_1s_2c_3c_5 $$

$$ {}^0_5t_{13} = c_1c_2c_3s_4 - c_1s_2s_3s_4 - s_1c_4 $$

$$ {}^0_5t_{14} = l_mc_1c_2s_3 + l_mc_1s_2c_3 + l_mc_1c_2 + l_{mo}c_1 $$

$$ {}^0_5t_{21} = s_1c_2c_3c_4c_5 - s_1s_2s_3c_4c_5 - c_1s_4c_5 + s_1c_2s_3s_5 + s_1s_2c_3s_5 $$

$$ {}^0_5t_{22} = -s_1c_2c_3c_4s_5 + s_1s_2s_3c_4s_5 + c_1s_4s_5 + s_1c_2s_3c_5 + s_1s_2c_3c_5 $$

$$ {}^0_5t_{23} = s_1c_2c_3s_4 - s_1s_2s_3s_4 + c_1c_4 $$

$$ {}^0_5t_{24} = l_ms_1c_2s_3 + l_ms_1s_2c_3 + l_ms_1c_2 + l_{mo}s_1 $$

$$ {}^0_5t_{31} = s_2c_3c_4c_5 + c_2s_3c_4c_5 + s_2s_3c_5 - c_2c_3s_5 $$

$$ {}^0_5t_{32} = -s_2c_3c_4s_5 - c_2s_3c_4s_5 + s_2s_3s_5 - c_2c_3c_5 $$

$$ {}^0_5t_{33} = s_2c_3s_4 + c_2s_3s_4 $$

$$ {}^0_5t_{34} = l_ms_2s_3 - l_mc_2c_3 + l_ms_2 $$

In Figure 2-22 l_m is the link length for links 2 and 3 and l_{mo} is the link length for links 1 and 5. In equation (2.1), excluding row 4, column 4 represents the position vector of frame $\{5\}$ with respect to frame $\{0\}$. Columns 1 through 3 contain the rotation matrix which describes the orientation of frame $\{5\}$ in frame $\{0\}$ with three direction cosines. In equation (2.2) the shorthand convention is used for trigonometric functions.

The Jacobian matrix can be used to describe differential motion such as the velocity of the wrist frame in relation to the joint variable velocities [34], [37]. Furthermore, the Jacobian transpose may be used, due to the virtual work principle, to solve joint torques given the forces acting on the wrist [34], [37]. In equation (2.3) $\dot{q}$ is the joint velocity vector, ${}^0_5J(q)$ is the Jacobian matrix as a function q, the joint position vector, and $\dot{X}$ the wrist velocity vector. Equation (2.4) gives the manipulator's Jacobian matrix in the explicit form. Rows 1 through 3 represent the relation of the joint velocities and the linear velocities of the wrist and rows 4 through 6 represent the relationship of joint velocities to the angular velocities of the wrist.

$$\dot{X} = {}^0_5J(q)\dot{q} \tag{2.3}$$

$${}^0_5J = \begin{bmatrix} {}^0_5j_{11} & {}^0_5j_{12} & {}^0_5j_{13} & 0 & 0 \\ {}^0_5j_{21} & {}^0_5j_{22} & {}^0_5j_{23} & 0 & 0 \\ 0 & {}^0_5j_{32} & {}^0_5j_{33} & {}^0_5j_{44} & {}^0_5j_{45} \\ 0 & s_1 & s_1 & {}^0_5j_{54} & {}^0_5j_{55} \\ 0 & -c_1 & -c_1 & {}^0_5j_{64} & {}^0_5t_{65} \\ 1 & 0 & 0 & & \end{bmatrix} \tag{2.4}$$

where

$${}^0_5j_{11} = -l_m s_1 c_2 s_3 - l_m s_1 s_2 c_3 - l_m s_1 c_2 - l_{mo} s_1 \tag{2.5}$$

$${}^0_5j_{12} = -l_m c_1 s_2 s_3 + l_m c_1 c_2 c_3 - l_m c_1 s_2$$

$${}^0_5j_{13} = l_m c_1 c_2 c_3 - l_m c_1 s_2 s_3$$

$${}^0_5j_{21} = l_m c_1 c_2 s_3 + l_m c_1 s_2 c_3 + l_m c_1 c_2 + l_{mo} c_1$$

$${}^0_5j_{22} = -l_m s_1 s_2 s_3 + l_m s_1 c_2 c_3 - l_m s_1 s_2$$

$${}^0_5j_{23} = l_m s_1 c_2 c_3 - l_m s_1 s_2 s_3$$

$$ {}^{0}_{5}j_{32} = l_m c_2 s_3 + l_m s_2 c_3 + l_m c_2 $$

$$ {}^{0}_{5}j_{33} = l_m s_2 c_3 + l_m c_2 s_3 $$

$$ {}^{0}_{5}j_{44} = c_1 c_2 s_3 + c_1 s_2 c_3 $$

$$ {}^{0}_{5}j_{45} = c_1 c_2 s_4 - c_1 s_2 s_3 s_4 - s_1 c_4 $$

$$ {}^{0}_{5}j_{54} = s_1 c_2 s_3 + s_1 s_2 c_3 $$

$$ {}^{0}_{5}j_{55} = s_1 c_2 c_3 s_4 - s_1 c_2 s_3 s_4 + c_1 c_4 $$

$$ {}^{0}_{5}j_{64} = s_1 s_3 - c_2 c_3 $$

$$ {}^{0}_{5}j_{65} = s_2 c_3 s_4 + c_2 s_3 s_4 $$

for joint torques

$$ \tau = {}^{0}_{5}J(q)^T f \tag{2.6} $$

Where τ is the generalized joint torque vector and f is the generalized force vector acting on the wrist point [34], [37].

Reduced Order Manipulator Kinematics:

In some cases the analysis for certain robot functions may be done in reduced order model. For example the cooperative climbing may be analyzed in 2D in the X-Z plane, leading to a 3-DOF manipulator model. Therefore, it is useful to find a homogeneous transformation for the forward kinematics where joints 1 and 4 are no longer variables (taken to be zero). The transformation for forward kinematics then becomes a $[3 \times 3]$ matrix. Subscript r is used to denote models as reduced order.

$$ {}^{0}_{5}T_r = \begin{bmatrix} {}^{0}_{5}t_{11_r} & {}^{0}_{5}t_{12_r} & {}^{0}_{5}t_{13_r} \\ {}^{0}_{5}t_{21_r} & {}^{0}_{5}t_{22_r} & {}^{0}_{5}t_{23_r} \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7} $$

where

$$ {}^{0}_{5}t_{11_r} = c_2c_3c_5 - s_2s_3c_5 + c_2s_3s_5 + s_2c_3s_5 \tag{2.8} $$

$$ {}^{0}_{5}t_{12_r} = -c_2c_3s_5 + s_2s_3s_5 + c_2s_3c_5 + s_2c_3c_5 $$

$$ {}^{0}_{5}t_{13_r} = l_mc_2s_3 + l_ms_2c_3 + l_mc_2 + l_{mo} $$

$$ {}^{0}_{5}t_{21_r} = s_2c_3c_5 + c_2s_3c_5 + s_2s_3c_5 - c_2c_3s_5 $$

$$ {}^{0}_{5}t_{22_r} = -s_2c_3s_5 - c_2s_3s_5 + s_2s_3s_5 - c_2c_3c_5 $$

$$ {}^{0}_{5}t_{23_r} = l_ms_2s_3 - l_mc_2c_3 + l_ms_2 $$

the reduced order Jacobian becomes

$$ {}^{0}_{5}J_r = \begin{bmatrix} {}^{0}_{5}j_{11_r} & {}^{0}_{5}j_{12_r} & 0 \\ {}^{0}_{5}j_{21_r} & {}^{0}_{5}j_{22_r} & 0 \\ -1 & -1 & 1 \end{bmatrix} \tag{2.9} $$

where

$$ {}^{0}_{5}j_{11_r} = -l_ms_2s_3 + l_mc_2c_3 - l_ms_2 \tag{2.10} $$

$$ {}^{0}_{5}j_{12_r} = l_mc_2c_3 - l_ms_2s_3 $$

$$ {}^{0}_{5}j_{21_r} = l_mc_2s_3 + l_ms_2c_3 + l_mc_2 $$

$$ {}^{0}_{5}j_{22_r} = l_ms_2c_3 + l_mc_2s_3 $$

With this reduced order model of the manipulator, the positional inverse kinematics can be solved geometrically [34], [37], where they are decoupled from the orientation inverse kinematics (involves only 2-DOF).

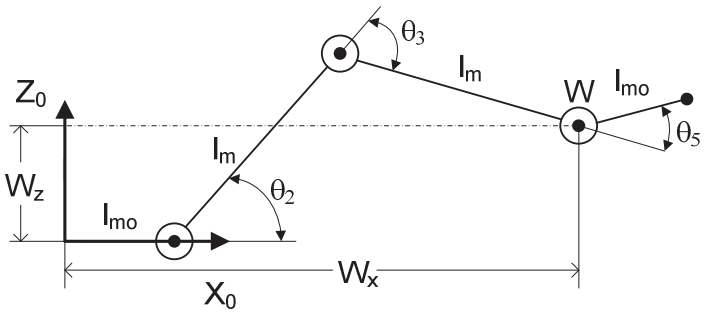

Figure 2-23: Reduced Order Manipulator Model, Inverse Kinematics

$$\cos\theta_3 = \frac{(W_x - l_{mo})^2 + {W_z}^2 - 2{l_m}^2}{2{l_m}^2} \tag{2.11}$$

$$\cos\theta_3 = \frac{{W_x}^2 + {W_z}^2 + {l_{mo}}^2 - 2{l_m}^2 - 2l_{mo}W_x}{2{l_m}^2} := D$$

for an elbow up configuration

$$\theta_3 = atan2\left(D, -\sqrt{1 - D^2}\right) \tag{2.12}$$

$$\theta_2 = atan2(W_x - l_{mo}, W_z) - atan2(l_m + l_m\cos\theta_3, l_m\sin\theta_3) \tag{2.13}$$

In Figure 2-23 it can be seen that W_x and W_z are the wrist point positions. The wrist orientation θ_5 is solved in subsequent models as needed by a relationship between the connected robot modules.

Fixed Base Pitching Model:

Critical to the function of R2TM3 is the ability to pitch a given robot's end off the ground for climbing as seen in Figure 2-2 and Figure 2-24. Therefore, the required joint torques of the manipulator and joint ranges must be known so that the manipulator may work as intended. Analysis was done on a model of the connected chain of three robot modules, in 2D (X-Z plane), for a variety of ground plane slopes and robot pitch angles. Although this analysis gave insight into the system behavior, the yielded results became difficult to interpret. This led to a simplified modeling approach referred to as the Fixed Base Pitching Model and the Fixed Base Reverse Pitching Model. The modeling assumptions for these models are as follows:

1. The front robot (Robot 1) in the connected chain requires the highest manipulator forces for pitching, since only the second robot's manipulator participates it the pitching. Other robots throughout the connected chain may have two manipulators participating in the pitching motion.

2. The rear robot (Robot 3) in the connected chain requires the highest manipulator forces for reverse pitching[6], since only its own manipulator participates in the pitching.
3. When pitching a given robot off the ground the neighboring robots in the chain need not pitch off the ground as well.
4. For pitching up Robot 1, any reaction forces and moments caused between Robot 2 and the ground can be reacted by friction and its tracks, Robot 2's weight and the rigid connection of Robot 3 acting as a counter weight to prevent Robot 2 from rotating.
5. For reverse pitching of Robot 3, the same assumption is made as in point 4, but with respect to Robot 2 and Robot 1 acting as the counter weight.
6. Consequently, for both pitching and reverse pitching a single robot module may be considered to quasi-statically pitch, while the other end of the participating manipulator is attached to a single rigid body fixed to the ground.

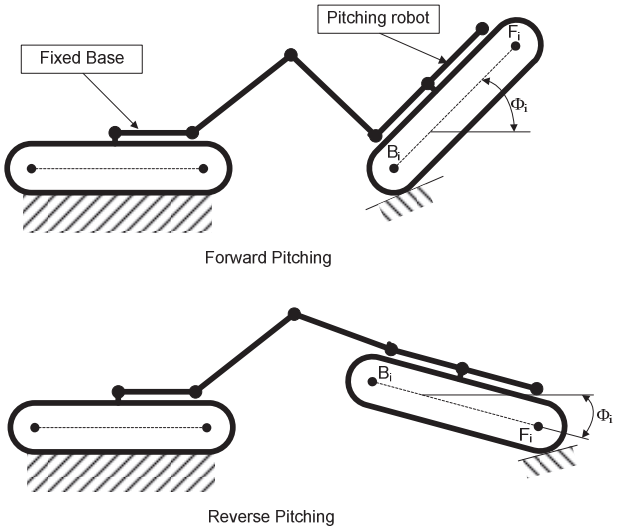

Figure 2-24: Pitching and Reverse Pitching Definition

With these assumptions, this simplified model allows analysis in which the results can be interpreted. Critical to this approach is the assumption that the tracks will not simply turn, when the robot is subject to a force parallel the ground plane. The internal resistance of the tracks and the use

[6] This is also referred to as self pitching, due to the manipulator symmetry

of a track speed controller validate this assumption. This model takes the effect of the connected robots, not being pitched, and lumps them into one single body fastened to the ground.

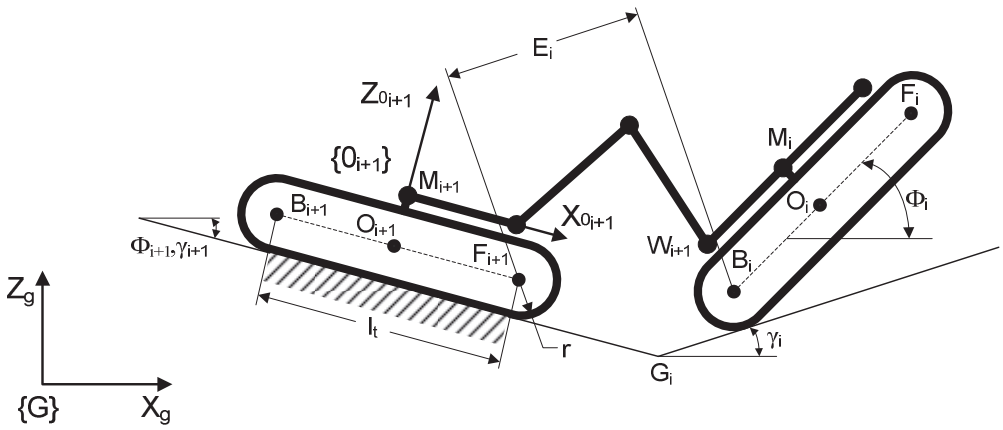

Figure 2-25: Fixed Base Model

The Fixed Base Model is simulated[7] by placing two ground planes, represented in 2D by line segments of arbitrary length, at an arbitrary distance away from the origin of the inertial (global) frame of reference $\{G\}$. Each ground plane's slope angle γ can be modified as a parameter. The pitching robot is denoted with subscript i. The fixed base robot $(i+1)$ is placed on ground segment $(i+1)$ at a reasonably small and arbitrary fixed distance away from the ground transition point G_i. The pitching robot is placed on the next ground segment at a variable distance E_i away from the fixed robot, where E_i is defined as the distance between B_i and F_{i+1} parallel to ground plane i. The model is simulated by varying E_i and ϕ_i where robot i pitches about point B_i. The positional simulation is helpful for finding the required joint angles to achieve a variety of configurations for different ground slope conditions.

Through geometric relationships the positions of various points of interest can be solved with respect to the global coordinate frame $\{G\}$. Most significantly, the distances $W_{x_{i+1}}$, $W_{z_{i+1}}$ and $M_{x_{i+1}}$, M_{i+1} are solved with respect to the global coordinate frame. Using the difference between W_{i+1} and M_{i+1} and a rotation matrix the wrist point position can be described in the manipulator's base frame of reference $\{0_{i+1}\}$.

[7] Simulation refers to the use of programmed spreadsheet

$$\begin{bmatrix} {}^{0_{i+1}}W_{x_{i+1}} \\ {}^{0_{i+1}}W_{z_{i+1}} \end{bmatrix} = \begin{bmatrix} \cos\phi_{i+1} & \sin\phi_{i+1} \\ -\sin\phi_{i+1} & \cos\phi_{i+1} \end{bmatrix} \begin{bmatrix} W_{x_{i+1}} - M_{x_{i+1}} \\ W_{z_{i+1}} - M_{z_{i+1}} \end{bmatrix} \tag{2.14}$$

Inverse kinematics from equations (2.11) to (2.13) can then be used to find $\theta_{2_{i+1}}$ and $\theta_{3_{i+1}}$. Where $\theta_{5_{i+1}}$ can be solved knowing the constraint between the robot pitch angles and manipulator joint angles.

$$\theta_{2_{i+1}} + \theta_{3_{i+1}} + \theta_{5_{i+1}} = \phi_i - \phi_{i+1} \tag{2.15}$$

$$\theta_{5_{i+1}} = \phi_i - \phi_{i+1} - \theta_{2_{i+1}} - \theta_{3_{i+1}}$$

Consequently, various conditions can be analyzed and the required joint angles can be figured out. Judgment has be exercised when strange, and likely impractical, configurations require large joint angle ranges. The determined joint angle ranges can be seen in Figure 2-11.

As mentioned previously, the pitching function is cooperative between the tracks and the manipulator. To simulate the pitching robot's tracks contributing, a maximum coefficient of static friction is used as a parameter that serves to limit the track torque that can be applied without slipping. Furthermore, a track force multiplier variable is used to change the applied track force from 100% through to -100%. This models the effect of having a range of track forces used to assist the manipulator in pitching, with the aim of analyzing out how much actual track torque is needed[8]. The manipulator and tracks together must create a moment to overcome the moment due to the robot's weight. In this model the weight of the robot's manipulator is ignored and the COM for the weight to be pitched acts at the geometric center of the robot base.

A free body diagram (FBD) for robot i can be done as a starting point to solve the required joint torques.

[8] The track force is due to the torque produced by both track drive motors

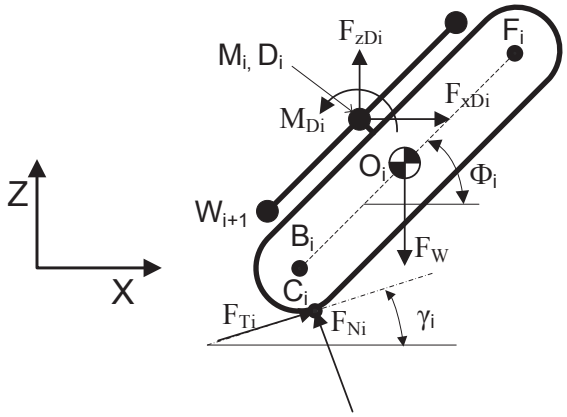

Figure 2-26: FBD Robot i, Pitching

The system in Figure 2-26 is statically indeterminate with four unknowns; the manipulator forces F_{D_i} (in x and z), manipulator moment M_{D_i} and the normal force F_{N_i}. Making the simplifying assumption, or rule, that the manipulator forces should not have an effect on the normal force, it can be solved as:

$$\left|F_{N_i}\right| = |F_W \cos\gamma_i| \tag{2.16}$$

$$F_{N_i} = \begin{bmatrix} F_{Nx_i} \\ F_{Nz_i} \end{bmatrix} = \left|F_{N_i}\right| \begin{bmatrix} -\sin\gamma_i \\ \cos\gamma_i \end{bmatrix}$$

Where F_W is the module weight. The applied track force is given as:

$$F_{T_i} = -\frac{\tau_i(tfm)}{r} \angle\gamma_i \tag{2.17}$$

$$F_{T_i} = \begin{bmatrix} F_{Tx_i} \\ F_{Tz_i} \end{bmatrix} = -\frac{\tau_i(tfm)}{r} \begin{bmatrix} \sin\gamma_i \\ \cos\gamma_i \end{bmatrix}$$

Where tfm is the track force multiplier, r is the track radius and τ_i is the combined applied track torque from both track drive motors. By the chosen sign convention, the minus sign is needed to related track torque to the force acting on the robot. The static equations can then be solved as:

$$\sum M_{D_i} \circlearrowleft_+ = 0 \tag{2.18}$$

$$M_{D_i} = -F_{Tx_i}(D_{z_i} - C_{z_i}) + F_{Tz_i}(D_{x_i} - C_{x_i}) - F_W(D_{x_i} - O_{x_i}) - F_{Nx_i}(D_{z_i} - C_{z_i}) + F_{Nz_i}(D_{x_i} - C_{x_i})$$

$$\sum F_x \rightarrow_+ = 0 \tag{2.19}$$

$$F_{Dx_i} = -F_{Tx_i} - F_{Nx_i}$$

$$\sum F_z \uparrow_+ = 0 \tag{2.20}$$

$$F_{Dz_i} = -F_{Tz_i} - F_{Nz_i} + F_W$$

By taking moments about point D_i, the manipulator forces drop out of the moment equation. C_i is the contact point found from simple geometry in the positional analysis.

The forces and moment acting on M_i (also point D_i) may be translated to the wrist point W_{i+1} of the connected manipulator with equation (2.21), including a sign reversal to describe the loading as acting on manipulator $i + 1$:

$$M_{W\,i+1} = -M_{D_i} - F_{zD_i}(D_{x_i} - W_{x_{i+1}}) + F_{xD_i}(D_{z_i} - W_{z_{i+1}}) \tag{2.21}$$

$$F_{Wx_{i+1}} = -F_{xD_i}$$

$$F_{Wz_{i+1}} = -F_{zD_i}$$

The forces acting on the wrist point of manipulator $i + 1$ are described in the global coordinate frame and can be described in the manipulator base frame using the same rotation matrix from equation (2.14).

Finally, motor torques may be solved utilizing the reduced order Jacobian transpose and changing signs to describe applied motor torques:

$$\tau_{i+1} = {}^{0}_{5}J^{T}{}_{r} \begin{bmatrix} -F_{Wx_{i+1}} \\ -F_{Wz_{i+1}} \\ M_{W_{i+1}} \end{bmatrix} \tag{2.22}$$

$$\tau_{i+1} = \begin{bmatrix} \tau_{2_{i+1}} \\ \tau_{3_{i+1}} \\ \tau_{5_{i+1}} \end{bmatrix}$$

Where $\tau_{1_{i+1}}$ and $\tau_{4_{i+1}}$ are assumed as zero.

