Title

Kinematics Analysis On Optimal Geometry For Robust Latching And Increased Load Capacity For LIMMS: A Modular, Multi-Modal Robotic System

Author

Fernandez, Gabriel Ikaika

Publication Date

2023

UNIVERSITY OF CALIFORNIA
Los Angeles

Kinematics Analysis On Optimal Geometry For Robust Latching And Increased Load Capacity For LIMMS: A Modular, Multi-Modal Robotic System

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Aerospace Engineering

by

Gabriel Ikaika Fernandez

2023

ABSTRACT OF THE DISSERTATION

Kinematics Analysis On Optimal Geometry For Robust Latching And Increased Load Capacity For LIMMS: A Modular, Multi-Modal Robotic System

by

Gabriel Ikaika Fernandez
Doctor of Philosophy in Aerospace Engineering
University of California, Los Angeles, 2023
Professor Dennis W. Hong, Chair

Delivering packages from a warehouse to the customer's front door consists of a diverse set of sub-tasks, from packing packages into a truck to organizing them based on destination and even carrying them to a doorstep. Such a variety of behaviors can often be difficult for a single robot to achieve efficiently. Numerous research papers have proposed splitting the tasks amongst specialized robots with some dedicated to robotic arm manipulation and others to wheeling packages to their destination. Recently, companies have even started to pivot to the full generalized solution of humanoid robots. Although some success has been seen with both solutions, these technologies occupy large amounts of space, often have large costs, and have significant limitations that prevent parallelization and scalability. As a result, this reduces the amount and efficiency of overall package delivery capacity.

To resolve these bottlenecks, we introduced a new robotic platformed called LIMMS, or Latching Intelligent Modular Mobility System. LIMMS is a symmetric 6 degree of freedom (DOF) modular manipulator robot with a latching mechanism and wheels at both ends. In one configuration, LIMMS can use its end effector to latch itself to designated anchor points to behave like a traditional manipulator and move boxes around. Meanwhile, by placing anchor points on the box, multiple

LIMMS can attach themselves to a box to act as legs in which the box is the body to facilitate transportation across larger distances including outside of the truck. In this way, all package delivery sub-tasks are covered by the different modes LIMMS can be in. The only limiting factor to the functionality of LIMMS is the anchor points. These anchor points, sometimes referred to as *latching patterns* or *patterns*, can be as close as a few inches or even tangent to each other given the box is sturdy enough. Many more LIMMS can work together to lift heavier boxes than any humanoid robotic system could. Not only has its physical constraints been reduced, but its spatial constraints as well. When parallelizing a task, having more robots can increase efficiency. However, too many robots working together can cause congestion leading to an inefficient system. With the latching mechanism, LIMMS can use any surface with latching patterns, e.g., walls, ceilings, boxes. This increases the overall operational surface area reducing the chance for gridlocks. With all of these features, LIMMS can be used to solve a large variety of problems more efficiently that no other system can due to its design.

This manuscript aims to detail the development and research findings of LIMMS. In particular there are five major contributions. The first being the introduction of LIMMS as a platform and the prototypes designed and built. Secondly, the latches on LIMMS are designed in such a way that it admits a large region for which latching is theoretically guaranteed. This theoretical boundary exceeds existing self-aligning mechanisms. The third contribution is the use of Jacobian fields to determine trajectories in which the load at the end effector is balanced through the structure of the robots, such that the effective torque required is much lower. Fourthly, we formulate an optimization problem to generate trajectories for LIMMS to use to deliver packages. Lastly, we demonstrate LIMMS as a concept with four LIMMS connecting to a box to form a quadruped to deliver itself and then return to the truck.

The dissertation of Gabriel Ikaika Fernandez is approved.

Guy Van den Broeck

Robert T. M'Closkey

Jason L. Speyer

Dennis W. Hong, Committee Chair

University of California, Los Angeles
2023

This dissertation is dedicated to my father Frank Michael Fernandez
February 18, 1957 - July 28, 2021 and the rest of the Fernandez Ohana.

TABLE OF CONTENT

LIST OF SYMBOLS

Regular Symbols

Symbols	Definition
q	joint angle
q_1	base joint of LIMMS robot which is anchored
q_2	joint next to base joint and q_3
q_3	joint next to q_2 and q_4
q_4	joint next to q_3 and q_5
q_5	joint next to q_4 and q_6
q_6	refers to joint at the end effector which is not anchored
t	time in seconds
m	meters
kg	kilograms
N	Newtons
s	seconds
lbs	pounds
RPM	revolutions per minute
V	voltages
$\|$	separates a functions inputs from given parameters
$\alpha_1, \alpha_2, \alpha_3$	arc length between intersections, Fig. 3.12
β_1, β_2	angle from sector edge to line through r', Fig. 3.12
F_N	normal force Fig. 3.4
μ	coefficient of friction, Fig. 3.4
γ	angle of blade tip about (r', θ), Section 3.3
IC	instant center, Fig. 3.4
l_1, l_2	arc length of unblocked α_1 and α_2, Fig. 3.12
p	probability of latching success for a position, Section 3.3