The simulation is helpful for solving what amount of torque is needed in the three manipulator joints that participate in the pitching motion, for a given track torque. From the opposite perspective, it helps determine what amount of applied track torque is needed to help the manipulator perform the pitching. What is revealed from the analysis is that the amount of help the manipulator needs from the tracks depends largely on its configuration, the ground slopes and the robot's pitch angle. In some cases, with the selected manipulator motors, the manipulator does not need any assistance, while in other cases a given joint in the manipulator requires torque that exceeds the motor torque limit even with the help of the tracks. Ultimately, when the manipulator is in a position of good mechanical advantage, it does not need to rely on much help from the tracks. In some cases, if the track torque is too high it can actually load the manipulator in the opposite direction. This simulation helped to show what the reasonable expectations should be for the cooperative approach; not all configurations are viable. It allowed for the proper selection of manipulator motors base on an appropriate amount of applied track torque assisting the manipulator. Some key insights are as follows:

- The applied track force needed for the pitching motion, which relies on the actual coefficient of friction of the ground, is not excessive. This means that exceptional traction is not required to pitch the robot. Therefore, unmodeled changes in normal force due to the connected manipulator will not be problematic during pitching.
- The cooperative pitching approach is viable in most manipulator configurations. In cases where the pitch angle is small (moment due to weight is large) and the manipulator is tucked in, the pitching may not be possible. The closer the arm is to full extension (singularity) the better it can pitch, in terms of needed joint torques.

The joint motors were selected on the judgment that excessive joint torque output is not needed, where the few cases that require it for pitching are rare, obvious and not of practical importance.

In the case of reverse or self pitching, the symmetry in the design allows this to be analyzed the same way, where the resolved joint torques are in reverse order. Since the same motor is selected for each joint, this makes no difference. Therefore, reverse pitching can be simulated by changing the pitch angle of robot i to be less than the ground plane angle and to consider the pitching to occur about point F_i where point C_i exists under the front on the track base. Following the same procedure as outlined the reverse pitching yields similar results to forward pitching.

Manipulator Base Joint Model:

A simple, yet critical model is needed to determine the requirements for the manipulator's base joint. From the cooperative shifting maneuver, it can be seen in Figure 2-4 that the minimum angular range for the base joint is -90° to +90°. A larger range would be useful, but given the 180° range limit of RC servos, this is considered sufficient.

The model of the base joint is then created to find the torque requirements. The approach used is to consider the mobile base sitting on a side-slope in which the manipulator's base joint must move with the fully extended manipulator holding a small pay load.

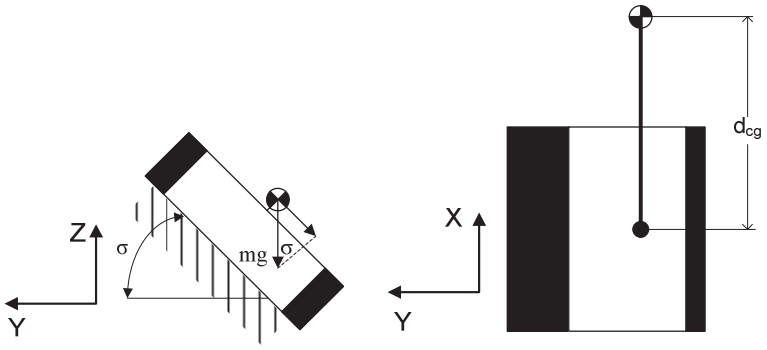

Figure 2-27: Base Joint Torque Requirement Model

$$\tau_1 = mg \sin(\sigma)\, d_{cg} \tag{2.23}$$

Where mg is the combined weight of the manipulator and payload of approximately 50% of the manipulator weight. The distance d_{cg} is to the combined COM for the manipulator and payload, found using CAD software. The side-slope angle σ is taken at a maximum of 45°. The required torque fell in a range similar to the torque requirements for the joints used in cooperative climbing. Thus, it was decided that all 5-DOF would use the same motor. The roll joint (joint 4) motor is equipped with

the same motor based on the judgment that it should have plenty of torque with this motor type, and any extra cost is worth the confidence of this joint not having torque limitations during testing.

Docking Linkage Analysis:

The docking mechanism, as stated, is a dual crank-rocker type four-bar linkage. Therefore, in order for this linkage to give the desired gripper range and output forces, the linkage needed to be analyzed.

A four-bar linkage may be represented using a positional vector loop [38], using complex vector notation. In this context the angles θ_2, θ_3 and θ_4 are the link vector angles, not manipulator joint angles. In Figure 2-28 it can be seen that θ_2 is actually the gripper motor angle and that link c is part of the gripper link.

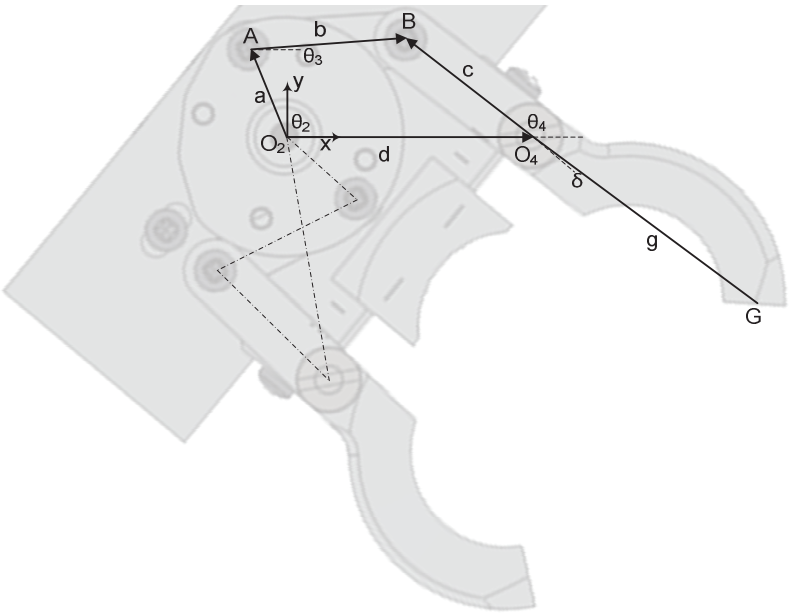

Figure 2-28: Gripper/Docking Mechanism as Vector Loop

Similar to the forward kinematic problem for the manipulator the goal is to find a relationship between the motor angle and the point of interest on the gripper G. Point G is chosen to be at the midpoint of the gripper flat to emulate an object opposing the gripper motor or a prying action on the gripper when the robots are docked. In this case, the linkage is a closed chain that allows this point to be described in terms of only the motor angle and link lengths. Following a standard analysis for a four bar linkage [38] the linkage positions and link angles may be solved. With this known, the position of the point of interest may be solved as:

$$G_x = g\cos(\theta_4 + \delta + \pi) + d \tag{2.24}$$

$$G_y = g\sin(\theta_4 + \delta + \pi)$$

Where θ_4 is a function of θ_2 and the link lengths; see [38] for more four-bar linkage analysis details.

The linkage design synthesis was performed by simulating the linkage motion with an unconstrained CAD sketch and a spreadsheet. However, simply having a linkage configuration that gives the required range of motion for the gripper is not enough. The forces through the linkage need to be optimized, so that the maximum output force occurs when the gripper is closed, so that the robots may remain docked with little output torque from the motor. From a force analysis equation (2.25) was derived that gives the magnitude of the output gripper force.

$$F_g = -\left(\frac{\tau_M}{g\sin\left(\delta + \frac{\pi}{2}\right)}\right)\left(\frac{c\sin\left(\theta_3 - \theta_4 + \frac{\pi}{2}\right)}{a\sin(\theta_2 - \theta_3)}\right) \tag{2.25}$$

Where τ_M is the motor torque and the gripper force F_g is assumed to be normal to length g seen in Figure 2-28.

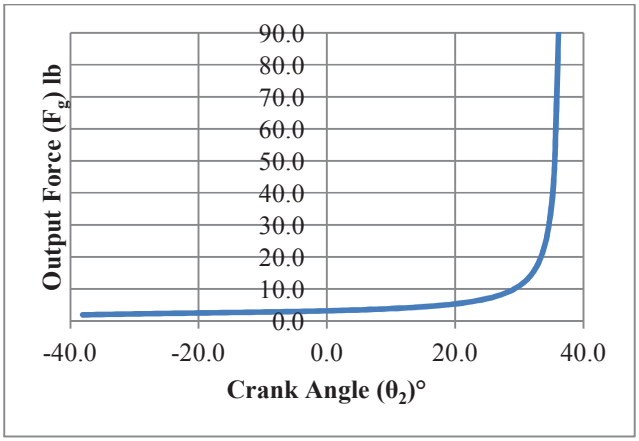

Figure 2-29: Gripper Output Force vs Crank Angle

In Figure 2-29 it is obvious that, for a given motor torque, the output force approaches infinity as the linkage crank (a) and coupler (b) approach a toggle position ($\theta_2 = \theta_3$). Therefore, the linkage is locked at or past this position and the motor does not need to output any torque. The linkage was actually configured to close the gripper just before the toggle point to avoid unwanted locking during

prototyping. The motor then requires a small amount of torque and therefore power to keep the gripper closed.

The gripper itself is under rather high loads compared to the rest of the modular robot system. It is also designed to be compact, so that it remains light on the end of the manipulator and avoids catching during docking. Consequently, the strength of the main gripper link was of concern. This component was modeled statically in a finite element analysis (FEA) simulation, where applied loads were found from the fixed base pitching analysis. In this model, the loads were simplified and applied to exaggerate the actual loading that would occur in both magnitudes and way they were applied. This coarse modeling approach showed that even with the exaggerated loading, the design yielded a generous factor of safety (FOS), greater than 3, for a common material type (6061 aluminum).

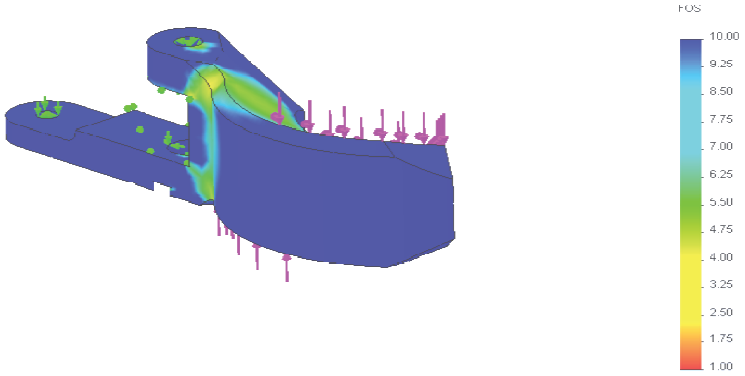

Figure 2-30: Gripper Link FOS Plot

From the previously described models it can be seen that in order to achieve the desired mechanical performance the system should be modeled to suit the design and analysis needs. Simplifying the system model by making a quasi-static assumption can allow for meaningful information to be more easily extracted for design.

2.3 Control Hardware Architecture

The R2TM3 control hardware architecture is distributed and modular in the sense that each robot has its own control hardware which communicates with the rest of the network wirelessly. This eliminates the need for wired connections, when docked, between modules to share data. It also provides system redundancy since each module has the same hardware functionality and architecture. Furthermore, the

hardware architecture within each module is distributed, allowing for the computational burden to be shared among components. A simple PC based operator control unit (OCU) is also used for high level computation and to act as a master node in the network.

2.3.1 Control Hardware

Given the low cost approach to module design it was decided to use a microcontroller as the main module controller. Most microcontrollers intended for use in low cost robotics designs are 8 bit, which could pose challenges for the computational requirements for cooperative control schemes. Furthermore, to reduce implementation time it was desired to select a controller with an available and user friendly integrated development environment (IDE). A common microcontroller platform that satisfies this need is Arduino [39]. These microcontrollers are open source, have a massive online community and provide a simple IDE with plenty sample code. However, they are only 8 bit and may not be able to handle the computational requirements of the R2TM3 project. Fortunately, another more powerful platform exists which is 32 bit, is around the same cost, has a very similar IDE and also has an online community. Therefore, the Leaflabs Maple 32 bit microcontroller has been selected as the main controller for the robot modules. The Maple has a 72MHz ARM Cortex M3 CPU, plenty of digital IO, many analog inputs and three hardware serial ports, see Appendix A for more details. This configuration with the enhanced computational resources and similar cost to 8 bit controllers make it an ideal choice for the R2TM3 prototype controller.

The 5-DOF manipulator contains six RC servos, including the gripper that need to be controlled with the appropriate signals discussed in 2.3.3. Controlling this many servos becomes a computationally intensive task. Furthermore, feeding wires between the track vehicle base and manipulator becomes a challenge given the configuration of the base and docking joints. For these reasons a dedicated Pololu serial servo controller (SSC) was chosen to handle the manipulator servo signals, excluding the base joint motor. This alleviates some of the load on the main controller and provides some controller distribution (robustness); if the servo controller goes down the tracks will still work. This controller can also turn servo motors on and off and can ramp servo position commands, giving the effect of some speed control.

The two DC motors used for the tracks require motor drivers that can handle approximately 1.5A. The DFRobot dual motor controller was selected since its intended for use with Arduino type controllers, can handle the current and can control both track motors.

The control architecture for one robot module can be seen in Figure 2-31. The thicker lines indicate serial communication, while the thinner lines indicate direct signals. All motor feedback and sensor signals are linked to the main controller.

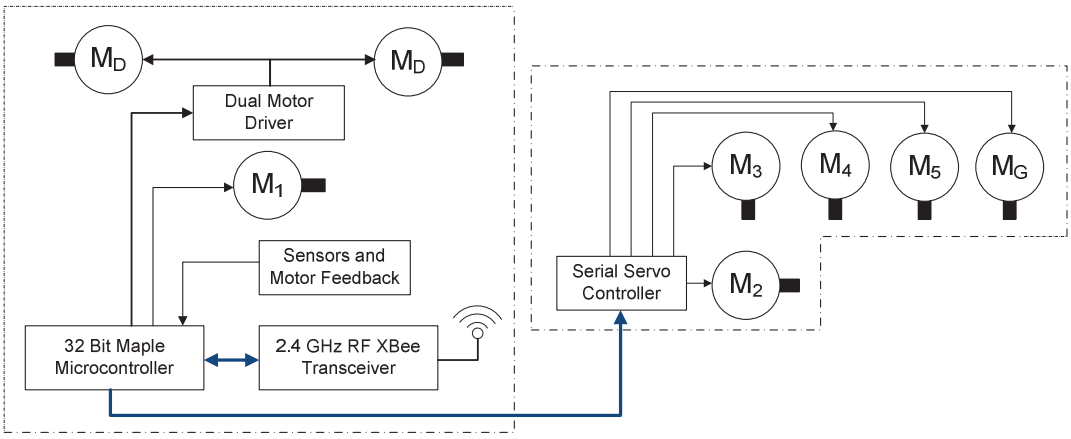

Figure 2-31: Control Architecture Diagram, Single Module

It can also be seen that a serial port on the main controller is connected to a wireless transceiver. This transceiver is used to link the module to the wireless network described next.

2.3.2 Communication

The universal synchronous asynchronous receiver/transmitter (USART) ports on the microcontroller are used for serial communication with the SSC and transceiver. This enables the modular and distributed control architecture mentioned. The specifics on this communication will be outlined here.

The wireless network between the robot modules and operator PC is facilitated by XBee series 1transceivers. XBee was selected for its ease of use, low power consumption, low cost and sufficient bandwidth. The XBee transceivers serve to replace the wires that would be needed on the respective USART. The network topology enabled with this hardware is a point-to-multipoint (PMP) type, in which the PC's node is the master or coordinator node on the personal area network (PAN). The transceivers are configured such that one is set to act as the coordinator node and to have a non-default PAN ID. The wireless communication is set up for full duplex, but the robot nodes cannot communicate directly with each other; they must communicate with the PC, then the PC can pass the signal along. Therefore this network employs polling to manage network signals [40].

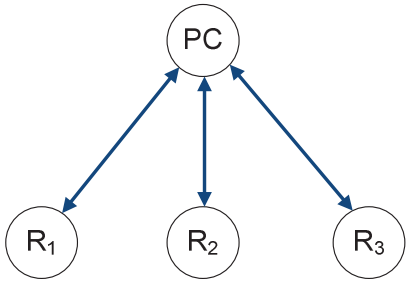

Figure 2-32: Network Topology

Having the PC act as the master node provides a means for teleoperation of the modules and also provides the opportunity for offline computing [40]. Since it's difficult to predict computational requirements when code is implemented, the option for offline computing is attractive for prototyping. In this case, general commands can be pre-computed on the PC so that more meaningful or direct commands can be sent to the robot modules. This offline computing is actually avoided as much as possible in order to keep the control distributed, but it gives some extra buffer for prototyping if need be.

The serial communication between the microcontroller and SSC is half duplex. The SSC only receives commands from the microcontroller and sends the appropriate commands to the respective servos. The motor feedback is wired directly to the main controller, since RC servos do not usually provide external feedback and need to be modified to do so.

Making use of serial communication and wireless communication has enabled a system with greater computational capabilities, more control distribution and greater convenience due to the lack of a tether to the robots.

2.3.3 Sensors and Actuators

The actuators have been described previously, but in a mechanical context. From a control hardware standpoint there are two types of motors in the R2TM3 system; the RC servos and the track drive DC gear motors. For both of these types feedback is required for control, but the type of feedback and how it's measured is different. Furthermore, there is other proprioceptive sensory within each robot module to provide feedback on the passive docking joint as well as other values. The details on how the actuators are controlled and how feedback is measured are now described.

RC servos contain their own internal potentiometer position feedback and motor driver for controlling their position. They are three wire devices and receive a pulse with modulation (PWM) signal of a certain pulse width in microseconds that corresponds to a certain angular position command. To do this they use the PPM (pulse position modulation) language. Figure 2-33 shows the typical nominal relationship between servo position and pulse width.

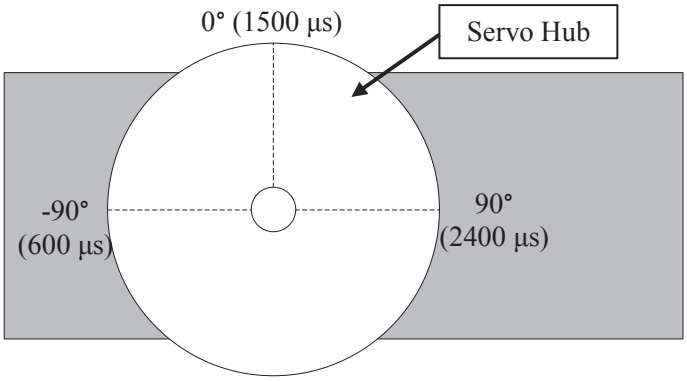

Figure 2-33: Servo Positions

Most controllers built for use with servos can readily be used to send an appropriate pulse width command such as the SSC and the microcontroller via the servo library. This makes servos very convenient to control with standard hardware of this type. A major drawback though, is that the actual motor position is not provided for external measurement. For a system like the R2TM3 this poses a problem since external feedback, from the low level controller, is needed for forward kinematics or cooperative control feedback. However, it is quite simple to modify an RC servo and solder a wire to its potentiometer wiper. This provides an analog voltage signal that measures the motor position that can go to the main controller. The linear function between voltage and angular position can be found experimentally. The motors at manipulator joints 1, 2, 3, 4 and 5 have been modified for external feedback in this fashion.

The track drive motors are controlled with the dual motor driver. The needed signals for this controller are a standard PWM signal for each motor and a digital direction signal for each. The PWM modulation ratio varies the supplied voltage to the respective motor, while the direction signal changes the polarity of the supplied voltage. With feedback measurement, speed control is possible. The built-in optical quadrature encoders are used for speed and direction measurement of each track

motor, enabling closed-loop track speed control. The encoder's 4x transitions per revolution (TPR) is 4992.

Other proprioceptive sensors in the R2TM3 modules include:

- Optical quadrature encoder for passive docking joint for angular displacement: TPR = 400.
- Two axis accelerometer (tilt sensor): provides two analog voltage signals.
- Current sensor for manipulator (excluding base joint): provides analog voltage signal.
- Current sensor for base joint: provides analog voltage signal.

The docking encoder and tilt sensor are included for feedback for cooperative control. The current sensors are added to provide a means of handling kinematic redundancy for control[9]. The added sensors used in the system are proprioceptive since the R2TM3 is teleoperated and only has what is needed for testing with respect to the primary research objective. The operator is intended to be near the system when testing, so that additional exteroceptive sensory is not needed.

Sensors that have not been added to the system either due to resource limitations or lack of an immediate need are:

- Ground contact switches: can provide knowledge of robot's interaction with the ground and some basic mapping.
- Initial docking position: would provide initial angular position of a connected manipulator with respect to the docking joint. The incremental encoder then can measure displacement.
- Digital camera: for end of manipulator situation awareness.
- IR range finders.

With the added sensory to the R2TM3 system, it is possible to achieve the cooperative function objectives. Figure 2-34 shows the location of included sensors.

[9] The on/off control of the manipulator motors can also be used for this.

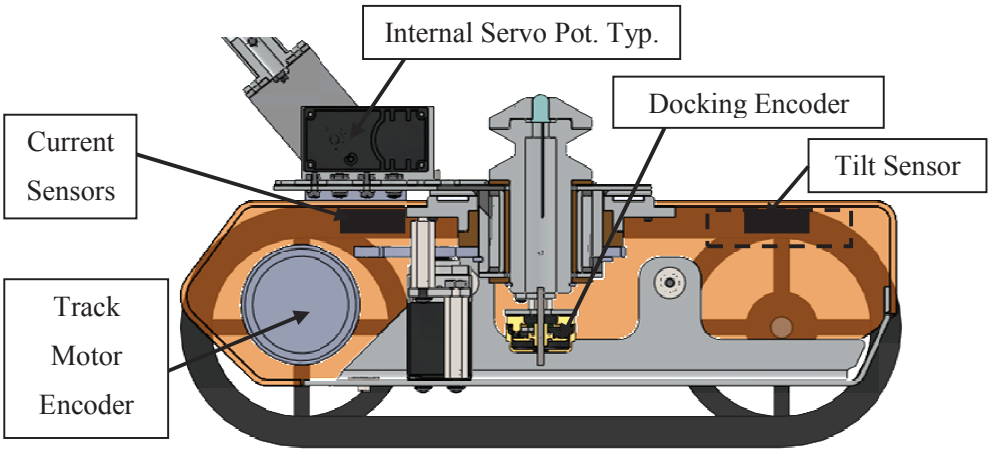

Figure 2-34: System Sensors

2.4 Electrical Power System

The electrical power system consists of four voltage levels from two separate power sources. The 7 V and 6 V actuator power comes from a 9.6 V NiMH (2600 mAh) battery pack, while the 5 V and 3.3 V controller and sensor power comes from a 7.2 V NiMH (800 mAh) battery pack. NiMH batteries were chosen due to their adequate performance and low cost. The extra size and weight of this battery type, over LiPo, is not problematic since they are housed at the bottom of the track base and opposite to the manipulator, keeping the COM low and centred.

The separation of power sources between actuators and electronics is to help protect and isolate the controllers from high voltage levels during prototyping. This also prevents the controllers from shutting off if the actuators have drained the battery or if an actuator causes a power surge that drops the supply voltage. The electrical system schematic for one robot module can be seen in Figure 2-35.