ϕ_1	half of the angle blocking the circle, Fig. 3.12
ϕ_2	angle defined by ϕ_1 and blocking point, Fig. 3.12
Ψ	misalignment tolerance metric, Section 3.3
r	length from cutout center to nearest hole edge, Fig. 3.2
r'	length from assembly center to cutout center, Fig. 3.4
R	radial length of cutout, Section 3.3
ρ	length from blade assembly center to blade tips, Fig. 3.2
τ	torque applied, Fig. 3.3b and Chapter 4
θ	angle between r' and nearest edge, Fig. 3.4
ϑ	angle between edges defining holes in pattern, Section 3.3
V	velocity, Fig. 3.4
W	length from blade assembly center to outer limit, Fig. 3.2
x, y	Cartesian coordinates of r and θ, Section 3.3
D_c	connector diameter or $2W$, Section 3.4
D_h	hole pattern diameter, Section 3.4
T_c	connector thickness, e.g., base to blade tips on latch, Section 3.4
$\delta_{La,i}, \delta_{Lg,i}$	binary variables for LIMMS region constraints, Chapter 5
$\delta_{Ba,i}, \delta_{Bg,i}$	binary variables for box region constraints, Chapter 5
$z_{B,i}, z_{L,i}$	binary variables for LIMMS and box mode constraints, Chapter 5
$z_{S,i}, z_{W,i}$	binary variables for anchor points mode constraints, Chapter 5
$z_{Ac,i}, z_{Lc,i}$	binary variables for LIMMS to LIMMS mode constraints, Chapter 5
pr	previous, Chapter 5
po	post, Chapter 5
$\mathbf{p}_B, \mathbf{R}_B, \mathbf{c}_B$	continuous variables for box in planning problem, Chapter 5
$\mathbf{p}_L, \mathbf{R}_L, \mathbf{f}_L$	continuous variables for LIMMS in planning problem, Chapter 5
$\mathbf{a}, b$	separating plane variables, Chapter 5
$\lambda_{La}, \lambda_{Lg}$	continuous convex combination variables for LIMMS in planning, Chapter 5
$\lambda_{Ba}, \lambda_{Bg}$	continuous convex combination variables for box in planning, Chapter 5

g	on ground in planning, Chapter 5
a	in air or unattached in planning, Chapter 5
$\mathcal{L}$	Lagrangian, Chapter 5
$\mathbf{w}$	dual variable of the Lagrangian, Chapter 5
$\mathcal{I}_M$	indicator function, Chapter 5
$\mathcal{I}_N$	nonlinear constraints, Chapter 5
$\mathbf{var}_1, \mathbf{var}_2$	containers for variables, Chapter 5
r	maximum residuals, Chapter 5
J	Jacobian, Chapter 4
γ_e	end effector forces and wrenches, Chapter 4

Subscripts & Superscripts

Subscripts	Definition
in	refers to inner alignment for latching mechanism, Chapter 3
out	refers to outer alignment for latching mechanism, Chapter 3
$\|$	separates a functions inputs from given parameters

Superscripts	Definition
$*$	indicates an optimal parameter
T	matrix transpose

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my father, Frank Fernandez, who encouraged me pursue higher education. Without him, I would have probably not done a Doctorate of Philosophy in engineering. My mother, Janis, and siblings, Cana, Jade, and Frank III have been unwavering in their support throughout this journey. To my extended family, I want to express my gratitude for your patience.

I would also like to thank my committee members. Dr. Hong, thank you for your support, guidance, and positive spirits. Prof. Speyer, you have always been an inspiration for me to pursue difficult academic topics. Prof. M'Closkey, I have always looked up to the way you carried yourself and your lectures. Guy, you have opened my mind to new ways of thinking, and you are always willing to help me in anyway possible. I really am grateful to each of you.

This research would also not be possible without the support and guidance from LG Electronics. In particular Seyul Son or Sonny has helped tremendously with organizing the project milestones and provided great feedback. Me as well as the entire Robotics and Mechanism Laboratory (RoMeLa) family is very appreciative of LG Electronics.

Also this project would not have been as successful as it was if not for all the RoMeLa lab members who helped with the LIMMS project. There is a long list and I will attempt to list them all: Taoyuanmin (Tym) Zhu, Samuel Gessow, Harry Nam, Yeting (Nick) Liu, Xuan Lin, Colin Togashi, Justin Quan, Ruochen Hou, Aditya Navghare, Mingzhang (Norman) Zhu, Quanyou (Fred) Wang, Kyle Gillespie, Donghun Noh, and Ji Sung Ahn. If anyone is missing, please forgive me.

Finally, I would like to thank all my friends whom I have not mentioned yet who have kept UCLA enjoyable. There are too many to list here but a few: Nancy and Alex, Simone and Steve, Reyn and Kevin, and Stephanie. Tiphaine Galy has especially helped me through so much during my time here.

VITA

2018–2023	Doctorate of Philosophy Candidate, Mechanical and Aerospace Engineering Department, RoMeLa, University of California, Los Angeles.
2023	Led RoboCup Adult-Sized Soccer Team, Mechanical and Aerospace Engineering Department, RoMeLa, University of California, Los Angeles.
2022	Best Paper Award at IEEE International Conference on Ubiquitous Robots 2022, Mechanical and Aerospace Engineering Department, RoMeLa, University of California, Los Angeles.
2020–2021	∼ $1.25M Proposal Written and Awarded from LG Electronics, RoMeLa, University of California, Los Angeles.
2019	Best Paper Award at IEEE/RSJ International Conference on Intelligent Robots and Systems 2019, Mechanical and Aerospace Engineering Department, RoMeLa, University of California, Los Angeles.
2018	Media Appearance on Netflix Magic for Humans, Mechanical and Aerospace Engineering Department, RoMeLa, University of California, Los Angeles.
2016–2018	Masters Degree, Mechanical and Aerospace Engineering Department, RoMeLa, University of California, Los Angeles.
2016–2017	Recipient of Graduate Opportunity Fellowship Program, 1st Year Fully Paid, University of California, Los Angeles.
2014–2015	Chinese Language Program HSK 5, Shanghai, China, Fudan University.

2012–2014	Yale China Fellowship Recipient, 2 Year Fellowship to Teach and Outreach in China's Countryside, Xiuning, Huangshan, China, Yale University.
2009–2012	Bachelor of Science, Mechanical Engineering, Yale University.
2009–2012	3-Year Offensive Lineman Starter, Yale Varsity Football Team, Yale University.
2012	National Football Foundation Hampshire Honor Society Member, Yale University.
2011	Ted Turner Award Recipient, Given to the Best Yale Offensive Lineman, Yale University.
2011	All-Ivy Selection, Ivy League's All Star Player Picks, Yale University.
2011	Light Fellowship Recipient, Study Abroad in Japan Fellowship, Sun Academy, Tokyo, Yale University.
2008–2009	Offensive Lineman on UCLA Varsity Football Team, Red Shirt or Acclimation Year, University of California, Los Angeles.