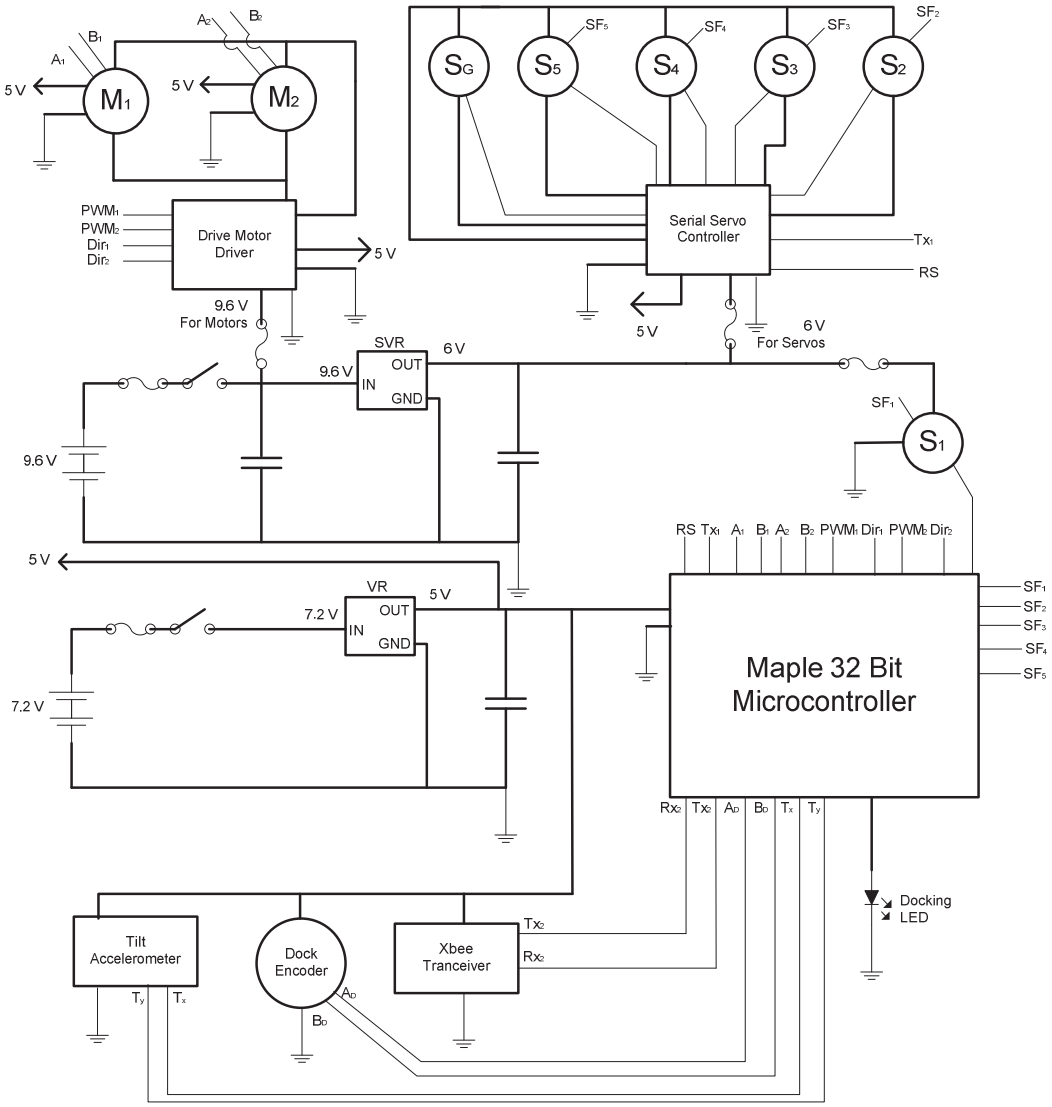

Figure 2-35: Robot Module Electrical Schematic

Bypass capacitors are used to help with power surges in the actuator circuit and to help filter noise in the controller circuit. It can be seen Figure 2-35 that the raw 9.6 V nominal battery voltage is connected to the track drive motors. A maximum modulation ratio of the PWM signal to the dual motor driver limits the maximum voltage to the appropriate level for these motors. For the 6 V servo supply it was decided to use a switching voltage regulator (SVR) for better energy efficiency (96%) due to the high current draw in this circuit from the servos. For the low power circuit a standard voltage regulator (VR) is used to keep the voltage at 5 V. Devices that need 3.3V are equipped with their own VRs that can accept the 5 V source.

In order to power, mount, connect and protect (with fuses) the various electrical components a prototype circuit board was constructed for each module which can be seen in Figure 2-16. Figuring out how the various IO should be connected to the microcontroller can be a difficult task. Not all inputs on the Maple are 5 V tolerant and the various PWM, UART and DAC compatible pins often have several conflicting needs. Therefore, even with ample controller IO, the specific pin mapping had to be carefully evaluated early on in the development phase.

Chapter 3

Controls and Teleoperation

3.1 Control Architecture

The control for R2TM3 is governed by the operational modes defined in 2.1.2. As a recap, they are:

- Single Robot Drive (0).
- Single Robot Manipulation (1).
- Cooperative Climbing (2).
- Cooperative Steering (3).
- Cooperative Shifting (4).

The remote OCU provides a means of teleoperation and acts as a supervisory system, which selects the active operational mode based on operator inputs. For some modes, the robot(s) to be operated are selected as well. Furthermore, off-line computing is utilized on the PC to generate meaningful commands for the appropriate robot(s). The OCU must also handle sending commands and receiving signals from the robot modules, including passing robot-to-robot messages. A high level description of the control system is shown in Figure 3-1.

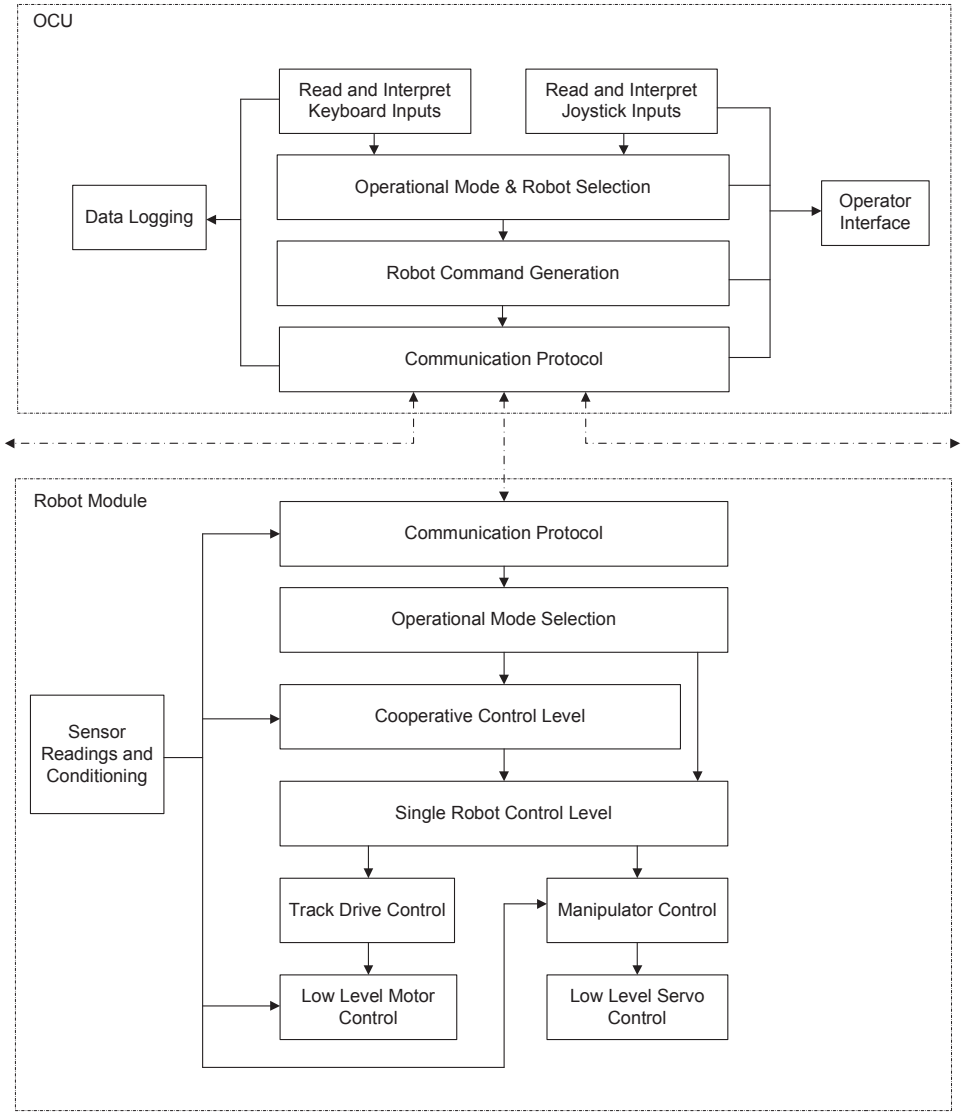

Figure 3-1: R2TM3 Control Block Diagram

Each robot module's software manages communication with the supervisory control system. It also interprets which operation mode the robot should be in and whether the robot is selected for operation. Depending on the mode, the cooperative control level may be used to run cooperative control algorithms. Single robot control algorithms interpret commands originating from the supervisory control system and/or the cooperative control algorithms. Control is then separated into track vehicle control and manipulator control. Finally, the low level control is performed for the

actuators. Sensor feedback is utilized at various levels depending on the specific need for the operational mode.

3.2 Teleoperation

The teleoperation scheme for the R2TM3 is intended to be simple and convenient. It provides a means of supervisory control, remote operation and data logging.

3.2.1 Operator Inputs and Interface

Using a standard laptop, operator inputs may be generated through the PC's keyboard or peripheral devices such as a mouse or game controller. In this case a Logitech Game Controller (GC) is used as the main device for inputting operator commands to the PC. The PC and GC together constitute a simple OCU. The GC has two, two axis, joysticks and a variety of buttons.

Figure 3-2: Game Controller

The interpretation of the GC inputs is different depending on the operational mode; the GC inputs are mapped to the appropriate robot command. The inputs to change the mode and selected robots for control are input from the keyboard. The meaning of the main keyboard inputs are shown in Table 3-1.

Table 3-1: Keyboard Inputs

Key	Meaning	Notes
0	Mode 0	-
1	Mode 1	-
2	Mode 2	-
3	Mode 3	-
4	Mode 4	-
a	Robot 1	For Mode 0 and 1
b	Robot 2	For Mode 0 and 1
c	Robot 3	For Mode 0 and 1
d	All Robots	For Mode 0 and 1
m	Manipulator On	For Mode 1
o	Manipulator Off	For Mode 1

Where, for simplicity, the robot modules have a fixed ID. The meaning of the GC inputs for the different operator modes are mapped in Table 3-2. See Figure 3-2 for GC inputs.

Table 3-2: Joystick Input Mapping

	Operation Mode				
GC Input	0	1	2	3	4
RH Joy X	Vehicle Yaw Rate	-	-	Path Radius Change Rate	Module Yaw Rate
RH Joy Y	-	Manip. X Rate	Module Pitch rate	Module Gap Change Rate	-
LH Joy X	-	Base Joint Rate	Front Module Yaw Rate	-	-
LH Joy Y	Vehicle Speed	Manip. Z Rate	Module Extend/Retract*	Group Speed	Module Speed
FL Up	-	Manup. Roll Joint +Rate	+Group Speed	-	-
FL Down	-	Manip. Roll Joint -Rate	-Group Speed	-	-
FR Up	-	Manup. Pitch Joint +Rate	+Module Speed and Extend	-	-
FR Down	-	Manup. Pitch Joint -Rate	-Module Speed and Extend	-	-
A	-	-	Change Extend/retract Type*	-	Use All Modules
B	-	-	-	-	-
X	-	Gripper Open	Decrement Module to Use	-	Decrement Module to Use
Y	-	Gripper Close	Increment Module to Use	-	Increment Module to Use

The specific interpretation of the inputs for the given modes and how they are used for various controllers are described in sections 3.3 and 3.4. In Table 3-2 module refers to the entire robot module and joystick inputs are proportional, while the other inputs are on/off. Consequently, the joystick inputs are used for the most suitable input in a given mode. Furthermore, use of the joystick's x and y input together is avoided, so that one can be used without inadvertently giving an input to the other.

The user interface in this scheme is simple since it's intended for prototyping and the operator is intended to be near the robots during testing. Operator feedback is given by printing values on the PC's program console. The LED located at the top of the robot docking shaft, seen in Figure 2-21, is also used to indicate which robot is active, to help the operator keep track of which robot has been selected, if applicable.

Another useful feature of this teleoperation scheme is that the PC's application may log data being sent back from the robots into a csv file, which can then be used for analysis and troubleshooting. This becomes invaluable during the implementation of control algorithms.

The teleoperation scheme used in this work is minimal, but is convenient and provides what is needed for prototyping. It's important to note that teleoperation communication delays are assumed to be negligible in this system. Details on the communication protocol are provided next.

3.2.2 Communication Protocol

The communication protocol used in this system is intended to be as simple as possible so that messages may be coded and decoded quickly. The communication protocol can be thought of as an application layer built over the XBee IEEE 802.15.4 protocol. The protocol header only consists of two bytes (characters) and a trailer consisting of one end byte (character). A message structure containing the header and trailer is shown in Figure 3-3.

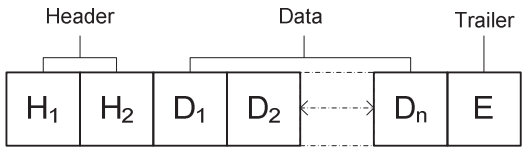

Figure 3-3: Message Structure

The first character in the header identifies the robot the message is intended for or sent from. In this point-to-multipoint network, the robots are always communicating with the master node, so a second

node ID is not needed. A special ID is used in indicate that a command is intended for all robots. The second character identifies the signal or command type, depending on which way the message is going. The meaning of this character has been specified in this protocol for all possible commands and signals used, as seen in Appendix B. The trailing character is always the same and is 'E'. This indicates that the end of the data in the message has been reached. The sent data is of variable length, can be integer or floating point and is read one character at a time.

Each of the robots in the network continuously checks their serial buffers for bytes of information. Messages are ignored unless the respective robot's ID or the "send all ID" is found. The next character indentifies the command type and tells the robot which internal variable to write the data to. The data is then read a byte (character) at a time to a string until the end character is read. The string is then converted and stored in the appropriate variable based on the command ID.

For the robots sending signals back to the PC's node, the decoding process works much in the same way, where the first ID tells the application on the PC which robot the message originated from. A caveat in this approach is that all robots may send messages at the same time potentially mixing signals. Therefore, a method of managing multi-robot communication must be devised at the application layer. A polling system has been implemented in which the master (PC node) polls each robot [40]. This is done by passing a permission character to each robot one at a time. The following steps outline the polling:

1. Master node sends $Robot_i$'s permission character as the last the command to $Robot_i$.
2. $Robot_i$ reads permission character and is enabled to send messages.
3. $Robot_i$ sends all relevant messages for given operation mode.
4. $Robot_i$ sends its permission character back to the master node.
5. $Robot_i$ disables message sending.
6. Master node receives $Robot_i$'s permission character back and then sends $Robot_{i+1}$'s permission character to $Robot_{i+1}$ as the last command.
7. Process continuous in a round-robin fashion.

With this polling method, the master node is only ever receiving messages from one robot at a time.

In the cooperative control methods described in section 3.4 the need for robot-to-robot communication has been minimized as much as possible to keep the control distributed. However, in some cases a single message needs to be sent between robots. Specifically, a given robot needs to send messages to the robot connected directly behind it, the following robot. Since there are so few messages that need to be sent (two unique messages total) this has been handled by creating a pseudo-command. In the appropriate mode the signal for the following robot, $Robot_{i+1}$, is sent to the PC by $Robot_i$. The signal is written to a new pseudo-command for $Robot_{i+1}$ and the command is sent. $Robot_{i+1}$handles this command data like any other and uses it as needed in the cooperative control algorithms.

Some of the pertinent protocol characters used in the protocol are shown in Table 3-3. A full list can be found in Appendix B.

Table 3-3: Protocol Characters

Character	Meaning	Notes
$	Robot 1 ID	-
%	Robot 2 ID	-
&	Robot 3 ID	-
#	All Robots ID	Send to all robots
:	Robot 1 Send	Permission character
*	Robot 2 Send	Permission character
-	Robot 3 Send	Permission character
E	End	Command or signal

This simple protocol has proved to be effective in the prototyping of R2TM3.

3.3 Single Robot Control

3.3.1 Modeling Single Robot Control

Single robot control can be categorized into manipulator control and track vehicle base control. The kinematic models for the manipulator are given in 2.2.3 and the reduced order 2D forward and inverse kinematics is utilized for manipulator control. In this scheme the desired wrist position can be related to θ_2 and θ_3 using inverse kinematic (see Figure 2-22 for definitions) and the actual θ_2 and θ_3

can be related to the actual wrist position using forward kinematics. The other manipulator DOF are controlled directly from operator inputs. Since the low level joint control is already implemented in the servo motors, no further manipulator modeling is needed for this manipulator control scheme.

The track vehicle base's kinematics was not previously analyzed like the manipulator was, but a model is needed for control. The track vehicle base is a skid-steered vehicle, which means that in order to steer, the tracks need to skid or slip relative to the ground [41],[42]. Inherent in this vehicle type is complex force interaction between the tracks and the ground. Consequently, the path of a skid-steered vehicle is very difficult to model accurately for a given set of track speeds and depends largely on the surface it is steering on [41]. In the case of a differential drive vehicle, it may be assumed that a nonholonomic constraint of zero lateral velocity exists [43]. This leads to a kinematic model for this vehicle type that accurately describes its motion. In the case of a skid-steered vehicle lateral velocity does exist due to the skidding.

A number of approaches are available for modeling and controlling skid-steer mobile robots such as estimating track or wheel slip online [43], estimating ground interaction parameters or making simplifying assumptions [41],[42]. If it is assumed that the vehicle travels at a relatively low speed, it steers on a horizontal plane in 2D, is on a firm surface and its COM is approximately at the geometric centre[10] of the vehicle, then a simple kinematic model may be used [41]. This model, similar to the differential drive model, is given as [41]:

$$\begin{bmatrix} v_T \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{\alpha B} & \frac{1}{\alpha B} \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix} \tag{3.1}$$

$$\begin{bmatrix} v_T \\ \dot{\psi} \end{bmatrix} = T_S \begin{bmatrix} v_L \\ v_R \end{bmatrix}$$

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \begin{bmatrix} 1 & \frac{\alpha B}{2} \\ 1 & -\frac{\alpha B}{2} \end{bmatrix} \begin{bmatrix} v_T \\ \dot{\psi} \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = {T_S}^{-1} \begin{bmatrix} v_T \\ \dot{\psi} \end{bmatrix}$$

$$R_p = \frac{v_T}{\dot{\psi}} \tag{3.3}$$

[10] Geometric centre as viewed from the top – in the X-Y plane.

In equation (3.1) v_L and v_R are the left and right track speeds respectively, B is the track width of the vehicle and v_T and $\dot{\psi}$ are the input vehicle speed and yaw rate. α is an expansion factor and is used to modify the model from a differential drive model to a skid steer model. This parameter is dependent on the surface the vehicle travels on and can be tuned experimentally, by changing it until the turn radius is correct for a given input speed and yaw rate, from equation (3.3). The assumptions for using this model have been validated for the R2TM3 vehicle type such as low speed travel and the COM in the centre of the vehicle. Using a simplified dynamic model for steady state steering, that can be found in [42], that includes the centrifugal effects of high speed steering the robot was simulated to find the valid low speed range and to analyze the effect of a shift in COM due to the manipulator.

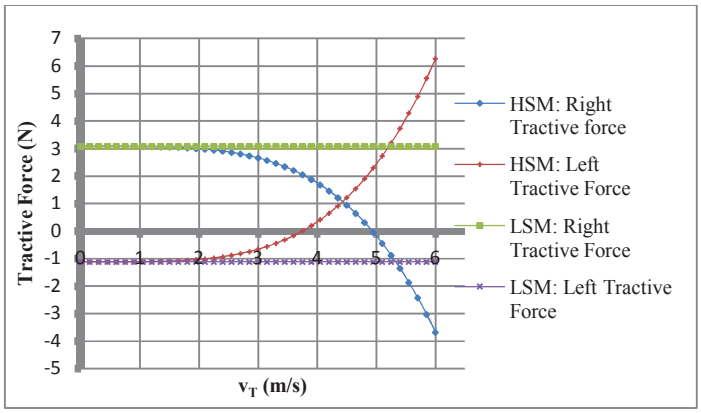

Figure 3-4: Low Speed Skid-Steer Validation

In Figure 3-4 the tractive force for the left and right tracks for a steady-state turn is plotted for various speeds. This is done for both a high speed model (HSM) which takes into account the centrifugal forces during the turn and a low speed model (LSM) which ignores this effect [42]; the straight lines on the graph. It can be clearly seen that for the R2TM3 vehicle base, the two models give the same results under approximately 2 m/s (6.6 ft/s). The R2TM3 vehicle base travels at speeds well below this upper speed limit. Therefore, the dynamic centrifugal effects are not significant when steering.

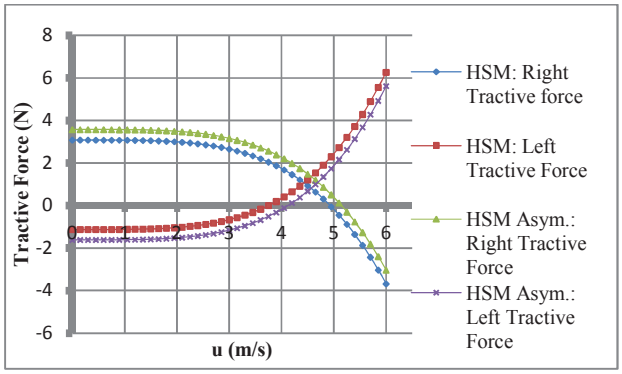

Figure 3-5: COM Symmetry for Steering

In Figure 3-5 a plot is shown for the case where the manipulator has been moved to create a shift in the COM to be at ¾ of the total width at one side of the vehicle. This case, along with others, show that asymmetry due to the manipulator position does not have a large effect on the system behavior during a turn. Due to the manipulator being light and the minimal effects of the COM shift, it is assumed that COM can be considered to be at the geometric centre of the vehicle base.

With a low speed and symmetry assumption, the kinematic model given by equation (3.1) can be used as the R2TM3 skid-steer model for control of the vehicle base. Low level speed control will use standard PID techniques described in 3.3.3.

3.3.2 Manipulator Control

The manipulator control is done using kinematics and the existing low level RC servo position controllers. Input velocities, either from the remote operator or the cooperative control level are integrated for the required wrist displacements, added to the actual wrist positions then inverse kinematics is used to find the reference angles θ_{2_r} and θ_{3_r}, where subscript r denotes the reference point, or set point. In addition the other joint angle rates are input directly and integrated for a desired change in angle, or the position reference point is solved from the cooperative control level. Figure 3-6 shows the control block diagram describing the control method for joints 2 and 3.