CHAPTER 1

Introduction

Last mile delivery presents a very challenging problem due to the uncontrolled, dynamic environments and the varied tasks that the robot must face to complete a successful delivery. To start, delivery vehicles are filled to capacity and packages often are shuffled and flipped around during transit. Then at the destination, packages need to be correctly identified and unloaded. Once outside of the vehicle, traversing the landscape to the door proves difficult as cracked, concrete paths and curbs in front of skyscrapers or single story houses with lawns and dogs must all be accounted for. The package may need to be delivered up multiple stairways while avoiding people and animals along the way. Even after all that, the package deliverer now needs to make it back to the delivery truck in a timely manner and prepare for the next delivery.

For robotics attempting last mile delivery, the criteria required for unloading the packages from the delivery truck and moving them to their desired location are vastly distinct. Often times robots are usually highly specialized for a particular task. As such a number of research directions explored the collaboration between different types of robots, such as, manipulators mounted on top of quadrupeds [6,7] or manipulators robots working with wheeled robots [8, 9]. These technologies, though theoretically capable of last mile delivery, are not specifically designed for package delivery. Two separate existing systems are simply combined.

Practical solutions that have been adopted recently only solve a sub-task rather than the full one, meaning that the robots can be specialized. For example, Starship robot [1] as seen in Fig. 1.2a is designed to wheel contents to the end user. They require the vendors and customers to do all the manipulation. Couriers and

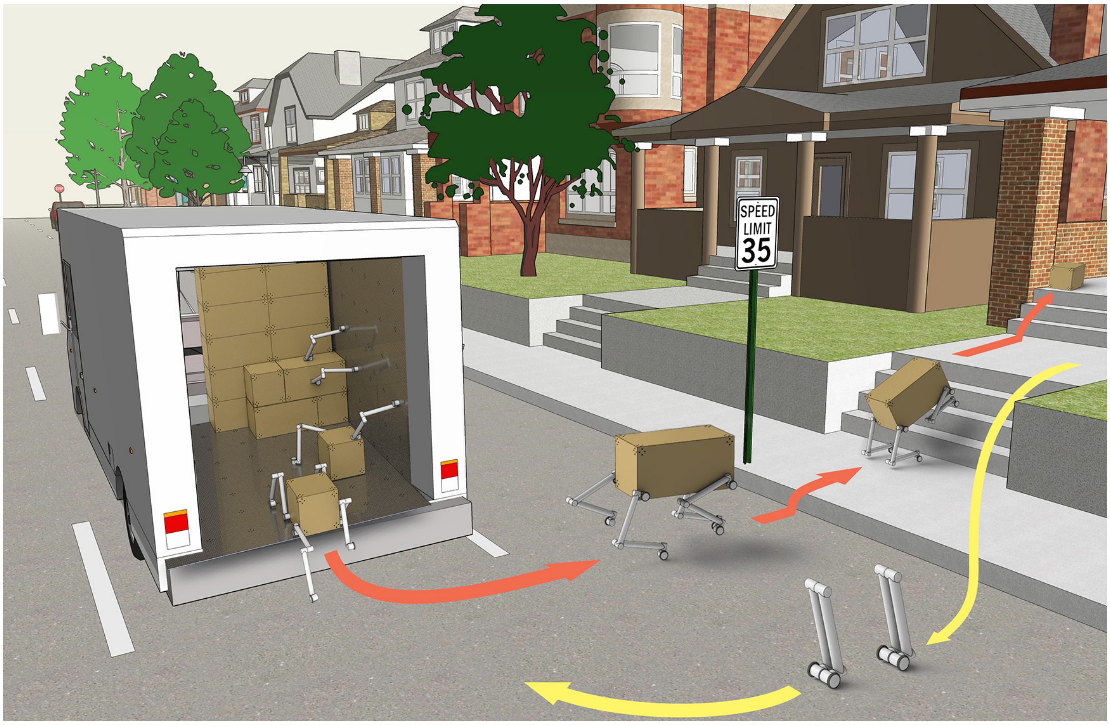


Figure 1.1: LIMMS, which stands for Latching Intelligent Modular Mobility System, concept depiction of it doing last mile delivery. First when inside the truck, LIMMS can sort packages ahead of time a queue them as needed. It can behave like a manipulator by fixing its base on the truck walls with a latching mechanism. Four LIMMS can then attach to the box to move the package like a quadruped robot by using the box as its body. Once the package is delivered, LIMMS can return to the truck in a wheeled mode.

businesses have to place the package into Starship robot's cargo bay, and recipients have to remove the package when the robot arrives. Starship robots only handle transportation. Numerous robots, such as Kiva robots [10] which move shelves in large warehouses for workers to sort, have been widely adopted in industry but confined to tasks preceding last mile delivery.

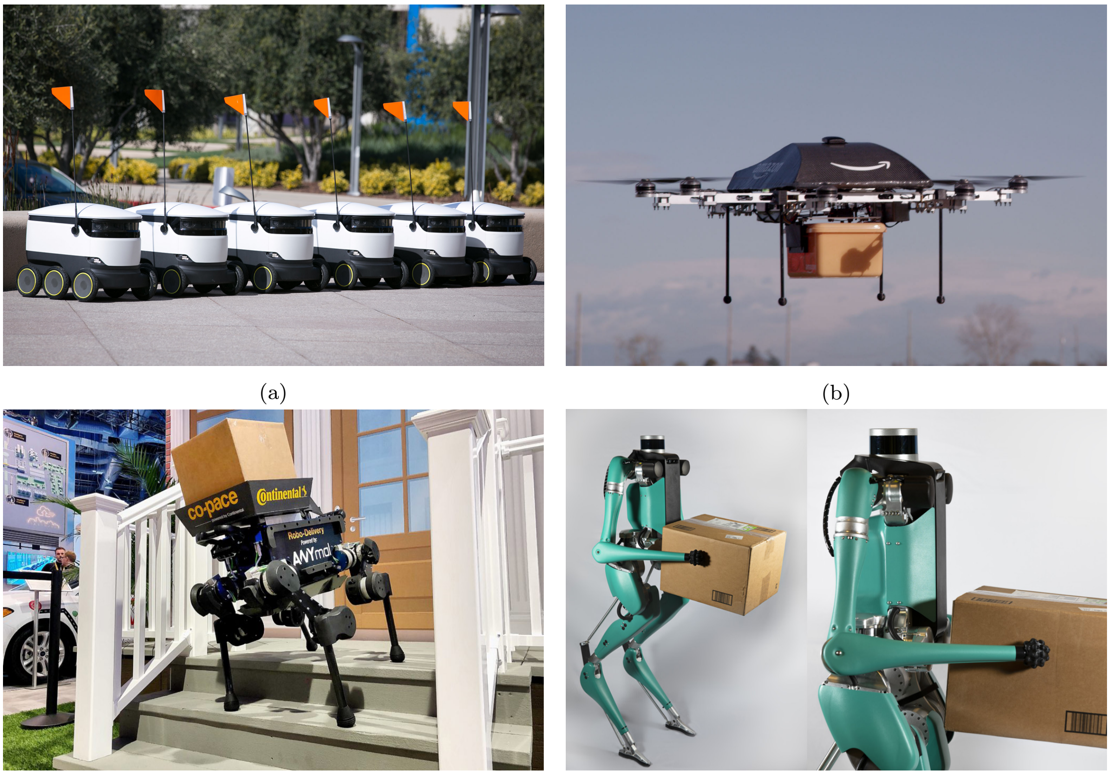
Wheeled robots by design are constrained to moving atop planes, though be it very efficiently. When the environment is unknown, it is difficult for these systems to plan in real-time a proper route that maintains the terrain is flat enough while avoiding obstacles or dead-ends. Even when operating in the presence of multiple wheeled robots, the two dimensional planar constraints often causes bottlenecks in the absence of complex coordination. In warehouses it is not uncommon to have robots waiting in place while others are moving to avoid colliding with them.

This can have a cascading effect where more and more robots end up stalled on the floor. Effectively, wheeled robots change the boundaries of other robots' operational space at any given time. Each move a robot establishes a new physical constraint. Scaling up to hundreds or thousands of these robots becomes very difficult in terms of efficient planning. Trucks, on the other hand, have a much smaller floor surface area than a warehouse. This exacerbates the gridlock issue. Having too many wheeled robots confined in a space can make it more inefficient than a single robot, e.g. a huge traffic jam.

From this perspective drones are much better since they can utilize all three dimensions. Amazon Prime Air [2], see Fig. 1.2b, has been testing this service for years without large adoption. This in part could be due to several factors including the types of spaces they can safely and reliably operate in. Noise and airspace regulations as well as safety concerns limit geographically where and when drone deliveries can be made. Additionally, any apartment complex with a mail room would not be able to easily receive any packages. The weather is another factor that can hamper drone usage as well.

Robots that are able to handle more complex tasks are often more complex in design and larger in scale. ANYmal for instance, as shown in Fig. 1.2c, is a quadruped robot [3, 7] about the size of a large dog and is capable of traversing a far greater variety of terrains and obstacles than a wheeled robot could. With the additional degrees of freedom (DOF) quadrupeds can even manage to twist their bodies to unload packages from their backs onto the ground. Quadrupeds like ANYmal have been outfitted with wheels to provide efficient locomotion over flat surfaces [11]. Other quadrupeds have been shown to be able to convert two of its legs to arms for simple manipulation [12].

In this sense humanoid robots can be viewed as a more complex quadruped robot with more general capabilities. Delivery robots would ideally be humanoid robots since they theoretically can do what humans do in environments designed



(a) (b) (c) (d)

Figure 1.2: Existing delivery robot platforms: (a) Starship robot [1], (b) Amazon Prime Air drone [2], (c) ANYmal quadruped robot [3], (d) Digit humanoid robot [2].

for humans. Humanoid robots such as Digit, as seen in Fig. 1.2d, have been proposed as all purpose platforms capable of last mile delivery [2].

General, legged-manipulation systems like humanoids and to an extent quadrupeds tend to have a higher cost of transport [13], which increases the cost per package. Legged systems, additionally, require a much larger spatial footprint reducing the total space devoted to packages. This compounding effect can drastically decrease the efficiency of an automated last mile delivery system. Therefore, legged systems, though capable of fulfilling the entire last-mile delivery task, are not cost efficient for scale and parallelizability for wide adoption.

To overcome these limitations other platforms possess, we propose LIMMS, which stands for Latching Intelligent Modular Mobility System. This robot as the name implies is a 6-DOF, modular arm-like robotic system with latching mechanisms at both ends allowing it to fix one end to a latching pattern. In this way, either side of the robot could be the robots base. Conceptually, each latch is driven by a single motor with two output shafts where the other output shaft drives a wheel. When multiple LIMMS fix their bases to a box, it can behave like a quadruped robot where the box is its body and wheels provide efficient transportation over flat terrain. By fixing its base to different surfaces, LIMMS can also behave like an arm for manipulation or modes that combine manipulation and locomotion. When a large number of LIMMS work together, they can cooperate and take on various responsibilities to optimize the last-mile delivery problem in more efficient ways not considered before.

Similar modular arm-like robotic concepts have been first proposed for spacecraft applications, although none of them have incorporated the use of them to transform themselves into a system like a quadruped. Using a system without a fixed base in space reduces the burden on each end effector actuator when serving the role as the base of the robot, which has to additionally carry any load as well as the full robot itself. The Mobile Servicing System with its Space Station Remote Manipulator System and the European Robotic ARM from the International Space Station (ISS) are a few examples of space manipulators that can both use attachment points and move end-over-end to different parts of the ISS for servicing and maintenance [14–17]. Truss manipulators have taken inspiration from these space manipulators in being able to construct large truss structures that have many latching points along each truss to traverse upon [18, 19].

More generally these types of robots can be classified as modular self-configurable robots (MSR), which tends to focus more on connecting with each other to reconfigure themselves into more sophisticated systems [20]. LIMMS technically could do this albeit with limited use cases since the scale of a singular robot is much larger, on the order of a meter, than the typical MSR. One of the first leading papers that started the MSR field compared MSR to cells reorganizing themselves with a single robotic cell being on the scale of centimeters

[21]. The field has now branched out to other MSR systems such as LIMMS where now its environment and objects it manipulates can be incorporated into the system's structure itself for even more complex behaviors [2].