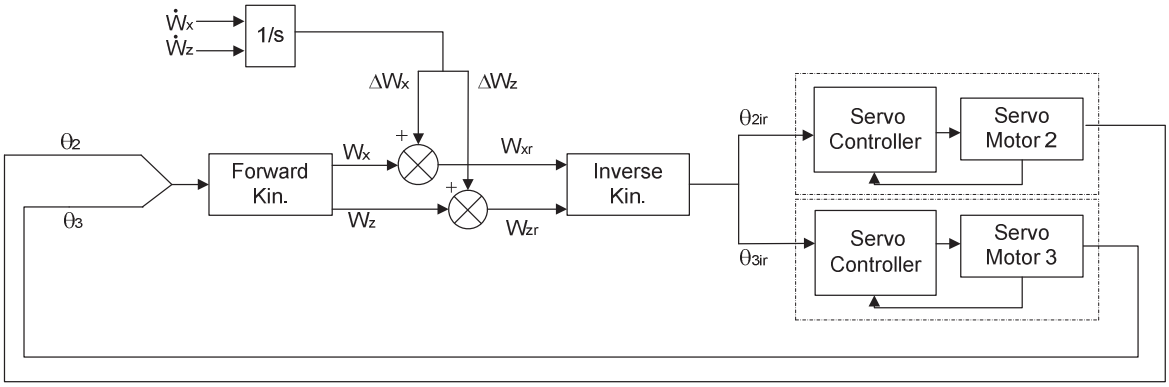

Figure 3-6: Manipulator Control Block Diagram

In Figure 3-6 it can be seen that forward kinematics is used to find the actual wrist x, z position. The control could have been implemented such that the commanded joint angles to the servo controllers are assumed to be the actual angles and that they reach the actual angles quite quickly. For smooth and relatively slow input rates, this assumption is valid and the control of the manipulator works well. However, for single robot teleoperated control an effort was made to use the external position feedback to make sure the actual wrist position is equal, or close to, the commanded position. The integrated input rates, a requested change in manipulator wrist position, are added to the actual position, to find the reference position. Once the actual position becomes equal to the reference position, the manipulator will stop moving.

For the manipulator joints (1, 4 and 5) that are controlled directly, a block diagram for a general joint can be seen in Figure 3-7.

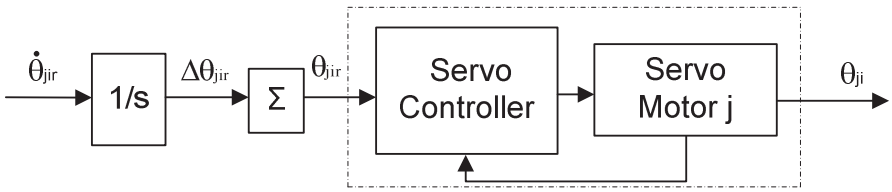

Figure 3-7: Single Joint Control

For the direct joint control, it can be seen that a Σ is used to show that the requested change in angle is added to the previous angle reference, to find the new angle reference. This notation is also used in a teleoperated control scheme in [44] for the HELIOS IX robot. This direct joint scheme does not make use of the external feedback of the motor and correctly assumes the servo motor responds fast

enough so that the actual angle is close to, or equal to, the commanded reference angle. If this were not the case, the motor would always appear to lag the commands of the remote operator, making the teleoperation awkward. The gripper motor is controlled similarly except that only two different reference angle values are commanded from the operator corresponding to the open and close positions of the gripper.

Another feature for the manipulator control is the ability to turn off any one servo motor. If the servo controller does not receive a signal after a small amount of time it will turn off. Since these servo motors are back-drivable this allows the manipulator to be freely moved if all motors are commanded off. In this case, once the manipulator is moved to a new position, the external position feedback can be used for the new servo motor commands once the manipulator is turned back on; meaning the arm can be turned back on and will remain where it was moved to while turned off. This feature proves very useful for module docking and handling kinematic redundancy.

3.3.3 Track Base Control

For control of the track base the inverse of the kinematic model T_S previously described is used to transform the speed and yaw rate inputs to desired track speeds. The track speed set points are then used by a PID control loop for each track to control the speed.

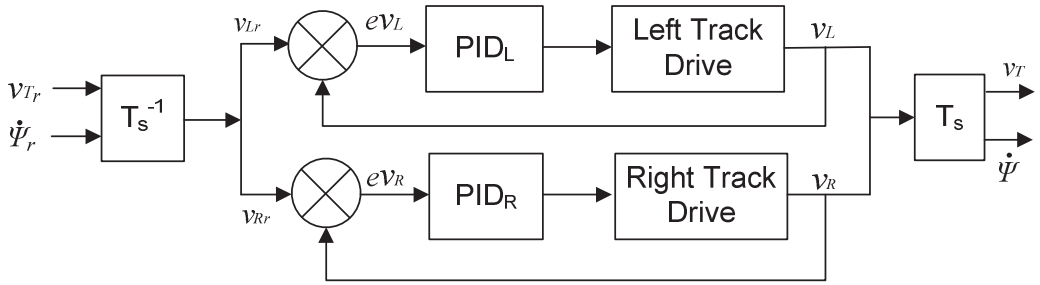

Figure 3-8: Track Control Block Diagram

The P, I and D gains are then tuned experimentally for the PID speed controllers.

3.4 Cooperative Robot Control

3.4.1 Modeling Cooperative Robot Control

For the different cooperative control functions models are needed for the connected modules that describe how they need to interact or move in relation to one another. The models for cooperative robot control are kinematic and involve kinematic transformations for position or velocity, just as in single robot control. Each of the main cooperative function models will be described next. The concepts and rationale for the cooperative functions may be reviewed in 2.1.2.

Cooperative Steering:

The goal of the model for cooperative steering is to describe the formation the robots are to assume based on inputs and to describe the corresponding set points for the individual robots. This model is 2D in the X-Y plane, is found geometrically and may be seen in Figure 3-9.

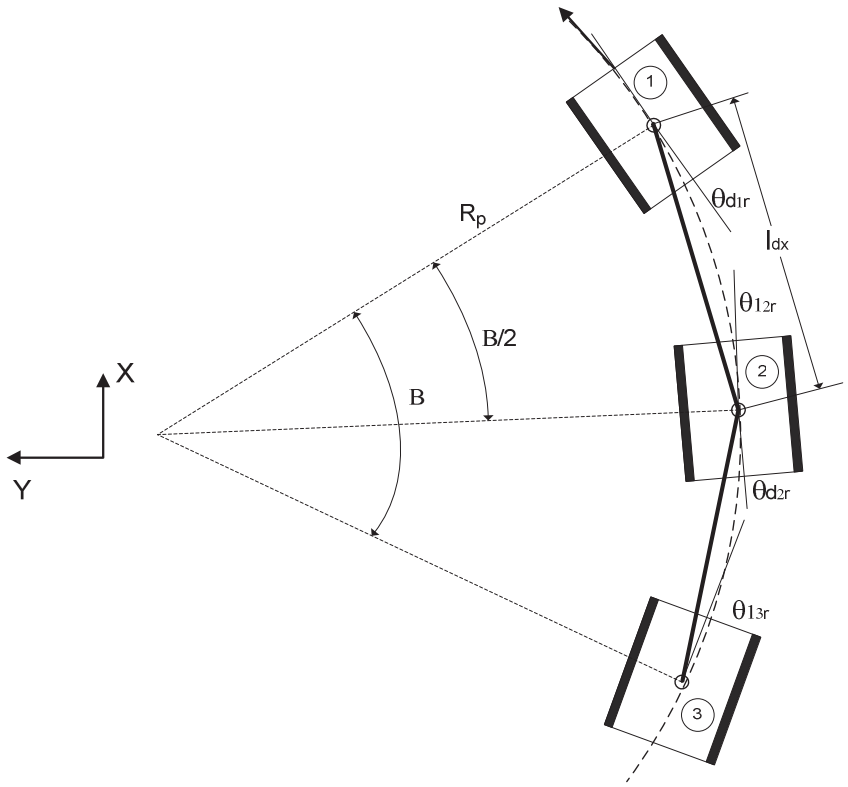

Figure 3-9: Cooperative Steering Model

In the cooperative steering model, every module is assumed to travel at the same speed v_T turning on a path radius R_p in order to maintain the arch formation. With the speed and path radius defined, it is of interest to find the set points for the module base joint angles and docking angles. From the simple geometry of fitting the discrete chain of connected modules onto a continuous arch, the joint angle set points can be found as:

$$\beta = 4\sin^{-1}\left(\frac{l_{dx_r}}{2R_p}\right) \tag{3.4}$$

$$\theta_{d_r} = -\frac{\beta}{4} \tag{3.5}$$

$$\theta_{d_r} = -\sin^{-1}\left(\frac{l_{dx_r}}{2R_p}\right)$$

$$\theta_{1_r} = \frac{\beta}{4} \tag{3.6}$$

$$\theta_{1_r} = \sin^{-1}\left(\frac{l_{dx_r}}{2R_p}\right)$$

In equations (3.4) - (3.6) l_{dx_r} is the gap distance set point and θ_{d_r}, θ_{1_r} are the angular set points. The docking angle requires a negative sign due to the chosen sign convention. It should be mentioned that the front and rear modules exclude the calculation for the base joint angle and docking angle respectively. The actual gap distance l_{d_x} may be related to the actual joint angles as:

$$l_{dx_i} = l_m\left(\cos\left(\theta_{2_i}\right) + \cos\left(\theta_{2_i} + \theta_{3_i}\right)\right) + l_{mo}\left(1 + \cos\left(\theta_{2_i} + \theta_{3_i} + \theta_{5_i}\right)\right) \tag{3.7}$$

With this model and the cooperative control described in 3.4.2 each module's single robot control can be employed as a lower control level to achieve the desired motion.

Cooperative Shifting:

Cooperative shifting is a special case of manoeuvring in the X-Y plane. In this case the cooperative model only consists of the modules' single robot vehicle base models and the means to select which of the robots should be moved. The details of the control algorithm are given in 3.4.3. The main assumption that makes this a special case of cooperative steering is that the robot modules are connected in a straight formation where the base joint and docking joint values are 0°.

Cooperative Climbing:

The goal of the model for cooperative climbing is to describe the relative motion of the robot modules during climbing, involving both the tracks and the manipulators in the connected chain. The model is reduced order in the X-Z plane and joints 1 and 4 are fixed at 0°. Several cooperative motions must be described so that proper relative motion is achieved:

- Pitching of a given module.
- Extending or retracting a given module with respect to the rest of the connected chain.
- Moving the connected modules together, while one moves in a different direction.

Velocity transformations will be described for each of these cooperative climbing motions. The general case of a middle robot, not on the end of the chain, will be used since both its own manipulator and the one connected to it participate in the cooperative motion.

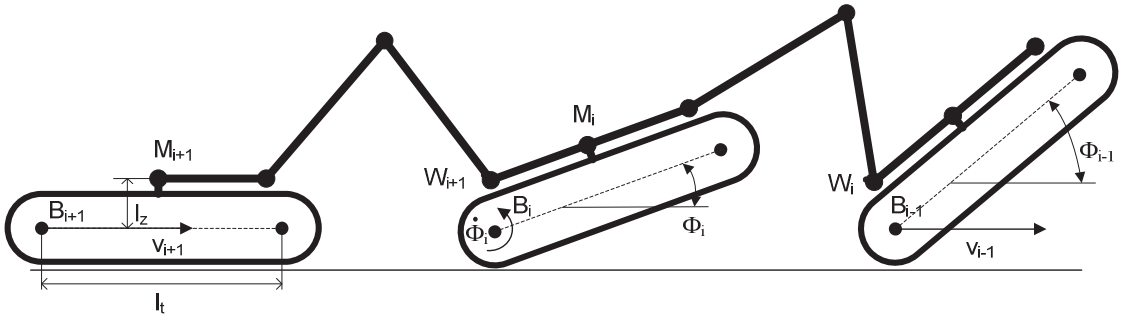

Figure 3-10: Cooperative Climbing Model

For a given module pitch rate $\dot{\phi}_i$, the module's track speed, manipulator rates and the connected manipulator rates need to be found. In this model, it is assumed that the module pitches around point B_i. Figure 3-11 describes the pitching velocity relationships.

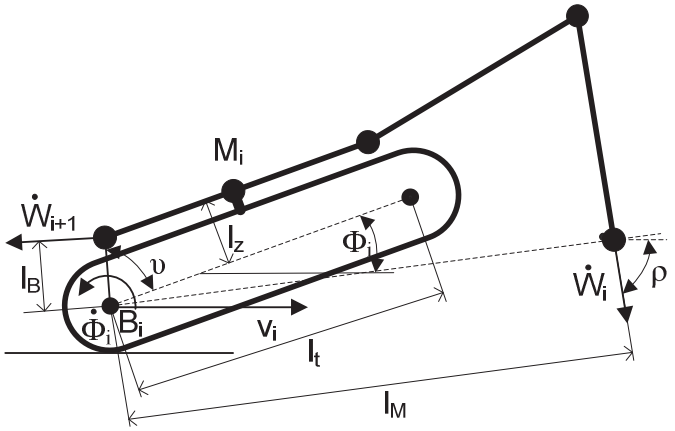

Figure 3-11: Pitching Velocity Manipulator i

Where for the tracks, of track radius r the speed can be found as:

$$v_i(\dot{\phi}_\iota) = \dot{\phi}_\iota r \tag{3.8}$$

The module's own manipulator rates at point W_i , due to pitching, can be solved as:

$$\begin{aligned} WB_{x_i} &= W_{x_i} + \frac{l_t}{2} \\ WB_{z_i} &= W_{z_i} + l_z \end{aligned} \tag{3.9}$$

$$\begin{aligned} \rho_i &= \tan^{-1}\left(\frac{WB_{z_i}}{WB_{x_i}}\right) - \frac{\pi}{2} \\ l_M &= \sqrt{{WB_{x_i}}^2 + {WB_{z_i}}^2} \\ \dot{W}_{x_i}(\dot{\phi}_\iota) &= \dot{\phi}_\iota l_M \cos\rho_i \\ \dot{W}_{z_i}(\dot{\phi}_\iota) &= \dot{\phi}_\iota l_M \sin\rho_i \end{aligned} \tag{3.10}$$

W_{x_i} and W_{z_i} are the wrist reference positions described in the manipulator's base frame. Finally the connected manipulator's rates at point W_{i+1}, due to pitch rate $\dot{\phi}_\iota$, can be solved as:

$$\Delta\phi_i = \phi_i - \phi_{i+1} \tag{3.11}$$

$$\dot{W}_{x_{i+1}}(\dot{\phi}_\iota) = \dot{\phi}_\iota l_B \cos\left(\upsilon + \Delta\phi_i + \frac{\pi}{2}\right) \tag{3.12}$$

$$\dot{W}_{z_{i+1}}(\dot{\phi}_\iota) = \dot{\phi}_\iota l_B \sin\left(\upsilon + \Delta\phi_i + \frac{\pi}{2}\right)$$

Geometric parameters and rates are detailed in Figure 3-11. The manipulator x, z rates have been solved with respect to their own base coordinate system. Therefore, they can be integrated and used for single robot inverse kinematics. The final joint rate for θ_5 is not needed since it is managed using a geometric relationship in the control method.

For extending or retracting a given module in the chain, two cases are considered; one where the module is extended horizontally with respect to the global frame and another where it is extended along a plane defined by its own pitch angle. Figure 3-12 shows both cases and how they relate to manipulator and track speeds.

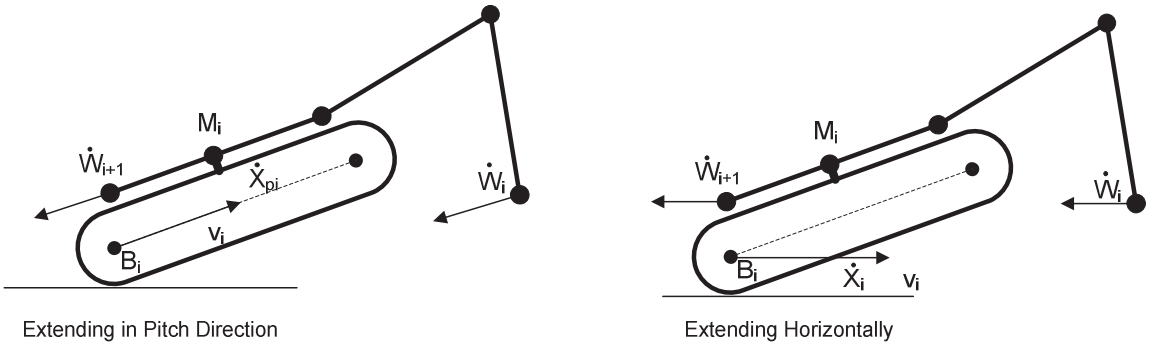

Figure 3-12: Module Extending

For the tracks, in either case, their speed is equal to the extend or retract rate even though the meaning is different. The manipulator rates, as before, need to be described in their own base frames. For the horizontal case:

$$v_i(\dot{X}_i) = \dot{X}_i \tag{3.13}$$

$$\dot{W}_{x_i}(\dot{X}_\iota) = -\dot{X}_\iota \cos(\phi_i) \tag{3.14}$$

$$\dot{W}_{z_i}(\dot{X}_\iota) = \dot{X}_\iota \sin(\phi_i)$$

$$\dot{W}_{x_{i+1}}(\dot{X}_i) = \dot{X}_i \cos(\phi_{i+1})$$
$$\dot{W}_{z_{i+1}}(\dot{X}_i) = -\dot{X}_i \sin(\phi_{i+1}) \tag{3.15}$$

For the extension/retraction along the pitch angle:

$$v_i(\dot{X}_{pi}) = \dot{X}_{pi} \tag{3.16}$$

$$\dot{W}_{x_i}(\dot{X}_{pi}) = -\dot{X}_{pi} \tag{3.17}$$

$$\dot{W}_{z_i}(\dot{X}_{pi}) = 0$$

$$\dot{W}_{x_{i+1}}(\dot{X}_{pi}) = \dot{X}_{pi} \cos(\Delta\phi_i)$$
$$\dot{W}_{z_{i+1}}(\dot{X}_{pi}) = \dot{X}_{pi} \sin(\Delta\phi_i) \tag{3.18}$$

Lastly, the case where a module extension occurs along its own pitch angle while the rest of the modules move horizontally will be analyzed. Figure 3-13 depicts this motion.

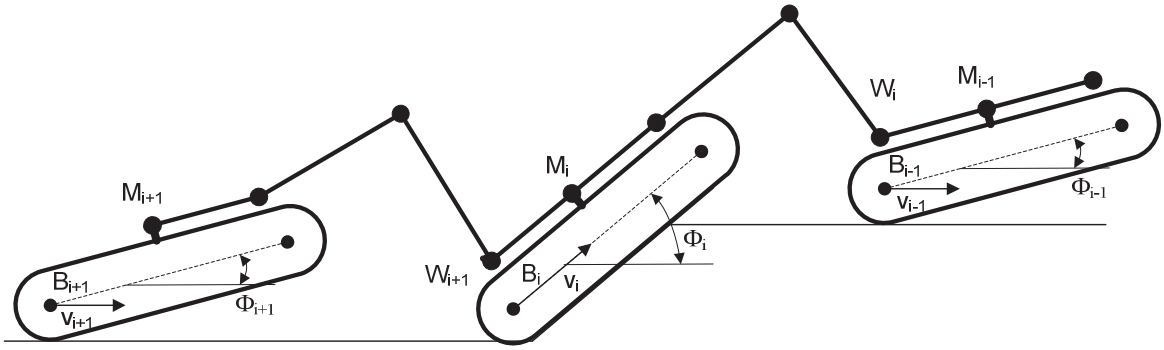

Figure 3-13: Combined Speed, One Module along Pitch

The track and manipulator rates due to this motion are:

$$v_i(v_{pi}) = v_{i+1}(v_{pi}) = v_{i-1}(v_{pi}) = v_{pi} \tag{3.19}$$

$$\dot{W}_{x_i}(v_{pi}) = v_{pi}[\cos(-\phi_i) - 1] \tag{3.20}$$

$$\dot{W}_{z_i}(v_{pi}) = v_{pi}\sin(-\phi_i)$$

$$\dot{W}_{x_{i+1}}(v_{pi}) = v_{pi}[\cos(\Delta\phi_i) - \cos(\phi_{i+1})] \tag{3.21}$$

$$\dot{W}_{z_{i+1}}(v_{pi}) = v_{pi}[\sin(\Delta\phi_i) + \sin(\phi_{i+1})]$$

Another simple case, where an overall group speed is desired can be given. In this case the manipulator rates are zero since there is no relative velocity between modules:

$$v_i(v_t) = v_{i+1}(v_t) = v_{i-1}(v_t) = v_t \tag{3.22}$$

With these cooperative robot models, the cooperative control methods can be described.

3.4.2 Cooperative Steering Control

For cooperative steering it was decided to have the manipulator's motors turned off, excluding the gripper, since this is much easier on the manipulator motors and the manipulator will not need to be active most of the time, when terrain is only moderately rough. However, in this connected formation, the robots are ready for when an obstacle is encountered that needs the manipulators to actively cooperate. This control strategy then becomes a type of formation control that uses a path tracking algorithm and a proportional gap controller. In this scheme the manipulator acts mainly as a feedback device, but has some passive stabilizing effects on the connected modules.

The inputs for the cooperative control are the travel speed v_T and the path radius R_p as seen in the cooperative steering model. The individual modules are intended to travel at the same speed and maintain a fixed distance, or gap defined by l_{dx}, away from one another. Each module receives the same speed command, but due to disturbances and variations in module parameters, the speeds will be different, breaking the desired formation. Therefore a proportional feedback controller, using manipulator position feedback, can be used to maintain the module gaps.

As described in the cooperative steering model, the modules are intended to travel on an arch defined by a commanded radius. For the lead robot this is a matter of finding the correct yaw rate, but for the following robots a path tracking algorithm is used to make sure the robots follow the path

defined by the arch radius. The following robots use the set points previously defined for their base joint angles and docking angles. The path tracker chosen is based on the "Control Theory Approach" [45] and, like most path trackers, makes use of a look ahead distance [45],[46]. The cooperative steering algorithms are depicted in Figure 3-14.