A conceptual rendering of how LIMMS operates is shown in Fig. 5.1. The delivery vehicle makes its way down the street to the address of the package recipients location. The walls, floors, ceiling of the truck and the surfaces of the boxes are lined with simple anchor points, also referred to as latching patterns, to which LIMMS can attach one end to. During the driving period, LIMMS attaches to these surfaces to find and move the next box towards the front of the truck. Once close to the desired location, four LIMMS can attach themselves to the delivery box to form a quadruped with the box serving as the body of the new system. The quadruped would then climb down from the truck and make its way over the curb and up stairs to the front door of the customer. The wheels at the end effectors closest to the ground can also be for more efficient locomotion when in this mode. Once it reaches its target location, the quadruped can rest its *body* near the front door, at which point each LIMMS unit can detach from the box. LIMMS can either return as a single unit or attach with other LIMMS, depending on the environment, to return to the delivery truck for the next package. Given the size of LIMMS many units can fit into the delivery truck by fitting on the roof or other less used places. This allows for LIMMS to scale to larger irregular packages and parallelize to a large quantity easily.

This section was followed arguments made from [2]. The rest of this manuscript is organized in the following manner: Chapter 2 talks about the design consideration from concept to hardware as described in [2]. It also goes over each different prototype that is used for experiments in latter chapters. Chapter 3 describes the self-aligning latching mechanism that maximizes theoretical guarantees derived in [22, 23] for successful latching. At the very end hardware implementations and a locking mechanism is presented. Chapter 4 introduces the latest work in LIMMS that takes advantage of the systems unique kinematics to reduce the overall load experienced by the joints. Chapter 5 discusses an offline mid-level planner to optimize for a particular task, such as coordinating between two to four LIMMS to move a box out of a truck [24]. The author collaborated with a colleague by forming logical constraints necessary for the optimization formulation and the environment in which the results were demonstrated. Finally, Chapter 6 covers final thoughts on the works presented in all previous sections, on the future and past experiences of LIMMS, and more broadly on the contributions to the field of robotics.

CHAPTER 2

Design

This chapter is focused on the overall design iterations of LIMMS body excluding any in-depth discussion on the latches. The latching mechanism has its own chapter. Please see Chapter 3 for a more detailed description of the latching mechanism. Since this is a complex project with a short development horizon, the focus has been on producing prototypes that were capable of demonstrating the feasibility of LIMMS. Henceforth, much emphasis has been placed on the mechanical design. Similarly, this chapter focus on the mechanical design, since most of the electronics, sensors, and computational units are mostly off the shelf and not optimized in any significant way. There are three distinct prototypes of LIMMS that we built and tested: prototype iteration 1, prototype iteration 2, and BEAR (Backdrivable electromechanical actuators for robotics) LIMMS prototype. These suffice for a proof of concept and will be referenced throughout. However, there are additional improvements required for achieving the original stated goal for last mile delivery. This section's contributions can be summarized by the following:

1. Detailed LIMMS kinematic considerations and assumptions,
2. Established minimum torque requirements for feasibility of each mode in simulation, and
3. Presented an early stage hardware prototypes of LIMMS.

This section was expanded upon from [2].

2.1 Considerations & Assumptions

For practical purposes the design requirements depend on the average case of package deliveries. This limits the weights and sizes that will be considered. The travel distance and obstacles encountered are also restricted to simple door-to-door delivery in a suburban neighborhood.

Based on public statistics and statements made, majority of Amazon's deliveries, 86%, weigh under 5 lbs, which is a target weight goal for their drone delivery service [25]. For LIMMS the target weight capacity per package is 2 kg. This would cover a significant portion of packages for an e-commerce company. Theoretically, in the case of much heavier packages the number of LIMMS working together can scale by adding more latching patterns.

The packages will take the form factor of cardboard boxes. To make stacking boxes considered consisted of those coming in multiples of 12 in (≈30 cm) for any dimension. The smallest box being 12 in × 12 in × 12 in. The largest box we consider is elongated in one dimension to be 24 in × 12 in × 12 in. This requirement makes for easier stacking and organization within a delivery vehicle.

For round trips from the vehicle to the porch, it is presumed to be 50 m, and the most common obstacles for locomotion are curbs and steps. In the United States curb heights are standardized to be 6 in (≈15 cm) [26]. Stairs leading to the front door, though, can vary much more. The California Building Code states that the maximum stair height is 7 in (≈18 cm) with a minimum depth of 11 in (≈28 cm) [27]. To determine the duty cycle for LIMMS, the following schedule is assumed:

1. Idle in delivery vehicle and charging battery (15 min).
2. Locate and attach to the delivery box (5 min).
3. Enter quadruped mode and get perception data (3 min).

4. Use wheels or trot towards destination (5 min).

5. Detach and return back to the delivery vehicle (5 min).

This section was expanded upon from [22, 23].

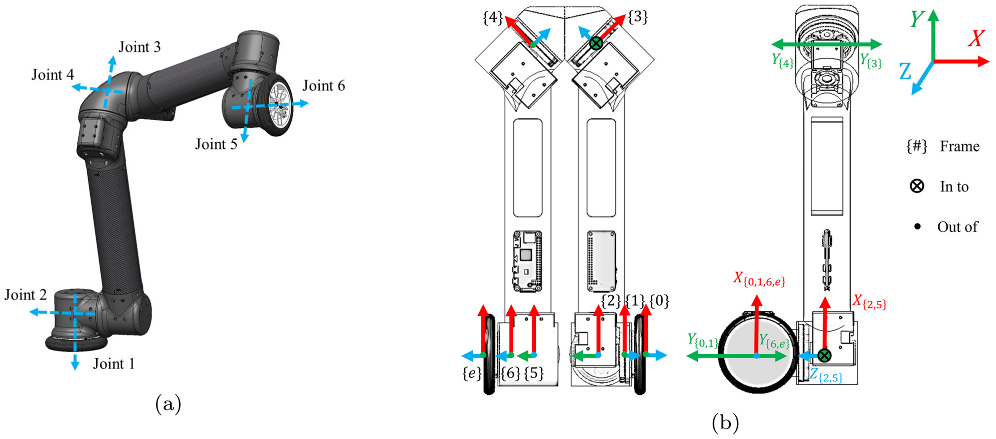
2.2 Kinematics

Given the complexity of trying to manipulate boxes that have been shuffled around in a truck without detrimental collisions and coordination with other LIMMS, it is advantageous to have as many DOFs as possible. Given that either side could be the base any additional actuators could hinder the robots performance. Therefore, 6 DOFs was decided upon, so that at minimum it would have sufficient DOFs to reach positions and orientations of latching patterns within its workspace. With the bases being able to switch, those DOFs were placed in a nontraditional symmetric fashion. The symmetry would allow for either end to serve in any position irrespective of the particular role the LIMMS unit was serving at the time. This would simplify coordination and eliminate constraints determined by tasks that require a particular end as a base.

2.2.1 Joint Configuration

For comparison a traditional 6-DOF robotic manipulator arm usually has a 3-DOF wrist for orientation and an elbow joint with 2 more DOFs at the base for yaw-pitch. This configuration features a couple of advantages. One is that solving its kinematics and inverse kinematics equations can be done in closed form. The second is that since only one end serves as its base the torque requirements for the base motors can be much larger to compensate for the weight of the robot while maintaining smaller lighter actuators at the wrist.

Flipping a traditional robotic manipulator, similar to LIMMS, where its wrist



(a)

(b)

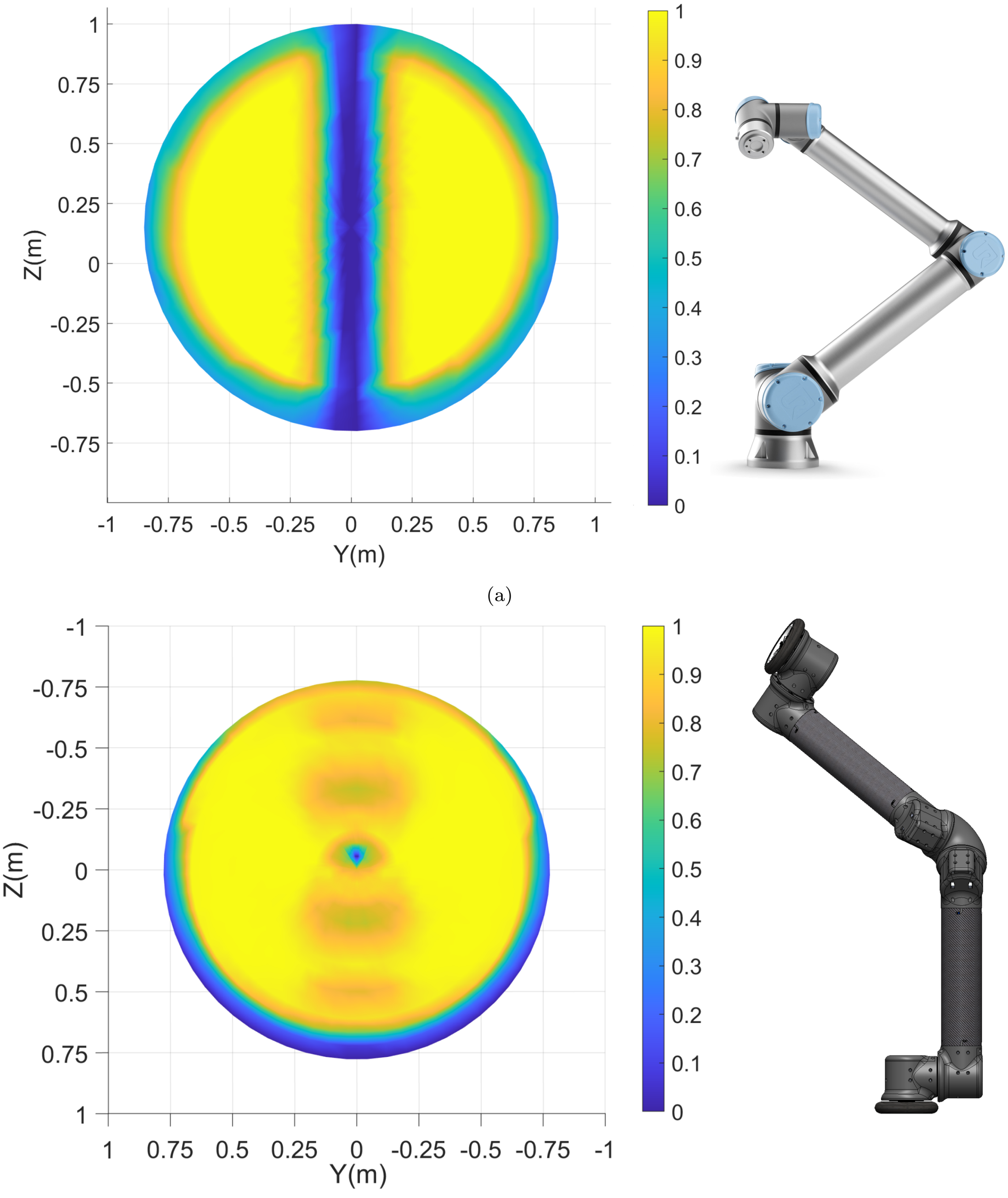
Figure 2.1: First prototype iteration of LIMMS joint configurations without latches shown for clarity. (a) LIMMS 6-DOF joint configuration. (b) *Left* depicts a front view of LIMMS with joint frames. *Right* shows a side view. Note that these are nontraditional DH frames.

would serve as the new base of the robot would result in numerous disadvantages. The arm would now consist of a *new* 3-DOF base and a *new* 2-DOF wrist. The base motors of traditional robot arms are optimized to carry much of the inertia, resulting in the *new* 2-DOF wrist being much heavier than its original configuration. Moreover, its *new* 3-DOF base would have a much lower maximum torque output that probably will not even be capable of lifting the arm itself with its small wrist actuators. Effectively, this leads to any traditional robotic manipulator configuration of the joints to favor one end as its base and suffer a loss in most of its payload capacity in its other, assuming it can even function with its *new* 3-DOF base. Even if all actuators consisted of similar power and weight, the system would still favor the wrist for fine orientation control. LIMMS, on the other hand, keeps a symmetric configuration with no single side being more advantageous than the other.