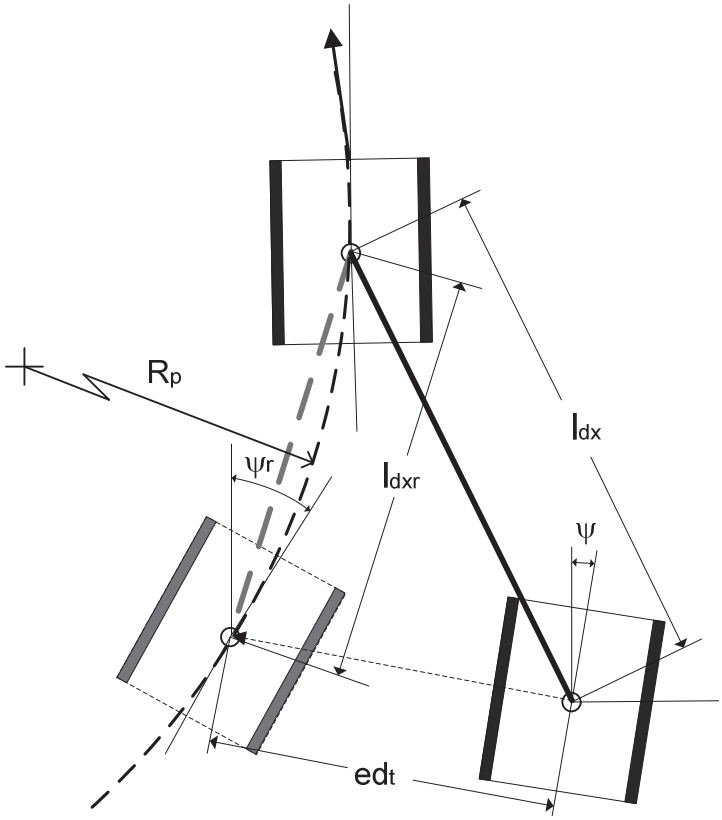

Figure 3-14: Path Tracker and Gap Controller

The error for the gap controller is:

$$el_{dx_i} = l_{dx_r} - l_{dx_i} \tag{3.23}$$

The gap controller for following robots is given as:

$$v_i = v_T - k_g\left(el_{dx_i}\right) \tag{3.24}$$

Where v_T is a feed-forward term from the operator speed command and k_g is the proportional gain for the gap error. Consequently, the module will speed up or slow down to compensate for gap errors.

It was determined experimentally that having the gain adapt to different speed conditions enabled better control. Therefore, k_g is a linear function of the module speed, $k_g(v_i)$.

For the lead robot no controller is used since it dictates the travel speed of the group.

$$v_i = v_T \tag{3.25}$$

The errors for the path tracker, based on the set points defined in the model are:

$$\psi_i = \theta_{d_{i-1}} - \theta_{1_i} \tag{3.26}$$

$$\psi_{ir} = \theta_{d_{(i-1)r}} - \theta_{1_{ir}}$$

$$e\psi_i = \psi_{ir} - \psi_i$$

$$e\theta_{d_{i-1}} = \theta_{d_{(i-1)r}} - \theta_{d_{i-1}} \tag{3.27}$$

$$|ed_{ti}| = \sqrt{{l_{dx_r}}^2 + {l_{dx_i}}^2 - 2l_{dx_r}l_{dx_i}\cos(e\theta_{d_{i-1}})}$$

$$ed_{ti} = \sin(e\theta_{d_{i-1}})\,|ed_{ti}|$$

The path tracker for following robots is:

$$\gamma_{p_i} = -k_h e\psi_i + k_c ed_{ti} \tag{3.28}$$

$$\dot{\psi}_i = \gamma_{p_i} v_i$$

For the lead robot:

$$\dot{\psi}_i = \frac{v_T}{R_p} \tag{3.29}$$

The heading ψ_i for each following robot is with respect to the coordinate frame of the robot ahead of it. γ_{p_i} is the instantaneous path curvature, k_h and k_c are the gains. It can be seen in equation (3.28) that the path tracking algorithm is a proportional controller with two terms. The proportional terms in the path tracker are opposite in sign, yielding a swerving behavior when the robot corrects for path errors. In this case the calculated cross track error is not a true cross track error since it is not perpendicular to the path tangent. However, this difference is small and the controller is not sensitive

to it. In fact, the look ahead distance for path trackers can be tuned to give different performance characteristics; a larger look ahead distance makes it less responsive, while a small look ahead distance makes it more responsive [45]. Therefore, this modified definition of cross track error has the same effect as having a look ahead distance that varies slightly online. The advantage of this modified cross track error definition is the ease of calculation in this implementation.

It's important to note that this control is mostly distributed. Each module responds to operator commands in a suitable way and relies mostly on its own feedback for maintaining the formation. The only exception is the need to send the docking angle of a given module to the one directly following it. The docking angle is always initialized at 0° requiring the modules to be in a straight formation when the mode is initiated. Additional sensory would eliminate this requirement.

3.4.3 Cooperative Shifting Control

For cooperative shifting control, the single robot track control is used (speed and yaw rate), but which robot in the connected chain to use needs to be determined. Shifting them all at once is also desired since it's faster. However, in this scheme the manipulators are active to help ensure that the modules rotate about their geometric centres, about the docking point. By shifting one at a time, the active manipulators are connected to the other stationary modules, which help to counteract any skid-steer forces that may tend to move the module off centre. Therefore, the option to shift all the modules at once or one at a time has been given. The shifting concept can be seen in Figure 2-4.

Another problem to manage for this cooperative function is dealing with the redundancy between the tracks and the manipulator's base joint. The base joint would constrain a module from steering in the connected chain. This is handled by utilizing the servo motor on/off control, where the base joint is switched off for a given module selected for shifting. For the speed command all modules simply respond to the command in the same way travelling in the same direction. Any differences in speed are prevented by the active connected manipulators. It is only appropriate, with this control approach, to input a speed command when the modules are not steering.

The cooperative shifting algorithm, as it pertains to steering for a single module, can be described as:

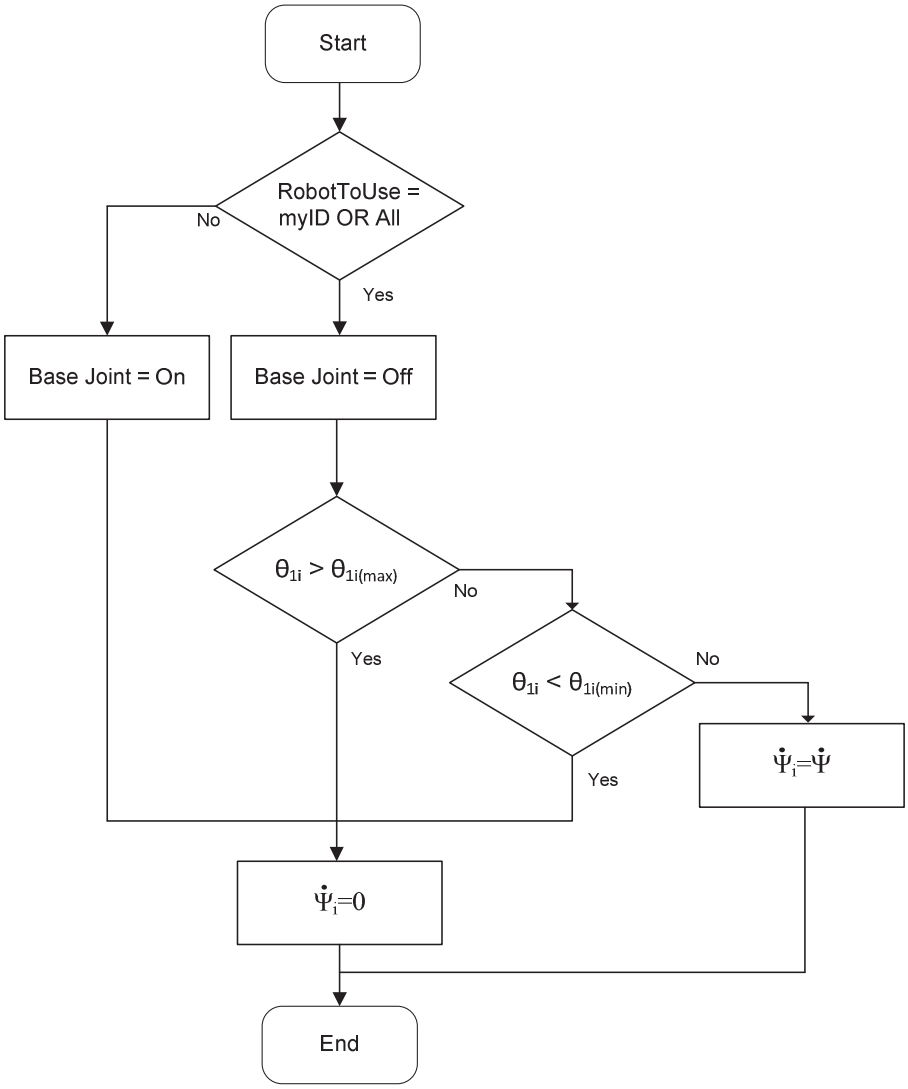

Figure 3-15: Cooperative Shifting Algorithm

It can be seen in Figure 3-15 that base joint feedback is used to detect joint limits to prevent further steering which might damage the base joint motor.

3.4.4 Cooperative Climbing Control

Cooperative climbing control is done by selecting a module in the connected chain that is to perform the actual climbing transition, over the obstacle. With a given module selected, the following inputs can be given to achieve ledge climbing:

1. Module pitch rate $\dot{\phi}_i$.
2. Module extend/retract rate (horizontal) with respect to the other modules in the chain $\dot{X}_i$.
3. Module extend/retract rate along its own pitch plane $\dot{X}_{pi}$.
4. Overall group speed (no relative speed between modules) v_T.
5. Overall group speed with selected module moving along its own pitch plane v_{pi}.

These inputs can be given to the selected module and/or to the group in any combination. Although, typically only one input is given at a time by the remote operator to negotiate the obstacle. How these inputs relate to the track speeds and manipulator rates can be seen in the description of the cooperative climbing model in 3.4.1.

This approach relies on several assumptions:

1. The modules rotation, for pitching, occurs at the rear center point of the tracks B_i.
2. Track speeds and manipulator speeds can be synchronized well enough so that excessive slippage and/or manipulator loading caused by the tracks does not occur. The low level feedback controllers can adequately synchronize the tracks and manipulators.
3. Speeds between connected modules can be synchronized well enough to avoid the same problems.
4. The combined track input and manipulator input is able to hold any given pitch angle required for climbing. Small pitch angles aren't useful and large pitch angles are easier to hold.

Robot feedback is not used directly in the cooperative climbing control; only the operator's own visual feedback is used to figure out the needed inputs. This control scheme therefore relies on the single robot control and low level feedback to adequately respond to the needed positions and speeds.

As before the required manipulator rates are integrated to find the displacements and set points, while the track speed PID controllers are used for track speed commands. The track control for cooperative climbing is described as:

$$v_i(\dot{\phi}_i, v_T, v_{pi}, \dot{X}_i, \dot{X}_{pi}) = v_i(\dot{\phi}_i) + v_i(v_T) + v_i(v_{pi}) + v_i(\dot{X}_i) + v_i(\dot{X}_{pi}) \qquad (3.30)$$

Where the track speed due to different inputs is summed to give an *OR* condition.

The manipulator rates, also summed, for a given module and the one connected behind it are:

$$\begin{aligned}\dot{W}_{x_i}(\dot{\phi}_\iota, v_T, v_{pi}, \dot{X}_i, \dot{X}_{pi}) \\ = \dot{W}_{x_i}(\dot{\phi}_\iota) + \dot{W}_{x_i}(v_T) + \dot{W}_{x_i}(v_{pi}) + \dot{W}_{x_i}(\dot{X}_i) + \dot{W}_{x_i}(\dot{X}_{pi}) \\ \dot{W}_{z_i}(\dot{\phi}_\iota, v_T, v_{pi}, \dot{X}_i, \dot{X}_{pi}) \\ = \dot{W}_{z_i}(\dot{\phi}_\iota) + \dot{W}_{z_i}(v_T) + \dot{W}_{z_i}(v_{pi}) + \dot{W}_{z_i}(\dot{X}_i) + \dot{W}_{z_i}(\dot{X}_{pi})\end{aligned} \tag{3.31}$$

$$\begin{aligned}\dot{W}_{x_{i+1}}(\dot{\phi}_\iota, v_T, v_{pi}, \dot{X}_i, \dot{X}_{pi}) \\ = \dot{W}_{x_{i+1}}(\dot{\phi}_\iota) + \dot{W}_{x_{i+1}}(v_T) + \dot{W}_{x_{i+1}}(v_{pi}) + \dot{W}_{x_{i+1}}(\dot{X}_i) \\ + \dot{W}_{x_{i+1}}(\dot{X}_{pi}) \\ \dot{W}_{z_{i+1}}(\dot{\phi}_\iota, v_T, v_{pi}, \dot{X}_i, \dot{X}_{pi}) \\ = \dot{W}_{z_{i+1}}(\dot{\phi}_\iota) + \dot{W}_{z_{i+1}}(v_T) + \dot{W}_{z_{i+1}}(v_{pi}) + \dot{W}_{z_{i+1}}(\dot{X}_i) \\ + \dot{W}_{z_{i+1}}(\dot{X}_{pi})\end{aligned} \tag{3.32}$$

Finally, θ_{5_i} may be solved by a geometric constraint:

$$\theta_{5_i} = \phi_{i-1} + \phi_i - \theta_{2_i} - \theta_{3_i} \tag{3.33}$$

The individual terms of the manipulator rates are described in 3.4.1. The pitch angles in (3.33) may be found from measured feedback from the tilt sensors or integrated and summed values from the pitch rates. The control scheme is as distributed as the cooperative steering control, where in this case, the pitch angle needs to be communicated from a given robot to the one connected directly behind it. The control scheme, for a given module, can be seen in a block diagram in Figure 3-16.

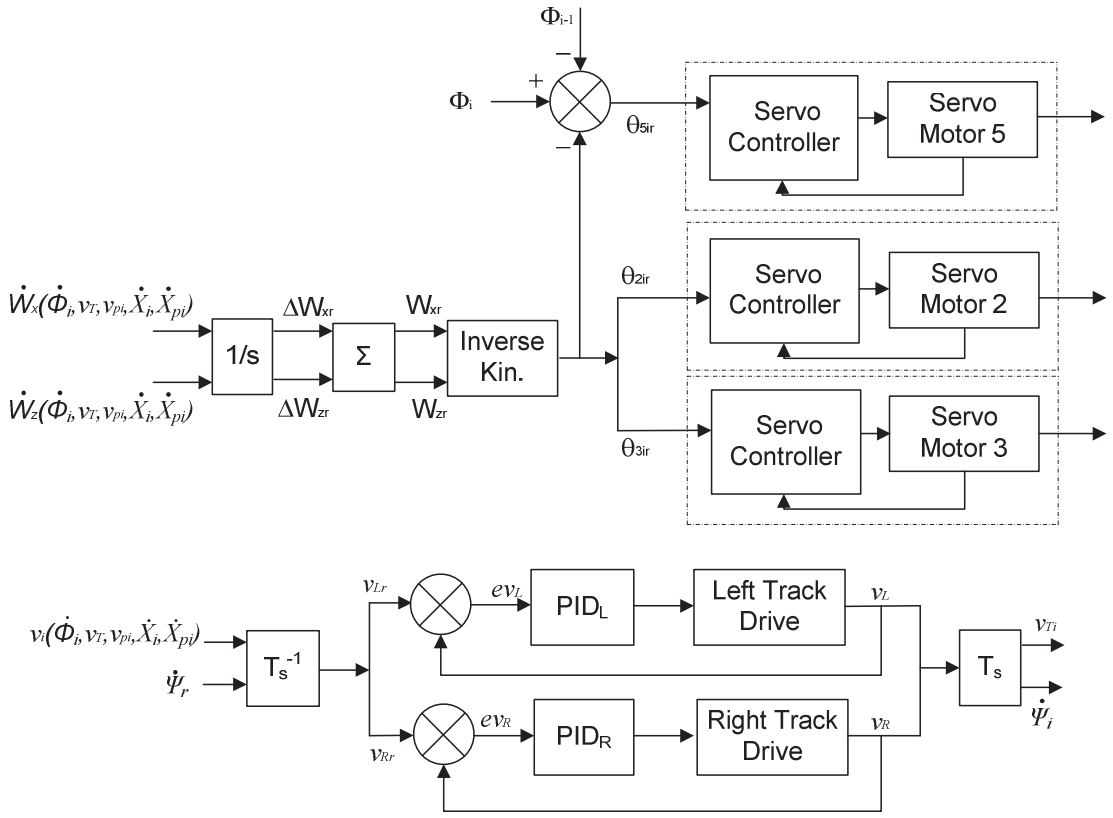

Figure 3-16: Cooperative Climbing Block Diagram

A steering input $\dot{\psi}_r$ is seen in Figure 3-16 that is only applicable for the front module. This module is not constrained by its base joint so steering can be used to compensate for heading errors due to slippage when climbing. The manipulator wrist reference position is initialized using forward kinematics when the cooperative climbing mode is initially selected. Commanded wrist displacement is summed to the previous reference position to find the new one, similar to the direct joint control in 3.3.2.

This cooperative control approach allows for the experimentation of the climbing performance of the R2TM3 with no mapping requirements or knowledge of the terrain. It requires the operator to make decisions on what inputs are appropriate to achieve climbing. This control approach gives the fundamentals on how to make the system climb and helps validate the concept. With further control and hardware enhancements the control could be more generalized; the modules could pitch about any point under the tracks while making climbing transitions, depending on the known contact conditions with the ground.

Chapter 4

Implementation and Testing

4.1 Implementation

This section will focus on several considerations and methods used for implementation of the control algorithms and other required program functions for the R2TM3. The controls were programmed for the robot modules using the open source Maple IDE, which uses a C based programming language called Wiring. In addition standard libraries as well as C++ libraries can be utilized. For example the Servo library can be used to conveniently control a servo motor. On the PC side an open source IDE called Processing is used, which uses a very similar programming language based on Java. Libraries are available for Processing such as the libraries used for serial port communication and interfacing with peripheral devices (game controller). The similarity between the two open source environments, simple programming environments and variety of open source libraries made these IDEs ideal for prototyping of the R2TM3. Many parts of the R2TM3 program borrowed from or were inspired from programs provided in the Arduino Playground [47]. Some selected sections from the R2TM3 program can be seen in Appendix C.

Further implementation considerations and techniques will be described as they pertain specifically to single robot implementation and cooperative robot implementation.

4.1.1 Single Robot Implementation

The majority of the implementation techniques and considerations pertain to single robot control, since at this level, the actuator control and most sensor feedback is handled. The first thing that was implemented was the PC to robot communication so that values could be sent back from the robot to the PC for troubleshooting.

Getting reliable communication between the robot and PC to work is a difficult task and is very sensitive to timing. As described in the protocol the robot module only reads commands intended for it and then determines what kind of command it is. The microcontroller can read bytes from the serial buffer quickly, but converting the message string to an integer, for example, using atoi[11] takes longer. Consequently, parts of the actual command data can be missed. By adding a 10 ms delay in the script the microcontroller had enough time to read the entire message reliably. However, this time delay can cause instability in feedback loops such as the track PID speed controllers. Fortunately, with other parts of the robot control eventually implemented, such as the PIDs themselves, the delays are no longer needed; the program takes long enough to compute. Ultimately, achieving reliable serial communication can be a challenging task, requiring some trial and error.

For the manipulator control, joint position feedback is used for forward kinematics. As seen in Figure 3-6 the actual wrist position and requested wrist position[12] are being compared at the summing junction. Any noise in the feedback used for forward kinematics causes the manipulator to jitter. Therefore, a simple software digital low pass filter and sample averaging (running average) is used for the joint position feedback, see Appendix C for detailed implementation. This smoothes out the position signals and helps with this problem greatly. Other techniques used to eliminate jitter are to use thresholds and disable parts of the control algorithm when there are no operator inputs. The joint angles are integer values and changes in one or two degrees can be mostly due to rounding. Therefore a threshold of two degrees is used; where any difference in angle set point and measured angle that is less than the threshold is ignored. In addition, inverse kinematics is not even computed when there are no commands from the operator. The combined effect of these rather simple techniques can greatly improve the smoothness of control.

The sensor noise and errors are due to several things; the serial communication wire running alongside the feedback wires, the integer rounding and the servo motors vibrating while under load.

[11] C++ function for converting a string to an integer
[12] Wrist rates are integrated using finite difference

Figure 4-1 and Figure 4-2 show the difference in sensor noise for the cases of the servo under no load and under load, which illustrates that the majority of the noise is from the servo motors vibrating under load.

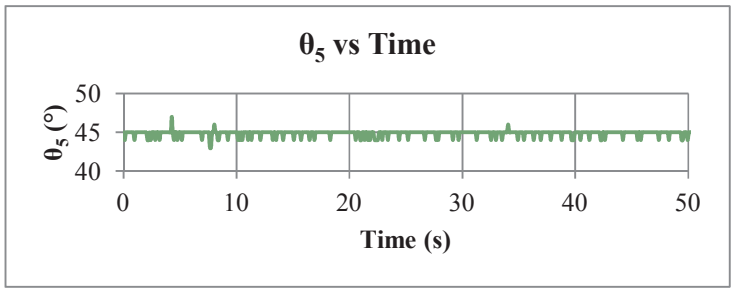

Figure 4-1: Sensor Noise, No Servo Load

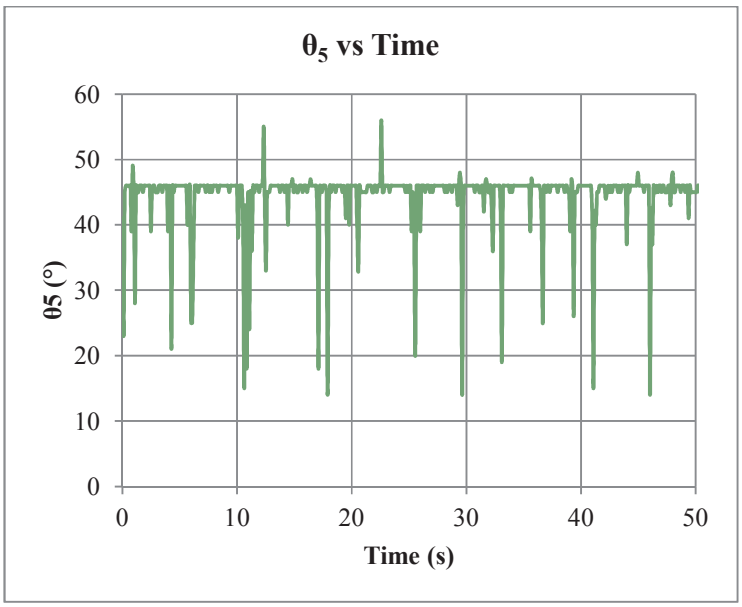

Figure 4-2: Sensor Noise, Servo under Load

The noise plots are shown for θ_5 and the vibration effect can be most prevalent at this joint. The vibration of other motors will cause further vibration on motors closer to the end of the serial manipulator. The filtering and sample averaging used to reduce noise can be problematic if used too generously as discussed in 4.1.2.

The implementation of the RC servo control requires the use of the serial servo controller's (SSC's) own serial communication protocol. The protocol requires start byte, device ID byte, command byte, servo number byte and two data bytes. The data bytes contain the actual position command that must

be in the range of 500 to 5500. Therefore, joint angle position commands need to be mapped to the appropriate SSC command then sent to the SSC using the defined protocol, one servo at a time. The SSC performs auto baud rate detection at start up, but the microcontroller's serial line is not clear during this time, causing the SSC to fail the auto baud rate detection. For this reason an additional reset line (RS) is used, seen in Figure 2-35, so the microcontroller can rest the SSC after some time has passed so auto baud rate detection will work.

Another implementation consideration is quadrature decoding for the track speed encoders and the docking encoder. It was decided to utilize the microcontroller's resources for this and perform decoding in software, using external interrupts. This approach works well for detecting position, speed and direction. Example code for this can be seen in Appendix C.