LIMMS is designed with at 6-DOF symmetric joint configuration as shown in Fig. 2.1a. The configuration has bilateral symmetry consisting of 2 joints near each end and 2 joints near the elbow. Viewing this with the perspective of a traditional 6-DOF manipulator configuration, the elbow of LIMMS has to now swing out to reach certain configurations which can make planning difficult. This is intuitive but quite simplistic though. Joints 3 and 4's orientation has an offset of $\frac{\pi}{4}$ from their vertical tubes. This further complicates the coupling of the positions and orientations by adding a 3 dimensional (3D) twist. This phenomenon complicates the analytic inverse kinematics (IK) making it nontrivial to solve in closed form unlike a traditional arm. Even symbolic IK solvers could not find a proper solution. Hence, damped least squares method [28] was used for numerical IK.

The joint axis frames shown in Fig. 2.1b are not set up using conventional Denavit–Hartenberg (DH) parameters. It is possible to use DH parameters by using multiple intermediate stationary frames with rotations. The motivation for non-standard DH parameters was to utilize symmetry as much as possible such that when switching bases the systems forward kinematics (FK) resulted in the least amount of change. Naturally aligned axes with motor output axes were chosen in such a way that each frames' location and orientation did not change with the base. A byproduct of this choice of frames with its bilateral symmetric design is that each FK transformation is very similar in form. Some frames are even the same. The only difference is that switching bases causes a few sign flips in a few offset angles. For algorithm developers this establishes a cleaner more consistent and intuitive approach. Switching bases is algorithmically equivalent to flipping a boolean, for example transforming from joint 6 to joint 5 is almost the same mathematical transformation as from joint 1 to joint 2 except the rotation around the X-axis is reversed. Thus, when solving FK only a single algorithm is needed when different bases are attached.

Using its FK, LIMMS workspace dexterity can be compared with traditional 6-DOF robotic manipulator arms, in particular the Universal Robot UR5e [5] depicted in Fig. 2.2a to the right, as a baseline comparison to evaluate the joint configuration. From the plot for the Universal Robot UR5e, it is clear that this manipulator suffers from low dexterity at the outer edges, and along the Y-axis there is a large region with approximately zero dexterity due to an offset between



(a)

(b)

Figure 2.2: Dexterity index over the workspace [4] with a higher value in *bight yellow*, denoting the end effector can reach more orientations at a specified location, and a value of zero in *dark blue*, implying the space cannot be reached, for: (a) Universal Robot UR5e [5], a traditional 6-DOF robot manipulator, and (b) LIMMS, a nontraditional 6-DOF robot.

two joints at the base. Despite LIMMS having a similar base configuration as the Universal Robot UR5e, the $\frac{\pi}{4}$ offset double joint elbow increases the workspace dexterity near the Y-axis, as shown in Fig. 2.2b. The slight difference in radius of the workspace dexterity shown is due to the actual physical size of each hardware in real life. They can always be scaled to be the same size for an abstract comparison.

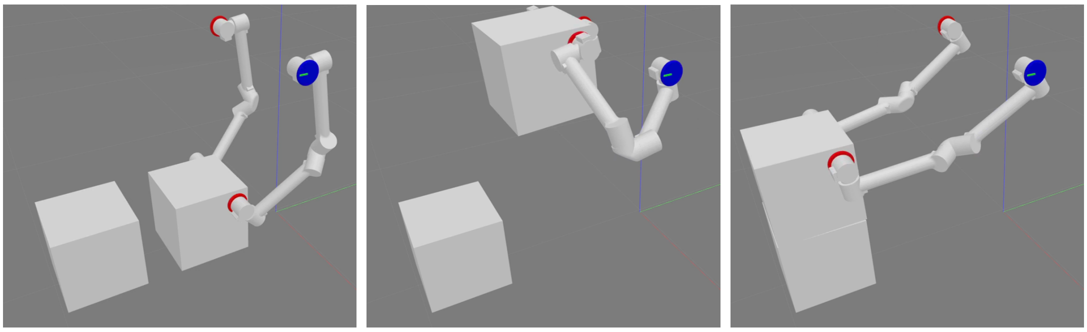
2.3 Torque Requirements

Even with the state assumptions and considerations provided in the previous section, determining the torque requirements for each actuator is a nontrivial task, especially without any preexisting prototype. A delicate balance is required in prototyping. Theoretical or numerical calculations inform hardware and vice versa. Both need to be taken into consideration without holding up the other. Therefore, before manufacturing the first prototype we setup a simulated environment in Gazebo simulator [29] and only considered three modes out of the potentially numerous modes as a minimal requirement for completing the full task:

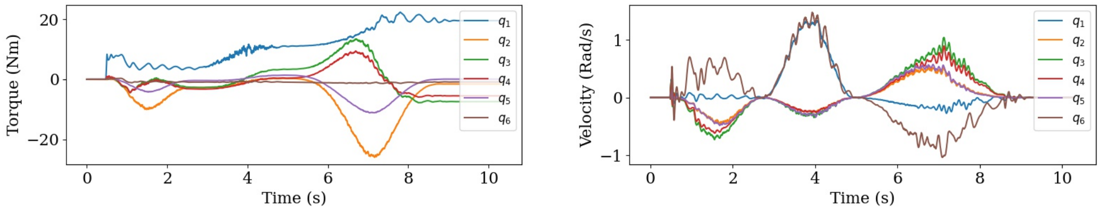
1. Two LIMMS as dual manipulators,
2. Four LIMMS attached to a box as a quadruped, and
3. Single LIMMS in self-balancing wheeled mode.

Simple controllers were implemented for each of these tasks to gain a rough estimate of what the torque and velocity requirements were at each actuator.