4.1.2 Cooperative Robot Implementation

An important implementation consideration for cooperative robot control is the reliability of robot-to-robot communication. For this to happen a robot must first send a signal to the PC, the PC interprets it and sends it to the appropriate robot encoded with an appropriate ID. Furthermore, the robots can only send signals one at a time, waiting for the permission character from the PC to do so. Given that this communication is indirect and each robot must wait their turn there is a greater communication delay and the likelihood for unreliable communication increases. To mitigate any delay or reliability problems, the amount of wireless communication is minimized to only what is essential. In cooperative control modes, the only signals being sent from the robots to the PC are their verified operation mode, the signal for robot-to-robot communication, optional signals for troubleshooting and the permission characters to indicate they are done sending. It has been verified that the robot-to-robot communication method is reliable and works for cooperative robot control.

Another cooperative implementation consideration is how to ID the robot modules. In this case it was decided to use fixed robot module IDs, in terms of network nodes and position in the connected chain for cooperative robot functions. This approach is simple and saves time for prototyping. Furthermore, each robot module has slightly different calibrations in terms of how servo motor position relates to pulse commands, control gains and other miscellaneous values. This is handled by using one robot program where the ID number is changed before compiling the code, where the ID changes a number of robot calibration values in the set-up function. It also has to be decided how the different modules should behave with respect to their positions in the connected chain. Ideally they all

behave the same, but the front robot module needs to behave fundamentally different for the cooperative controls. Additionally, the rear robot does not need to send any signals to the PC for robot-to-robot communication. The differences are handled simply by having the fixed robot ID determine if the robot is either the first robot or last robot; where the total number of modules is fixed. This simple technique is sufficient for prototyping, but more sophisticated robot ID assignment would be beneficial for allowing variable robot ID assignment, for when the robots dock.

The manipulator joint position sensor noise mentioned previously, has implications for cooperative robot control. The generous amount of filtering and sample averaging initially used for single robot control caused problems for the cooperative steering where the manipulator is used primarily as a feedback device. If there is too much filtering and/or sample averaging the feedback noise frequency is decreased causing significant feedback variations that change slowly enough for the tracks to respond. Figure 4-3and Figure 4-4 show how the feedback can change for two different cases of a step input for the gap reference command.

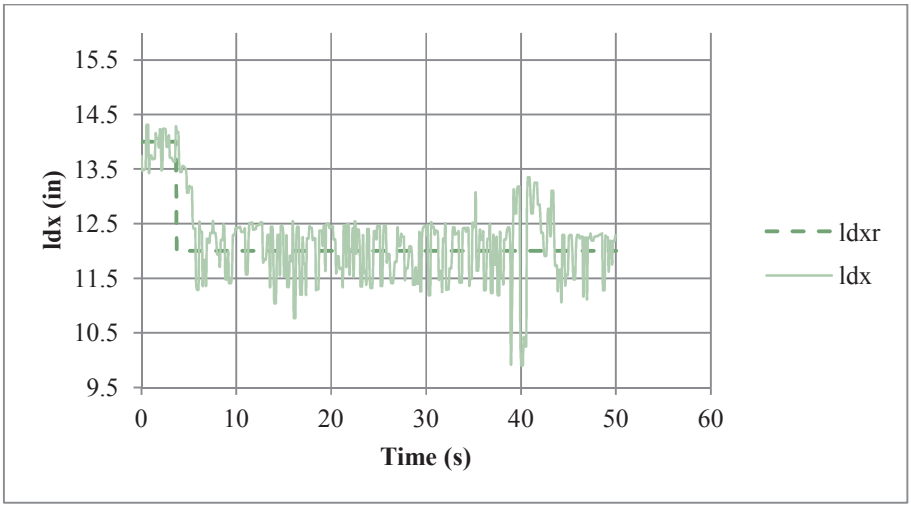

Figure 4-3: Gap Controller Step Response, 0% Filtering and No Averaging

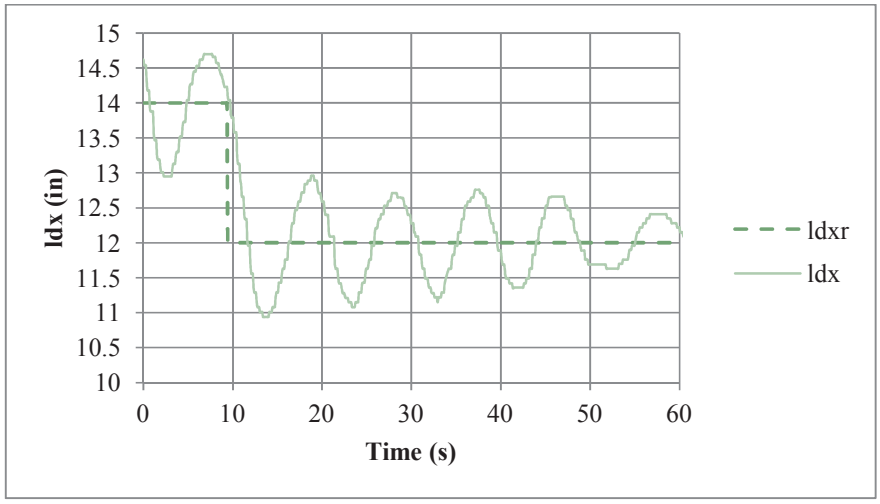

Figure 4-4: Gap Controller Step Response, 75% Filtering and 80 Samples.

In Figure 4-3 it can be seen that there is a high frequency noise, but the variation from the true feedback value does not sustain long enough for the tracks to respond. In Figure 4-4 the lower frequency shows large deviations that sustain long enough, for track speed change to occur. The observation is that unless the gap controller gain is very low the tracks will oscillate, following the noise. Therefore, in this case, the upper limit in gain is quite small and is not large enough for adequate control response. In the higher frequency noise case, the gains can be increased significantly and the track control works well; the modules maintain their gap distance and do not follow the noise. The overall solution to make the cooperative control work as well as single manipulator control, is to reduce the filtering and sample averaging to reasonable values that provide enough smoothing for the manipulator, but allow for larger gap control gains. 20% filtering and 20 samples works well. Also, since the majority of the noise is actually from the manipulators vibrating under load, in the cooperative steering control the noise is much lower. In this case the errors in feedback are mainly due to the joint angle integer rounding, which is compounded in the l_{dx} calculation using three angles in the calculation, seen in equation (3.7). Ultimately, the controls work in both cases, but they could be enhanced by using higher resolution calculations and perhaps different filtering techniques.

With these implementation considerations taken into account, both single and cooperative robot control can be successfully implemented allowing prototype testing of the R2TM3.

4.2 Testing

This section gives the results of the tests performed with the R2TM3. The test methods used depend on the test performed and methods use several guidelines from the set of ASTM standards, resulting from the RoboCup Rescue competitions [48]. They define different obstacle types, test apparatuses and assessment methods [48]. However, given that the ASTM tests are intended for human scale prototypes, opposed to scaled-down prototypes, and are intended for comprehensively complete systems with mapping capabilities and advanced operator interfaces, they are not completely appropriate for the R2TM3 tests. Generally though, a few guidelines have been used from these test standards where appropriate:

- Test apparatus made from wood: common frictional parameters.
- Statistical Reliability R = 80%, Confidence C = 85%. Requirement: 0 failures out of 10 trials or 1 failure out of 20 trials.

Tests where small operator errors or changes in traction can cause large variations in test performance make use of the statistical reliability and confidence assessment from ASTM. Additionally, where appropriate, a maximum or minimum test is performed where only one successful trial is required. In some cases, trial performance is governed by a more deterministic factor such as joint limits and in these cases multiple trials aren't performed. In the tests, trial failure is determined by a success criterion that is based on the apparatus setting, such as ledge height and the performance is usually assessed by a relative measure between the robot module and the test apparatus.

For other tests, the test is a demonstration case and the results are assessed qualitatively; determining whether the system can do the test case or not, with no quantitative measure. This test method is appropriate for unique test cases or cases where the performance is easily observed.

The decided test methods are also consistent with relevant literature. Some authors publish maximum capabilities of first tests and may not go into details on number of successful trials [19], [18], [20] while others may distinguish between a reliable apparatus range and a maximum range [10]. Other tests are simply done as a demonstration case [21]. The test methods used in this work result from an effort to be consistent with common-practice, be quantitative where appropriate and borrow from developed test standards. However, due to the intension of the tests and the low cost

prototype design principle, the right to call mistrials has been reserved, where a trial failure due to severe operator error or equipment failure is considered a mistrial.

In order to provide a comparison between the single robot capabilities and cooperative capabilities, many of the tests have been performed for both cases. This helps reveal the specific advantages to the modular robotic approach.

4.2.1 Ledge Climbing Tests

The rationale of these tests is to assess the climbing ability of the R2TM3 of severe obstacles. The ledge or hurdle is considered a very difficult obstacle type due to the vertical transition. During the climbing of the vertical transition, the normal force due to gravity, which usually provides traction, will become zero. Therefore, in order to make the transition, traction must be maintained in another manner from the climbing method. Therefore, ledge climbing is used as a worst case test.

The test for ledge climbing is set-up with two flat sheets of Canadian softwood (CS) plywood. The transition is created with several softwood lumber frame segments of 38 mm (1.5 in) in thickness. The apparatus test height is adjusted by adding or removing frame segments.

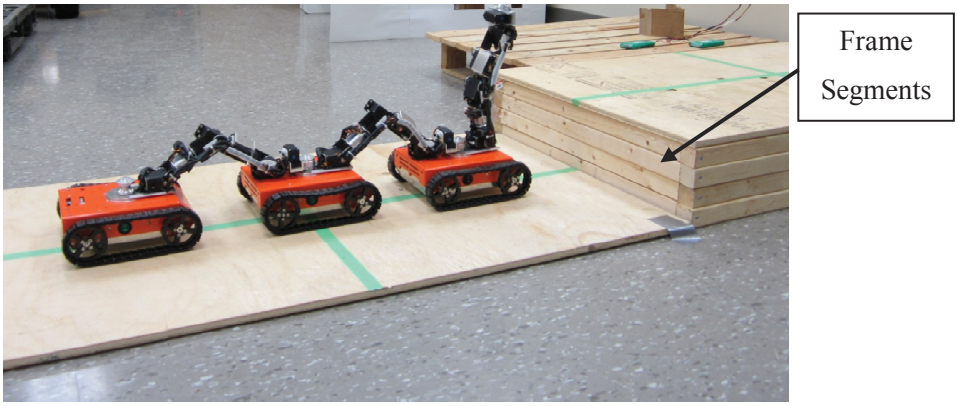

Figure 4-5: Ledge Climbing Test Apparatus

The procedure for each test trial is:

1. Starting on a flat horizontal surface, the robot is placed several inches away from the transition, with a heading normal to the ledge.
2. Robots are driven remotely to climb the ledge. The robots cannot be assisted directly by the operator and a 2 min time limit is put on the trial in the case where the robots struggle to complete the trial.
3. Once the entire robot is on top of the upper flat surface the trial is complete.

Pertinent test parameters are shown in Table 4-1:

Table 4-1: Ledge Climbing Test Parameters

Parameter	Value	Meaning
μ_s	0.96	Coefficient of static friction
l	222 mm (8.75 in)	Robot length
r	38 mm (1.5 in)	Track radius

The climbing performance is assessed with a relative measure between the robot and the ledge height. Considering a standard track base mobile robot climbing a ledge, the track radius compared to the ledge height might seem like a good assessment. However, if the mobile robot possesses any sort of climbing mechanism or method, then the track length may become the limiting vehicle parameter in ledge climbing since the tracks are likely rotated in some fashion to make the transition. Furthermore, the lead radius of the tracks with this sort of climbing technique does not serve to extend the climbing capability of the vehicle. Therefore, the track length excluding the lead radius becomes the best assessment for climbing and can be compared to the ledge height for assessment as:

$$a_l = h/(l - r) \tag{4.1}$$

This assessment provides a means of comparing tracked mobile robots of different sizes with different climbing mechanisms with respect to ledge climbing height h. It gives a measure of the climbed height to usable track length.

To test the modular approach, single robot climbing and cooperative robot climbing are tested. Additionally, for single robot climbing, the test is done both with and without the use of the

manipulator. The manipulator serves to enhance single robot climbing ability. The ledge climbing sequence for single robot climbing with the use of the manipulator is shown in Figure 4-6.

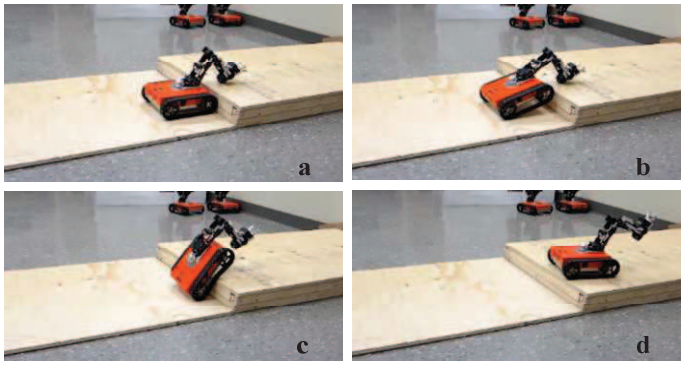

Figure 4-6: Single Robot Climbing Sequence

The ledge climbing sequence for cooperative robot climbing is shown Figure 4-7.

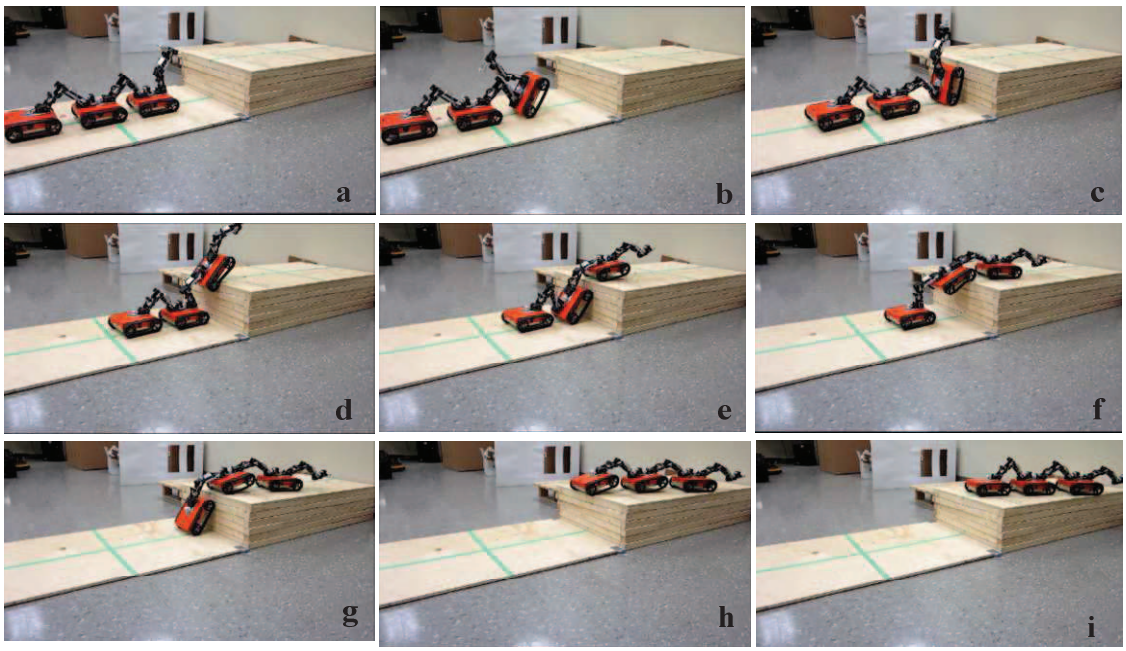

Figure 4-7: Cooperative Climbing Sequence

The single robot climbing sequence is just a matter of driving directly up the rather small ledge, except where the manipulator is used. In that case, the manipulator is used to first prop the track base

up on the ledge, then to stabilize it. It can be seen that the cooperation between the tracks and manipulators, in the cooperative robot test, provides greatly enhanced climbing performance, where the ledge height is actually greater than the module track length.

Table 4-2: Ledge Climbing Test Results

Test Name	Apparatus Setting	Test Type	Assessment	Result
Single Robot (no manipulator)	h = 38 mm (1.5 in)	20 trials	$a_l = 0.21$	14/20 pass rate. Inconsistent.
Single Robot (with manipulator)	h = 38 mm (1.5 in)	10 trials	$a_l = 0.21$	10/10 pass rate. Consistent.
Single Robot (with manipulator)	h = 76 mm (3 in)	10 trials	$a_l = 0.41$	6/10 pass rate. Inconsistent.
Cooperative Robot	h = 152 mm (6 in)	10 trials	$a_l = 0.83$	10/10 pass rate*. Consistent.
Cooperative Robot	h = 229 mm (9 in)	max. (1 trial)	$a_l = 1.24$	Able to climb.

*Two mistrials were called due to a blown fuse and a major operator error.

The test results show that the single robot climbing is quite limited, but is enhanced with the use of the manipulator. Even still, the use of the manipulator is limited and climbing a ledge of 76 mm (3 in) with a single robot is not reliable due mainly to the instability of the robot when propped up on the ledge of that height. It is clear from the test results that the modular, cooperative method greatly enhances the climbing performance, where it achieves an assessment value of 1.24. Climbing was generally inhibited by loss of traction, which is partly due to the simple track design. With a better track design, the climbing performance for the system would likely increase overall.

4.2.2 Corridor Turn-around Tests

The corridor turn-around tests are done to assess and demonstrate the connected manoeuvrability of the modules when steering. It also serves to show the advantage of a modular system, where modularity can be used to effectively change the robot size.

These tests are set-up with the robots driving on a flat surface, the lab floor, in a corridor with a dead end. The apparatus adjustment is the corridor width. The corridor is made by arranging and securing small frame segments on the floor.

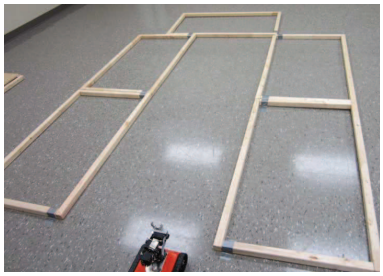

Figure 4-8: Corridor Test Apparatus

The procedure for the corridor test is:

1. The robot is positioned several inches away from the corridor entrance and several inches offset from the right side of the corridor.
2. The robot is then remotely operated to enter the corridor. The robot is steered to turn around once it approaches the dead end. If at any time any part of the robot lifts over the corridor boundary, the trial is failed. The robot may make contact with the corridor edges, but the trial is failed if it gets jammed.
3. The robot is driven out of the corridor and trial is complete once the entire robot has exited the corridor.

Pertinent test parameters are shown in Table 4-3:

Table 4-3: Corridor Climbing Test Parameters

Parameter	Value	Meaning
μ_s	0.64	Coefficient of static friction
d	300 mm (11.77 in)	Robot base diagonal

The corridor steering performance is assessed with a relative measure between the robot diagonal d and the corridor width w:

$$a_c = d/w \tag{4.2}$$

Single robot steering and cooperative robot steering are tested to demonstrate the modular advantage. For the single robot, the test sequence is straight forward, where it simply turns around in a corridor near the end:

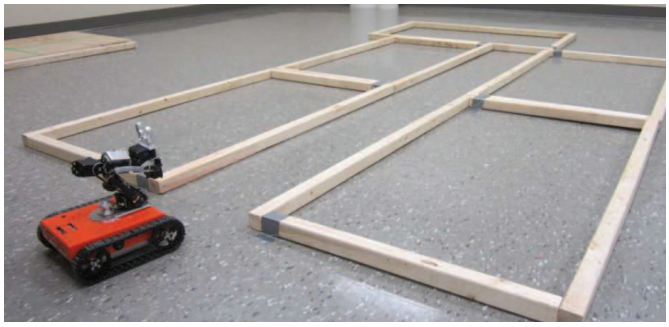

Figure 4-9: Single Robot Corridor Steer

For cooperative steering, more care must be taken when steering the connected chain.

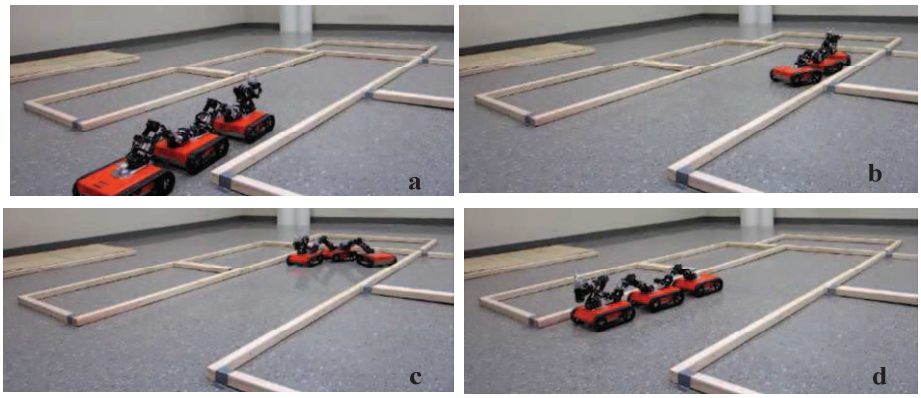

Figure 4-10: Cooperative Steering Sequence

It can be seen that the connected chain of robots can steer quite well and the connected configuration does not tend to limit the turning radius that can be achieved. It can also be seen that for very tight confined spaces a single robot can be used for manoeuvring.

Table 4-4: Corridor Turn-around Test Results

Test Name	Apparatus Setting	Test Type	Assessment	Result
Single Robot Corridor Turn	w = 305 mm (12 in)	10 trials	$a_c = 0.98$	10/10 pass rate. Consistent.
Cooperative Corridor Turn	w = 864 mm (34 in)	10 trials	$a_c = 0.35$	10/10 pass rate. Consistent.
Cooperative Corridor Turn	w= 813 mm (32 in)	min. (1 trial)	$a_c = 0.37$	Able to turn.

The test results show that the single robot is quite manoeuvrable and that the cooperative chain is manoeuvrable considering the fact that the modules are connected to one another. These test results indicate that the connection method and the modular approach yield a system with great manoeuvrability characteristics.

4.2.3 Cooperative Shifting Test

As a special case of connected manoeuvrability, cooperative shifting was tested. Since the performance of this test is deterministic and the apparatus corridor width just needs to be wide enough to allow the robots to travel through at 305 mm (12 in), this test was done as a demonstration. The demonstration shows the convenient use of shifting to enhance overall manoeuvrability.

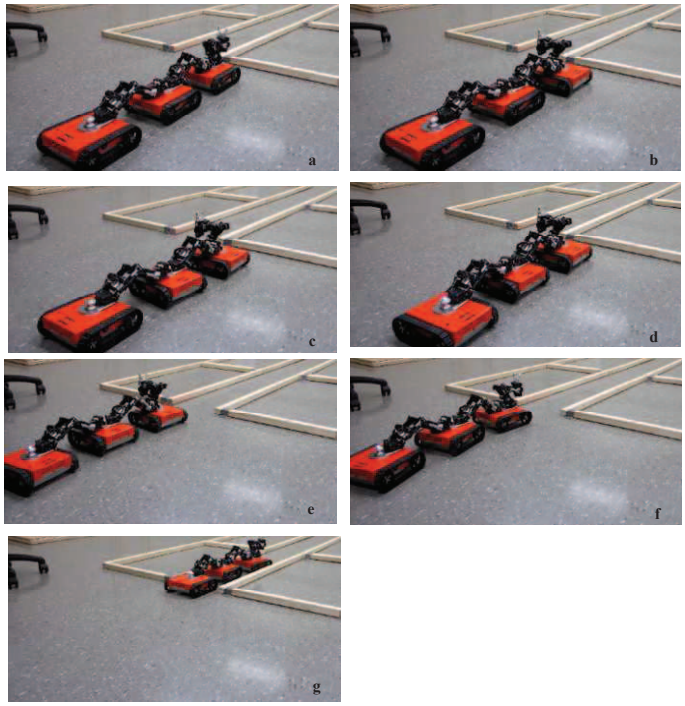

Figure 4-11: Cooperative Shifting Sequence

The cooperative shifting strategy is successful in providing a convenient way of achieving lateral movement. In this case the connected robot chain approaches an obstacle to the right of the corridor entrance. Instead of having to swerve and steer to re-position the chain at the corridor entrance

shifting is used. In the sequence it can be seen that at first the modules are turned individually, but after they are turned together, showing that the manoeuvre can be performed either way.

4.2.4 Slope Crossing Tests

Slope climbing and crossing tests were done to assess the steepness of a slope the robot can climb and steer on, assessed by the slope angle. The test was also done to test whether there is any quantifiable advantage to cooperative slope crossing. With the R2TM3 robot design, the COM is low and the limiting factor when climbing a slope is traction. As mentioned previously, skid-steering involves slipping and skidding to turn, therefore when steering on a slope the robot risks a significant drop in friction and will slide down the slope. Consequently, tests were done to determine the threshold of climbing a slope and traversing a slope by steering on it.

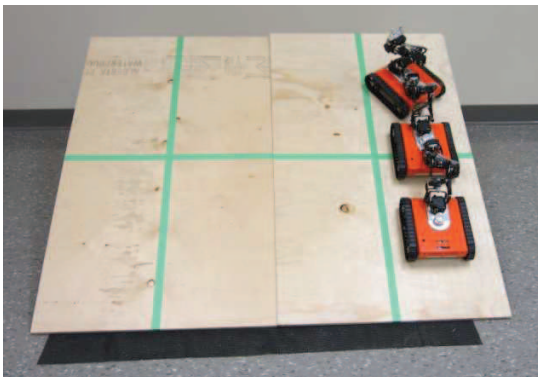

Figure 4-12: Slope Crossing Test Apparatus

The horizontal line marked across the middle of the slope in Figure 4-12 was used to determine a pass or fail with regards to the single robot sliding down the slope when steering. Some sliding is inevitable, so the line was used as a consistent cut-off point for assessment.

It was suspected, that utilizing the shifting technique for cooperative steering on a slope would enhance the slope the robots could steer on. By turning one robot at a time in the connected chain, with the manipulators on, the other stationary robots would serve to anchor the group of robots and hold the one turning on the slope. If this were true, then the cooperative approach would allow the robots to steer at close to the same slope angle that they can climb up. Unfortunately, this was not realized in testing. The forces due to skid-steering on the slope are very unevenly distributed under the robot tracks, which creates a large tendency for the turning robot to steer off center and drift.

These forces were strong enough to overcome the connected manipulators, which would then allow the module to slide down the slope, bringing the group with it. The present hypothesis is that if the manipulator base joint motor was stronger then perhaps the cooperative shifting method would yield measurable advantages over single robot slope steering. However, there are presently no test results that show this.

The testing did serve to provide an assessment of the slope climbing and crossing ability of the robots.

Table 4-5: Slope Crossing Test Results

Test Name	Apparatus Setting	Test Type	Assessment	Result
Single Robot Slope Crossing	Climb: 36.5° Steer: 25°	10 trials 10 trials	-	10/10 pass rate. Consistent.
Single Robot Slope Crossing	Climb: 39°	Max. (1 trial)	-	Able to climb.
Cooperative Corridor Turn	Steer: 30°	-	-	Test failed.

From the results it can be seen that there is a large difference between the slope the robot can drive up and the slope it can steer on without significant sliding. Perhaps the simple track design aggravates this behavior.

4.2.5 Docking Tests

Several different docking tests were done as demonstration test cases, showing the ability of the system to dock in a variety of terrain conditions, simulated by robot offsets and misalignments. In each case the success criteria for docking was to, first dock the robots, but then to make a connected formation, such as being connected and straight on the same plane. The tested cases are shown in the flowing sequences.

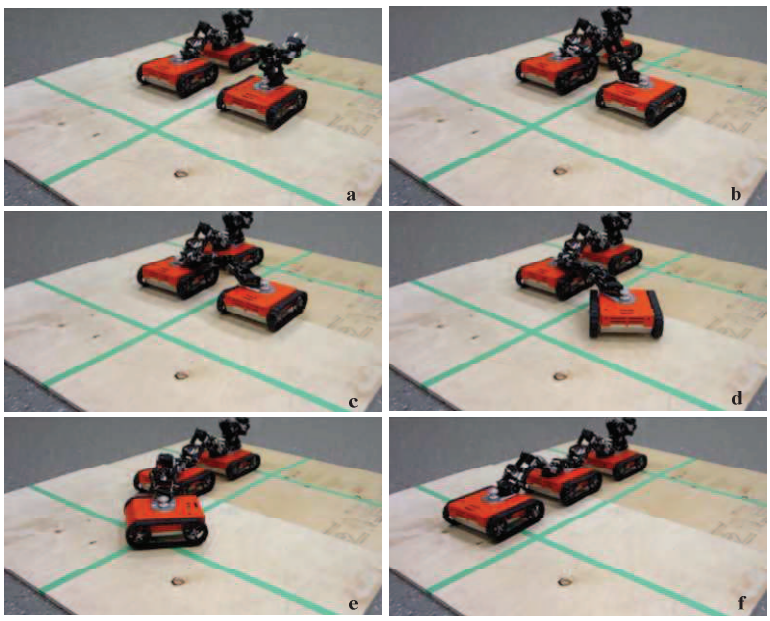

Figure 4-13: Successful Docking, Maximum Lateral Offset (14 in)

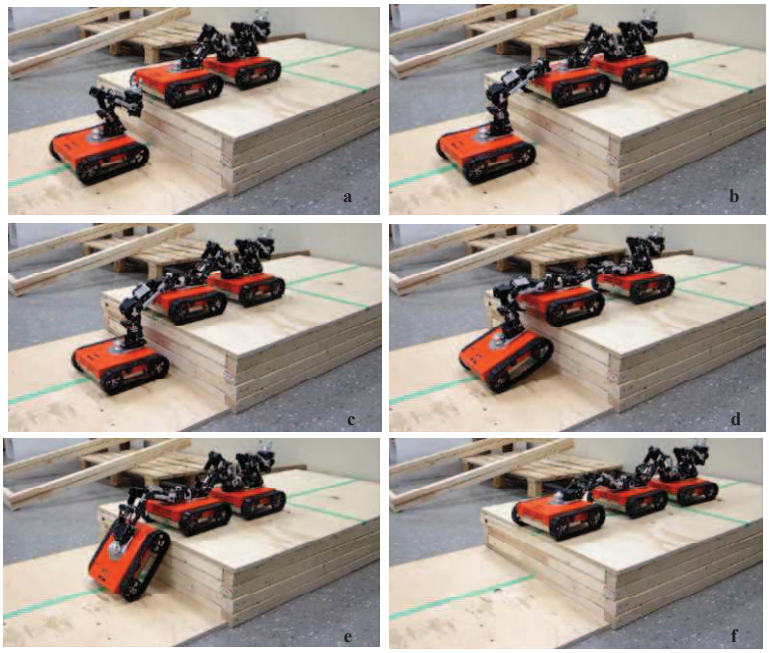

Figure 4-14: Successful Docking, Maximum Vertical Offset (6 in)

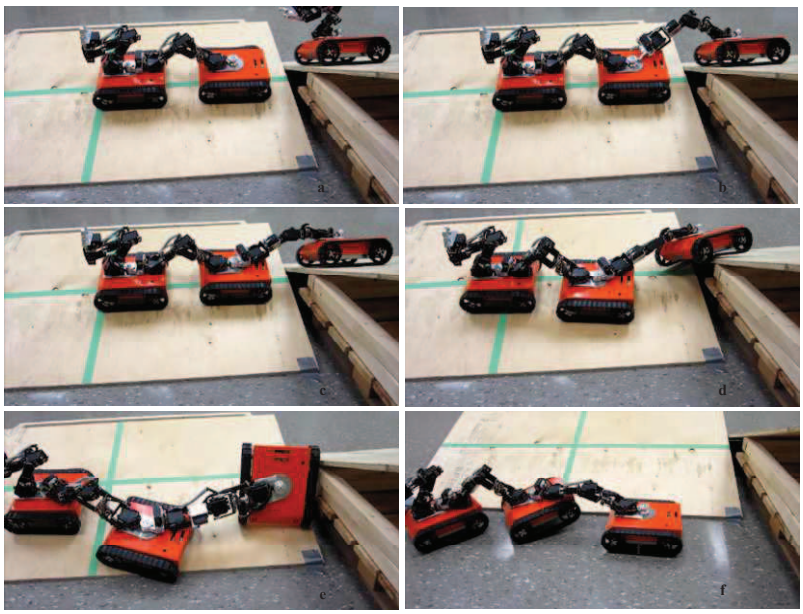

Figure 4-15: Successful Docking, Angular Offset (50°)

In each of the docking tests, the docking and making the final formation was successful. In some cases, this docking approach can be looked at as a robot retrieval technique. In the sequence shown in Figure 4-15 it can be seen that the passive stabilizing effect of the connected manipulators prevented the robot from tipping over in this rather awkward docking condition. The tests show that the system can handle severe terrain conditions in terms of docking the robots, which is essential for robot cooperation in such terrains.

4.2.6 Robot Retrieval via Docking

Considering docking as a robot retrieval technique resulted in another test case in which the one robot is separated from the group by some barrier, simulating debris.

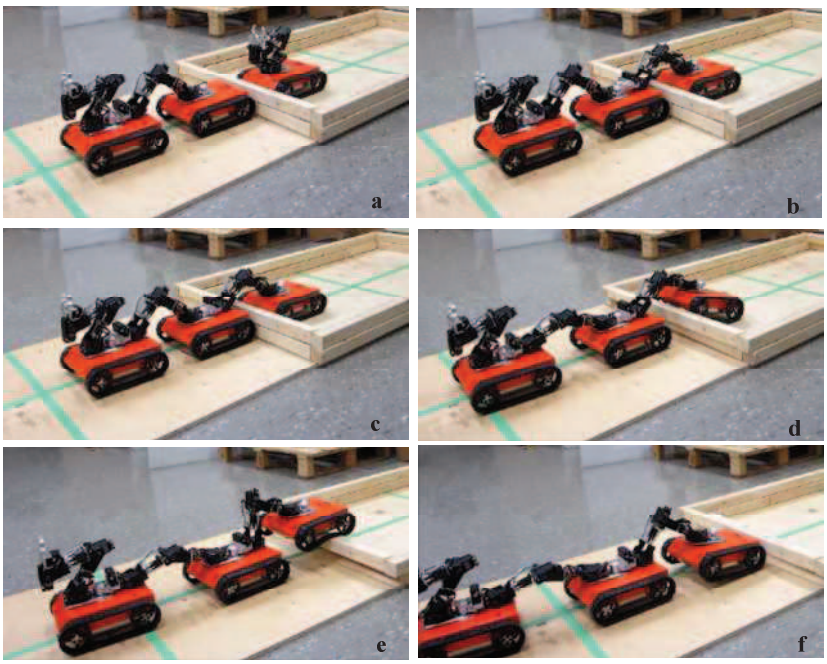

Figure 4-16: Robot Retrieval Sequence via Docking

The sequence shows another test case in which successful docking was achieved as a means of robot retrieval.

4.2.7 Low Friction Surface Crossing Tests

Another test performed to demonstrate the advantage of the modular robot cooperation is the low friction surface (LFS) crossing. In this test a low friction surface on a slight slope is to be crossed. In this situation a single robot cannot cross the surface due to insufficient traction. However, when the robots are used cooperatively they may cross a surface of a certain maximum length. As long as some part of the connected rigid chain has traction, the entire chain can be pushed or pulled along. This test is intended to show the inherent traction advantage of having several modules connected in unstructured environments; a much greater chance of maintaining enough traction to continue driving.

Figure 4-17: Unsuccessful Single Robot LFS Crossing

Figure 4-18: Successful Cooperative LFS Crossing (33.5 in)

Pertinent parameters for this test are:

Table 4-6: LFS Crossing Test Parameters

Parameter	Value	Meaning
μ_s	0.2	Coefficient of static friction
Slope Angle	15°	-
l_s	33.5 in	Surface Length
l_c	36.75 in	Max. connected length

From the test results it is clear the connected chain of robots has an advantage over a single robot in terms of traction.

4.2.8 Light Manipulation

Due to the 5-DOF manipulator, the robot modules are capable of manipulation with a rather good work envelope for reaching small objects. A test case was done to demonstrate the capability shown in the sequence:

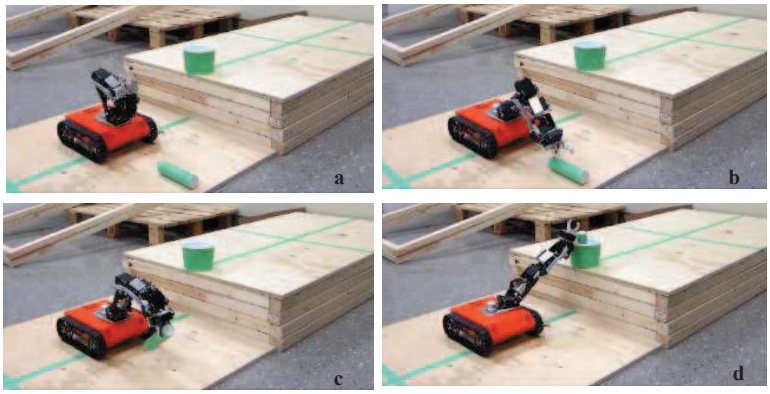

Figure 4-19: Light Manipulation

There may also be instances where an object is out of reach of the single robot's manipulator. Utilizing the cooperative pitching approach already shown, the robots may cooperatively enhance the manipulation capabilities in terms of reach:

Figure 4-20: Object Out of Reach for Single Robot

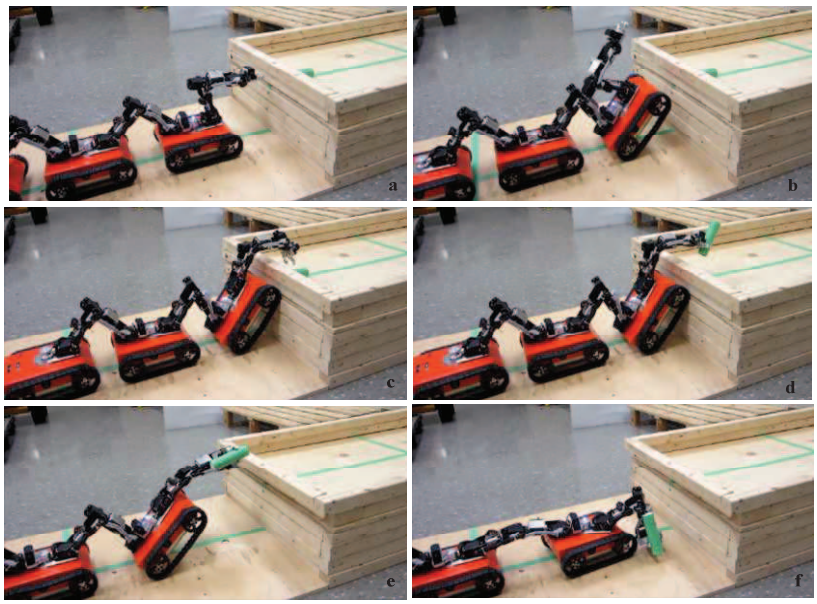

Figure 4-21: Cooperative Manipulation

From the demonstrated test results for manipulation it can be seen that single robot manipulation is possible for cases that require a high range of motion. Furthermore, robot cooperation can be used to enhance this ability.

4.2.9 Discussion of Results

The test results for the R2TM3 system show that the system successfully enhances mobility through modularity and is highly manoeuvrable. With the cooperative approach of actively using the manipulators and tracks, the cooperative climbing yields climbable object heights greater than the length of the single module. Furthermore, due to the use of the 5-DOF manipulator, docking can be achieved in severe terrain conditions, which can also be viewed as a type of robot retrieval.

The R2TM3's test results also show that it is still highly manoeuvrable, even when connected. The robots can steer in a tight arch formation and they can shift to re-position. Moreover, for even tighter manoeuvring tasks, the modularity allows for the system to undock and steer as a single robot, achieving a type of system scaling effect.

Using relative assessment measures, the results can be interpreted for the scaled-down robot prototype and can be extrapolated as a measure of what a larger scale prototype might achieve. The rather simple track design proved to be a limiting factor in climbing for several reasons:

1. Lack of track grousers makes maintaining traction harder. Higher sensitivity to surface discontinuities.
2. Lack of track tension in the centre portion reduces track torque output. If tracks are deflected significantly then they cause resistance to the track drives, reducing available output for driving. Lack of track tension in the centre portion makes the tracks more sensitive to slippage; the normal force under the tracks is very non-uniform.

In spite of some inhibiting factors, the scaled concept prototype testing shows that the R2TM3 system can perform well regarding its intended functions.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis presents the conceptualization, development, implementation of controls and testing of the R2TM3 for the purpose of enhancing mobility in unstructured environments. The test results demonstrate that this concept has merit. Specifically, through a mobile modular approach, severe obstacle climbing can be achieved as well as a high level manoeuvrability. Through the use of manipulator and track cooperation, the robot modules can be pitched to manage steep ledges and can be extended up the ledge achieving heights greater than the length of the module itself. This cooperation approach allows for a light serial manipulator with limited force output to move a robot module for severe obstacle climbing, which would otherwise not be possible.

By using a 5-DOF manipulator as the docking mechanism and the on/off manipulator control, module docking in severe terrain conditions can be achieved, which is the essential first step in cooperative obstacle managements if the modules are not already docked. In fact, the allowable docking misalignments and offsets are such that robot module recovery may be performed, where the difference between module docking and module recovery may be arbitrary.

Due to the passive docking joint, the joint's position and use of the specific manipulator configuration as the docking mechanism, the system is highly manoeuvrable when connected. Through steering or shifting, the connected chain of modules can move around obstacles or in

confined spaces rather easily without having to undock. Furthermore, if manoeuvring through extremely tight spaces is required, modularity can be utilized to enable a single smaller skid-steered robot module to be used in such spaces. Mobility is further enhanced through modularity due to the enhanced traction of the connected modules and the stabilizing effect of the connected manipulators.

In this homogeneous modular system each module is also highly functional due to self mobility and manipulation capabilities with their 5-DOF manipulators, possessing the same capabilities of any skid-steered mobile robot with an onboard manipulator. Each module contains its own control hardware and power source. Therefore, this modular approach yields increased robustness through module redundancy.

Through the developed proof of concept prototype, cooperative control approaches and testing, this work has shown that the R2TM3 concept has significant merit for enhancing mobility in unstructured environments.

5.2 Future Work and Improvements

The system merits have been described and demonstrated, however several features have not been included in the proof of concept design and some disadvantages have been identified. Topics for future work and improvements will be given, where several points have already been mentioned in the body of this thesis.

Improved track design with tension and grousers:

Tracks equipped with grousers and a tensioning mechanism would greatly enhance traction and would allow the tracks to make contact on an obstacle edge without causing excessive track deflection, loading the drive motors. The prototype tracks have been coated with rubber, but further redesign would go a long way in climbing performance.

Docking joint lock:

It was suspected during the concept development, that a docking joint lock might be needed to keep the modules in their desired alignment during climbing. This pertains mainly to the front module where its own manipulator is not connected to another module. For other modules, their manipulator's base joint constrains them. It was found during testing that if one side of the front module lost traction, then it would rotate about its own docking joint and the test could fail. For this

reason, steering control was added to help compensate for this. Ultimately a docking joint lock would be ideal. This was not implemented due to limited time, but would greatly enhance the climbing performance.

Contact bumpers:

It can be seen in the cooperative climbing control that it is assumed that all modules rotate about point *B*. With the control approach and no knowledge of the module contact conditions, an assumption has to be made. A rather simple way to detect the contact conditions is to use contact bumpers, where it can be sensed if a given module is in contact on the rear, front, both or none of its ends. This approach allows for the contact to be sensed while the modules are pitched on their ends, but contact between the bumpers would have to be assumed to be in the center of the tracks. Another challenge with the bumpers is to make sure they are robust enough to withstand the side forces during skid-steering.

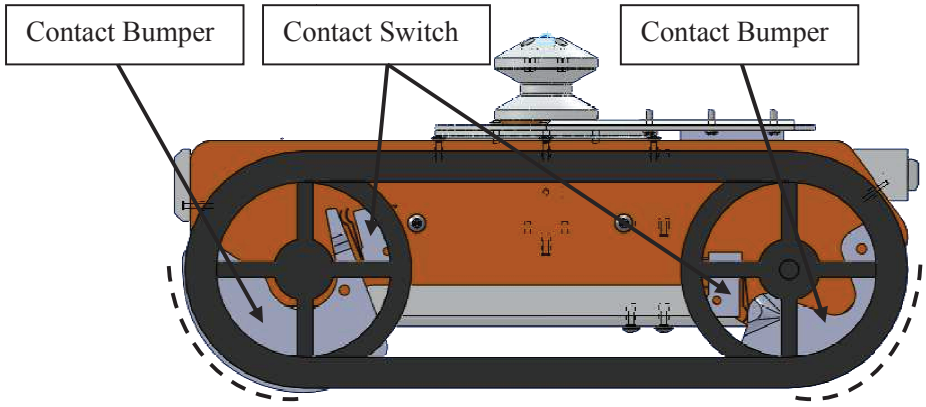

Figure 5-1: Contact Bumpers

The contact bumper idea is likely the simplest way to implement this functionality on the current prototype, but other force sensing methods could be utilized that would allow estimation of contact conditions.

More generalized and potentially more autonomous control:

An area where significant work could be done would be to generalize the control approaches already used. Using some method to provide a basic map of the ground or obstacle could result in more autonomy being possible. Furthermore, more generalized control could allow for higher obstacles to be climbed since they could move more simultaneously. For example, when the front module makes the ledge transition and begins the reach the full extension of the manipulator, the

second module could begin to pitch simultaneously, lifting its manipulator and extending the reachable height of the front module. With more modules even greater heights could be possible.

Furthermore, generalizing the control to 3D, where general 3D obstacles could be managed, would bring the system capabilities to the point where it could be used in almost any 3D terrain condition. The system manoeuvring and climbing could then be more seamless. Also, through dynamic modeling and more sophisticated controls, the system performance could be enhanced, since the system dynamics can actually assist in cooperative functions [32].

Method for active tip-over recovery or module symmetry:

Probably the biggest disadvantage of this system is the asymmetry of the modules regarding tip-over, which necessitates an active tip-over recovery strategy. This problem has been mitigated by the light manipulator design, but tip-over could still occur. The modular approach makes active tip-over recovery easier since it could be done cooperatively, where one or more nearby modules would use their manipulators to rotate another module upright. Nonetheless, tip-over could still be problematic and module symmetry is an ideal solution; the module can function the same way when tipped over. With the manipulator base joint and docking joint in this system, achieving this kind symmetry could prove challenging. It would require further integration of the manipulator and track base and possibly another DOF between them. An additional DOF might alternatively make a single module tip-over recovery strategy the easiest solution. Through further track base and manipulator integration and clever design, effective tip-over recovery or module symmetry could be achieved, while maintaining the system's docking capabilities.

Modules bumpers:

It was recognized in the concept development phase, that the gap spanning ability of the system would be limited roughly to gaps the width of the module track length. In order for the light serial manipulator to move or pitch a module, the module needs to be in contact with the ground at least at one end, so that the manipulator torque limits are not overcome. It was recognized that if the modules had bumpers on the front and back that a form of secondary docking could be accomplished. By pulling the modules in close so that the bumpers make contact, the forces and moments from gap spanning a module could be reacted and shared with the manipulator.

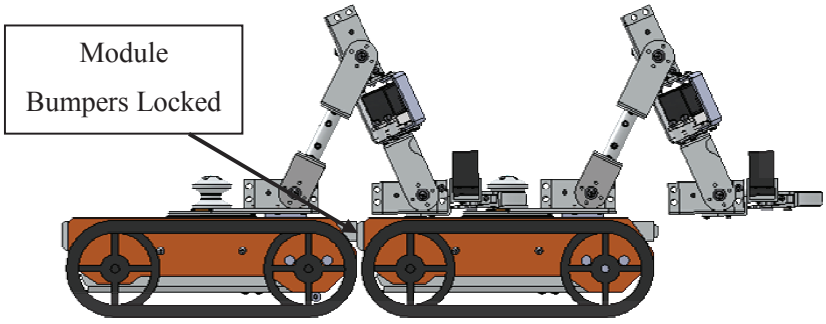

Figure 5-2: Module Bumpers

Since gap spanning was not a priority for testing and time was limited, the bumpers were not implemented or tested. If this idea works, then the manipulator and bumper connection would create a rigid set of modules that could gap span more extreme gap widths, limited by the total COM of the group. With three or more modules, the gap spanning ability could be impressive.

Next generation prototype:

If the aforementioned items for future work were to be worked on, it would likely make sense to develop a next generation prototype, intended for field use with possibly more modules. Incorporating the improvements already mentioned as well as making the design more robust would result in a rather impressive device for field testing. Additional computational power and sensory could lead to a system ready for application.

Bibliography

[1] M. R. Blackburn, H. R. Everett, and R. T. Laird, "After Action Report to the Joint Program Office: Center for Robotic Assisted Search and Rescue (CRASAR) Related Efforts at the World Trade Center," SPAWAR Systems Center, San Diego, Technical Document 3141, 2002.

[2] R. Siegwart and I.R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Cambridge, Massachusetts: The MIT Press, 2004.

[3] Public Safety Canada. (2012, Mar.) Canadian Urban Search and Rescue (USAR) classification guide. [Online]. http://www.publicsafety.gc.ca/prg/em/usar/usar-guide-eng.aspx

[4] The City of New York. (2012) 9/11 Health: Rescue and Recovery Workers, What We Know From the Research. [Online]. http://www.nyc.gov/html/doh/wtc/html/rescue/know.shtml

[5] National Post. (2012, June) Crews in Elliot Lake call off search for mall collapse victims because building is 'totally unstable'. [Online]. http://news.nationalpost.com/2012/06/25/crews-in-elliot-lake-call-off-search-for-mall-collapse-victims-because-building-is-unstable/

[6] A. Wolf, H.H. Choset, H.B. Brown Jr., and R.W. Casciola, "Design and Control of a Mobile Hyper-Redundant Urban Search and Rescue Robot,"," *Advanced Robotics*, vol. 19, no. 3, pp. 221-248, 2005.

[7] P. Scerri et al., "Towards an Understanding of the Impact of Autonomous Path Planning on Victim Search in USAR," in *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, 2010, pp. 383-388.

[8] Q. Zhang et al., "Mission-Oriented Design: A Fully Autonomous Mobile Urban Robot," in *2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Salt Lake City, 2010, pp. 261-266.

[9] M. Onosato et al., "Disaster Information Gathering Aerial Robot Systems," in *Rescue Robotics: DDT Project on Robotics and Systems for Urban Search and Rescue*, S. Tadokoro, Ed. London: Springer, 2009, ch. 3, pp. 33-55.

[10] P. Ben-Tzvi, "Experimental Validation and Field Performance Metrics of a Hybrid Mobile Robot Mechanism," *Journal of Field Robotics*, vol. 27, no. 3, pp. 250-267, Feb. 2010.

[11] M. Yim, P. White, M. Park, and J. Sastra, "Modular Self-Reconfigurable Robots," *Encyclopedia of Complexity and System Science*, pp. 19-32, 2009.

[12] D. J. Christensen, "Elements of Autonomous Self-Reconfigurable Robots," Ph.D. Thesis, The Maersk Mc-Kinney Moller Institute, Univ. of Southern Denmark, Odense, Denmark, 2008.

[13] A Lyder, R. F.M. Garcia, and K. Stoy, "Genderless Connection Mechanism for Modular Robots Introducing Torque Transmission Between Modules," in *ICRA 2010 Workshop "Modular Robots: Sate of the Art"*, 2010, pp. 77-81.

[14] J.C. Larsen, R.F.M. Garcia, and K. Stoy, "Increased Versatility of Modular Robots through Layered Heterogeneity," in *ICRA 2010 Workshop "Modular Robots: State of the Art"*, 2010, pp. 24-29.

[15] R. O'Grady, A.L. Christensen, and M. Dorigo, "Autonomous Reconfiguration in a Self-assembling Multi-robot System," in *ANTS 2008, Sixth International Conference on Ant Colony Optimization and Swarm Intelligence*, Brussels, 2008, pp. 259-266.

[16] S. Tadokoro, "Earthquake Disaster and Expectations of Robotics," in *Rescue Robotics: DDT Project on Robotics and Systems for Urban Search and Rescue*, S. Tadokoro, Ed. London: Springer, 2009, ch. 1, pp. 1-16.

[17] M. Fumitoshi et al., "On-Rubble Robot Systems for the DDT Project," in *Rescue Robotics: DDT Project on Robotics and Systems for Urban Search and Rescue*, S. Tadokoro, Ed. London: Springer, 2009, ch. 6, pp. 105-129.

[18] B. Li et al., "AMOEBA-I: A Shape-Shifting Modular Robot for Urban Search and Rescue," *Advanced Robotics*, vol. 23, pp. 1057-1083, 2009.

[19] M. Guarnieri et al., "HELIOS IX Tracked Vehicle for Urban Search and Rescue Operations: Mechanical Design and First Tests," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, 2008, pp. 1612-1617.

[20] W. Wang, H. Zhang, G. Zong, and Z. Deng, "A Reconfigurable Mobile Robot System Based on Parallel Mechanism," in *Parallel Manipulators, Towards New Applications*, H. Wu, Ed. Vienna, Austria: I-Tech Education and Publishing, 2008, pp. 347-362.

[21] W. Wang, H. Zhang, W. Yu, and J. Zhang, "Docking Manipulator for a Reconfigurable Mobile Robot System," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, 2009, pp. 1697-1702.

[22] RoboCup Rescue. (2008) RoboCup Recue: About Us. [Online]. http://www.robocuprescue.org/about.html

[23] Centre for Robot-Assisted Search and Rescue (CRASAR). (2012) About CRASAR. [Online]. http://crasar.org/about/

[24] iRobot, iRobot 510 Packbot, 2011, Technical Specification.

[25] R. Stopforth, G. Bright, and R. Harley, "Performance of the Improvements of the CAESAR Robot," *International Journal of Advanced Robotic Systems*, vol. 7, no. 3, pp. 217-226, 2010.

[26] A. Kamimura and H. Kurokawa, "High-Step Climbing by a Crawler Robot Dir-2 - Realization of Automatic Climbing Motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, 2009, pp. 618-624.

[27] Y. Liu and G. Liu, "Interaction Analysis and Online Tip-Over Avoidance for a Reconfigurable Tracked Mobile Manipulator Negotiating Slopes," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 4, pp. 623-635, Aug. 2012.

[28] A. Ferworn et al., "Expedients for Marsupial Operations of USAR Robots," in *IEEE International Workshop on Safety, Security and Rescue Robotics*, Gaithersburg, 2006.

[29] H.B. Brown Jr., J.M.V. Weghe, C.A. Bererton, and P.K. Khosla, "Millibot Trains for Enhanced Mobility," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 452-461, Dec. 2002.

[30] D.M. Hensinger, G.A. Johnston, E.M. Hinman-Sweeney, J. Feddema, and S. Eskridge, "Self-Reconfigurable Robots," Sandia National Laboratories, Albuquerque, Tech. Rep. SAND2002-3237, 2002.

[31] S. Hirose, S. Takaya, and E.F. Fukushima, "Proposal for Cooperative Robot "Gunryu" Composed of Autonomous Segments," *Robotics and Autonomous Systems*, vol. 17, pp. 107-118, 1996.

[32] A.D Deshpande and J.E Luntz, "A methodology for design and analysis of cooperative behaviors with mobile robots," *Auton Robot*, vol. 27, pp. 261-276, July 2009.

[33] M. Guarnieri, I. Takao, E.F. Fukushima, and S. Hirose, "HELIOS VIII Search and Rescue Robot: Design of an Adaptive Gripper and System Improvements," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, 2007, pp. 1775-1780.

[34] M.W. Spong, S. Hutchinson, and W. Vidyasagar, *Robot Modeling and Control*, 1st ed. Danvers, MA: John Wiley and Sons, Inc., 2005.

[35] L. He, S. Phillips, S. Waslander, and W. Melek, "Task Based Pose Optimization of Modular Mobile Manipulators," in *ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis (ESDA)*, Nantes, 2012.

[36] S. Phillips, V.S. Muniappan, S.L. Waslander, H. Karbasi, and J.P. Huissoon, "Modular Mobile Robotics: Obstacle Management through Reconfiguration," in *23rd Canadian Congress of Applied Mechanics*, Vancouver, 2011, pp. 924-927.

[37] O. Khatib, "CS223A — Introduction to Robotics," Stanford Engineering Everywhere [Online], 2008.

[38] R.L. Norton, *Design of Machinery*, 4th ed. New York, NY: McGraw-Hill Companies, Inc., 2008.

[39] J.D. Warren, J. Adams, and H. Molle, *Arduino Robotics*. New York, NY: Springer, 2011.

[40] T. Bräunl, *Embedded Robotics*, 3rd ed. Berlin: Springer-Verlag, 2008.

[41] W. Yu, E. Collins, and O. Chuy, "Dynamic Modeling and Power Modeling of Robotic Skid-Steered Wheeled Vehicles," in *Mobile Robots - Current Trends*, Z. Gacovski, Ed.: InTech, 2011, ch. 14, pp. 291 - 317.

[42] J.Y. Wong, *Theory of Ground Vehicles*, 3rd ed. New York, NY: John Wiley & Sons, 2001.

[43] H. Wang et al., "Modeling and Motion Stability Analysis of Skid-Steered Mobile Robots," in *IEEE International Conference on Robotics and Automation ICRA*, 2009, pp. 4112-4117.

[44] K. Ueda, M. Guarnieri, R. Hodoshima, E. Fukushima, and S. Hirose, "Improvement of the Remote Operability for the Arm-Equipped Tracked Vehicle HELlOS IX," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, 2010, pp. 363-369.

[45] O Amidi, "Integrated Mobile Robot Control," The Robotics Institute, Carnegie Mellon University, Pittsburgh, Technical CMU-RI-TR-90-17, 1990.

[46] M Lundgren, "Path Tracking and Obstacle Avoidance for a Miniature Robot," Umeå University, Umeå, Master Thesis 2003.

[47] Arduino. (2012) Playground. [Online]. http://arduino.cc/playground/

[48] ASTM, "Standard Test Method for Evaluating Emergency Response Robot Capabilities: Mobility: Confined Area Obstacles: Hurdles," ASTM, Test Standard E2802 – 11, 2012.

Appendix A

Controller Specifications

Main Microcontroller (LeafLabs Maple):

PCB Size: 2.05” x 2.1”

Clock Speed: 72 MHz

Operating Voltage: 3.3 V

Input Voltage: 3.0-12 V

Digital I/O: 39

Analog Input Pins: 16

Flash Memory: 128 kB

SRAM: 20 kB

64 Channel nested vector interrupt handler (including external interrupt on GPIOs)

Integrated SPI/I2C and 7 Channels of Direct Memory Access (DMA)

Current Supply: 800 mA @ 3.3 V

Low Power and Sleep Current: < 500 μA

Serial Servo Controller (Pololu SSC):

PCB Size: 0.91” x 0.91”

Number of servo ports: 8

Pulse width range: 0.25-2.75 ms

Resolution: 0.5 μs (0.05 degree)

Supply voltage: 5-16 V

I/O voltage: 0 and 5 V

Baud rate: 1200 – 38400 (auto detect)

Current consumption: 5 mA (average)

Dual Motor Driver (DFRobot Arduino Motor Shield):

Motor Drive: Two way 7-12 V

Current Output (each channel): 2 A

Pins for motor drive: 5, 6, 7 and 8

Speed Control: PWM:

Advanced control supported: PLL (phased locked loop)

Appendix B

Protocol Characters

Description	Command or Signal	Assigned Character	Data Type
Robot 1	C	$	Character Only
Robot 2	C	%	Character Only
Robot 3	C	&	Character Only
All Robots	C	#	Character Only
Robot 1 Can Send	C/S	:	Character Only
Robot 2 Can Send	C/S	*	Character Only
Robot 3 Can Send	C/S	-	Character Only
End of Command/Signal	C/S	E	Character Only
Speed	C/S	S	Integer
Yaw Rate	C/S	H	Integer
Left Track Speed	C/S	(	Integer
Right Track Speed	C/S	)	Integer
Manipulator Position x	C	X	Integer
Manipulator Position z	C	Z	Integer
Manipulator Orien. Roll	C	R	Integer
Manipulator Orien. Pitch	C	P	Integer
Manipulator Speed x	C	J	Integer
Manipulator Speed z	C	L	Integer
Manipulator Speed Roll	C	M	Integer
Manipulator Speed Pitch	C	N	Integer
Gripper	C/S	G	Integer

Table A-1: Protocol Characters 1

Servos On/Off	C/S	F	Integer
Joint 1 Position	S	Q	Integer
Joint 2 Position	S	V	Integer
Joint 3 Position	S	D	Integer
Joint 4 Position	S	B	Integer
Joint 5 Position	S	W	Integer
Tilt 1 (Base Pitch)	S	T	Integer
Tilt 2 (Base Roll)	S	U	Integer
Manipulator Current	S	I	Integer
Base Joint Current	S	C	Integer
Contact Switch 1	S	<	Integer
Contact Switch 2	S	>	Integer
Contact Switch 3	S	!	Integer
Contact Switch 4	S	\	Integer
Mode Number	C	A	Integer
Gap Change	C	x	Integer
Docking Angle	S	d	Integer
Leading Robot Dock Angle	S	l	Integer
Path Radius	C	r	Integer
Temp test value	C/S	t	float
Temp test value 2	C/S	b	float
Leading Robot Pitch Angle	S	p	Integer
Coop Translate Rate	C	k	Float
Coop Translate Rate Hor	C	h	Float
Pitch Rate	C	j	Float
Robot to Use	C	g	Integer
Speed and Climb	C	v	Integer

Table A-2: Protocol Characters 2

Appendix C

Program Sections

Main Loop:

```
void loop() {
 receiveCommands();
 getSensorFeedback();
 trackSpeedInput();  //speed feedback from encoders
 handleSpeedHeading(); //transformation to track speeds
 trackPIDs(); //track speed PIDs
 handleTrackSpeed(); //track PWM and direction
 if(opMode == 0){
  gapInitialized = false;
  dockInitialized = false;
  coopClimbManipInit = false;
 }
 if(opMode == 1){
  handleManipulatorCommands();
  handleManipOnOff();
  gapInitialized = false;
  dockInitialized = false;
  coopClimbManipInit = false;
 }
 if(opMode == 2){
```

```
 handleManipulatorCommands();
 handleManipOnOff();
 gapInitialized = false;
 dockInitialized = false;
 cooperativeClimbing();
}
if(opMode == 3){
 cooperativeSteering();
 handleManipulatorCommands();
 handleManipOnOff();
 coopClimbManipInit = false;
}
if(opMode == 4){
 cooperativeShifting();
 handleManipulatorCommands();
 handleManipOnOff();
 gapInitialized = false;
 coopClimbManipInit = false;
}
if(opMode != 4){
 slewOnOff = true;
}
sendManipulatorCommands(); //send serial manipulator commands to servo controller
sendSignals();
testPrint();
}
```

External Interrupt (Quadrature Decoding):

```
void leftTrackEncoderCounter(){ //Interrupt handler
   LTsstatus<<=1; //shift previous A and B one step closer to correct location
   LTsstatus|=digitalRead(5); //read A
   LTsstatus<<=1; //shift previous A and B to correct location as well as new A
   LTsstatus|=digitalRead(9);  //read B
   int fbs = LTsstatus&15; //filter out highest four bits
```

```
    if (fbs==2||fbs==4||fbs==11||fbs==13){ //forward direction
      ++LTcount;
    }
    else {
      --LTcount; //reverse direction
    }
  }

  void trackSpeedInput(){//Speed Update
   if ((millis()-trackSpeedPreviousMillis) > 10){
    LTspeed = (LTcount-LTlastCount)/10; // count/ms
    LTspeed = LTspeed/TPR; // rev/ms
    LTspeed = LTspeed*(2000*3.1416); // rad/s
    LTspeed = LTspeed*trackRad; // in/s
    LTspeed = LTspeed*leftTrackSpeedSign;
    LTlastCount = LTcount;
   //Right Track Speed not shown
    trackSpeedPreviousMillis = millis();
    speedSignal = int((LTspeed+RTspeed)/2);
    headingSignal = int((LTspeed-RTspeed)/(trackExpansion*trackWidth));
   }
  }
```

Communication (Robot Receiving):

```
  void receiveCommands(){
   while (Serial3.available()>0){
    currentReChar = Serial3.read();
    if(currentReChar == mySendTokenID){
      myTurnSend = 1;
    }
    if(currentReChar == myID || currentReChar == allRobotID){
      int i = 0;
      while(currentReChar != commandEnd){
```

```
        currentReChar = Serial3.read();
          serialReData[i] = currentReChar;
          i++;
       }
      }
      if(currentReChar ==commandEnd){
        captureClearData(serialReData);
       // delay(5);
      }
     }
    }
```

Track Speed PID:

```
  void trackPIDs (){
  //Right track not shown
   LTspeedSmooth = smooth(LTspeed, trackSpeedFilterVal, LTspeedSmooth);
   int timeChangePID = int(millis())-previousTimeTrackSpeedPID;
   if (timeChangePID >= trackSpeedPIDsampleTime){
    //Compute working variables
    float LTError = leftTrackSpeedRef - LTspeedSmooth;
    leftTrackITerm +=(kil*LTError);
    //check bounds on I term
    if(leftTrackITerm>maxTrackSpeed){
     leftTrackITerm=maxTrackSpeed;
    }else if(leftTrackITerm<minTrackSpeed){
     leftTrackITerm=minTrackSpeed;
    }
    float dLTError = LTError - lastLTError;
     //PID Outputs
    leftTrackPIDOutput = kpl*LTError+leftTrackITerm+kdl*dLTError;
    //check bounds on output
    if(leftTrackPIDOutput>maxTrackSpeed){
     leftTrackPIDOutput=maxTrackSpeed;
    }else if(leftTrackPIDOutput<minTrackSpeed){
     leftTrackPIDOutput=minTrackSpeed;
```

```
    }
    //Remember errors and time
    lastLTError = LTError;
    previousTimeTrackSpeedPID = millis();
   }
  }
```

Sensor Filtering:

```
  int smooth(int data, float filterVal, float smoothedVal){
   if (filterVal > 1){      // check to make sure param's are within range
    filterVal = .99;
   }
   else if (filterVal <= 0){
    filterVal = 0;
   }
   smoothedVal = (data * (1 - filterVal)) + (smoothedVal  *  filterVal);
   return (int)smoothedVal;
  }
```

Sample Averaging:

```
  void getSensorFeedback(){
  //Running average for joint angle measurement
   // Subtract the last reading:
   theta1Total = theta1Total - theta1Readings[thetaIndex];
    // Read from the sensor:
    theta1Readings[thetaIndex] = analogRead(18);
    // Add the reading to the total:
    theta1Total = theta1Total + theta1Readings[thetaIndex];
   // Advance to the next position in the array:
   thetaIndex = thetaIndex + 1;
   // If at the end of the array...
   if (thetaIndex >= manipNumReadings) {
      thetaIndex = 0;      // ...wrap around to the beginning
   }
    theta1Average = theta1Total / manipNumReadings;
  }
```

Inverse Kinematics:

```
void inverseKin(){
 DKin =pow(XPosReq,2);
 DKin= DKin -2*lmo*XPosReq;
 DKin = DKin +lmo2-2*lm2+pow(ZPosReq,2);
 DKin = DKin*lm2Inv*0.5;
 //Check position limits
 if(DKin > 1 || DKin < -1){
  if(opMode == 1){
    XPosReq = XPos;
    ZPosReq = ZPos;
  }else if(opMode == 2){
    invalidInverseKin = true;
    XPosReq = XPosReqPrev;
    ZPosReq = XPosReqPrev;
  }
 }
 //Re-calculate for reset positions
 DKin =pow(XPosReq,2);
 DKin= DKin -2*lmo*XPosReq;
 DKin = DKin +lmo2-2*lm2+pow(ZPosReq,2);
 DKin = DKin*lm2Inv*0.5;

 if(DKin ==1){
  theta3 =0;
 }else{
  double DKinY = pow(DKin,2);
  DKinY = -sqrt(1-DKinY);
  theta3 = atan2(DKinY,DKin); //elbow down (rad)
  }
  theta2 = atan2(ZPosReq,XPosReq-lmo)-atan2(lm*sin(theta3),lm*(1+cos(theta3))); //(rad)
}
```

Appendix D

Video of Test Results

This appendix is a video of test results of the R2TM3. The file name of this video is "Phillips_Sean.wmv". If you accessed this from a source other than the University of Waterloo then you may not have access to this file. You may access it by searching for this thesis at http://uwspace.uwaterloo.ca.