As part of the manipulation task for in last mile delivery, boxes need to be moved towards the back of the truck or reoriented in such a way that LIMMS can enter into quadruped mode. To approximate what torques the motors needed during this phase of last mile, two LIMMS were simulated to simultaneously lift the same box and stack it onto another box as seen in Fig. 2.3. The box weighed the target 2 kg weight. A simple position PID (proportional-integral-derivative) controller was used independently for each LIMMS, i.e. decentralized control or no feedback on coordination. The operational space trajectory for the stacking



(a) $t = 0s$ (b) $t = 5s$ (c) $t = 10s$

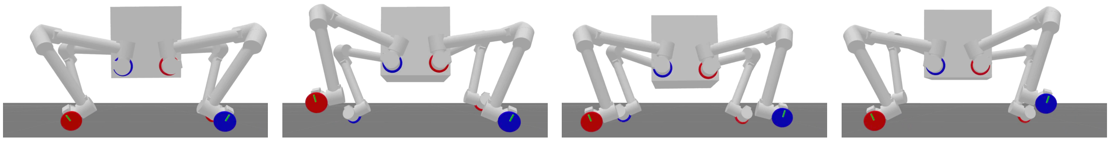


(d) Torque of the joints on a single LIMMS. (e) Velocity of the joints on a single LIMMS.

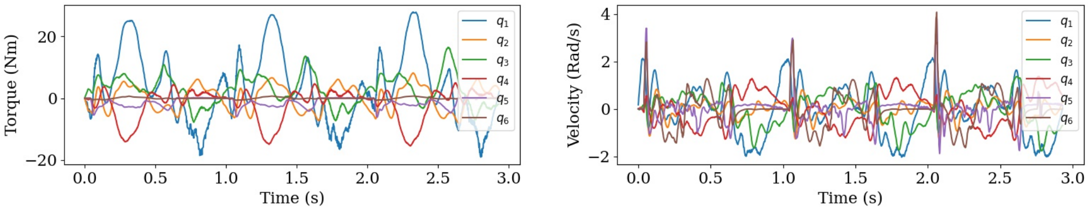
Figure 2.3: Two LIMMS where the bases are assumed to be anchored in the vertical plane lifting a box. t frames are the trajectory via points used to create this trajectory. The qs in the plot indicate the joint number. The lowest number joint refers to the base of the robot whereas the higher the number the closer it is to the end effector.

motion used three via points. Interpolation was used between each points to get the joint position commands.

By observing Fig. 2.3d and Fig. 2.3e an estimate of the torque and velocity, respectively, can be made for the manipulation portion of last mile delivery. Since each LIMMS unit performs similarly mirrored motions to lift the 2 kg box, the torque output is similar. From the graphs, the results are in line with what was expected. The motors at the base holding the weight of the entire robot along with part of the load from the box experience consistently large loads as seen in Fig. 2.4e q_1. As the box moves further away, the torque arm becomes larger. In a few instances joint 2, q_2, surpasses the base joint in torque on two occasions. This is of interest because joint 2 should not have any large loads acting on it since that would imply forces in the direction perpendicular to the trajectory of the box as well as gravity. This implies that one LIMMS was pulling away from the other. Since a simple decentralized controller was used, the LIMMS possessed no reference for coordination, which is probably the reason for the spike



(a) $t = 0s$ (b) $t = 0.25s$ (c) $t = 0.5s$ (d) $t = 0.75s$



(e) Torque of joints on a single LIMMS. (f) Velocity of joints on a single LIMMS.

Figure 2.4: Four LIMMS attached to a 2kg box performing a trot gait. Frames of the trot gait were taken starting from (a) through (e). The joints, q, are taken from the base, q_1, to the end effector joint, q_6.

in torque. Though outside the scope, better coordination in the control algorithm could slightly lower the overall torque requirement. The measured output torque required from this test was around 25 Nm. The velocity for these trajectories visually seemed acceptable.

Fig. 2.4 shows LIMMS in quadruped mode when attached to the four corners of the box. The top of the figure shows the frames for simulating walking using a trot gait. Raibert heuristics were used for footstep planning and Bezier interpolation for swing leg trajectories [12]. The stance period and swing period for the trot gait ere both 0.5 s. Decentralized position PID controllers then tracked the joint positions from the gait's planner similar to that for the manipulation task. In Fig. 2.4a the quadruped transitions from stance to lifting red wheeled foot up as seen in Fig. 2.4b. It then transitions back to stance in Fig. 2.4c and then to both blue feet up Fig. 2.4d. By repeating this cyclically, LIMMS can move with the 2 kg box at 0.3 m/s despite no consideration for leg dynamics and an open loop controller.

Fig. 2.4e and Fig. 2.4f show the torque and velocity when running three gait cycles of trot gait in simulation, respectively. The peak torque reached was 28.6 Nm, whereas the maximum velocity reached 4.1 m/s. Though, that velocity was only sustained for 0.005 s. Since the ends of LIMMS are as heavy as its base, the inertia and speed at which the foot slams into the ground is large resulting in a large spike in velocity due to the momentum change. Given how the controller used usually assumes no mass [30], a more complex controller to account for the mass would relieve this issue. Though, for a rough estimation of torque requirements this suffices.

Similar to manipulation, the base motor in quadruped mode requires the most torque in swing phase since it has to lift the entire LIMMS leg. Most quadrupeds do not suffer from this issue since most of their mass is concentrated near the base of the leg and body to minimize the nonlinear effects [30]. LIMMS symmetric joint configuration spreads the mass throughout the entire robot. Without the leg dynamics modelled and taken into account of the controller, changing phases in particular lifting off and touching down causes an unplanned stance phase otherwise not seen in regular trot gaits. Further exacerbating the problem, LIMMS weighs roughly double that of its 2 kg box body. As a result, the algorithm used for trot became very sensitive to parameter variations in gait timing, step height, and desired velocity.

Once delivering the package in quadruped mode, it is assumed that each individual LIMMS unit can return to the delivery truck in a self-balancing mode. By commanding all motors besides the ones at both ends of LIMMS to hold a folded position, LIMMS can balance itself on two wheels to move around, e.g., a Segway [31]. A PD controller can be used to achieve a desired velocity while balancing [32]. This provides a simple strategy all LIMMS units can use to move. Needless to say the requirements for torque and velocity were below those needed for manipulation and quadruped.

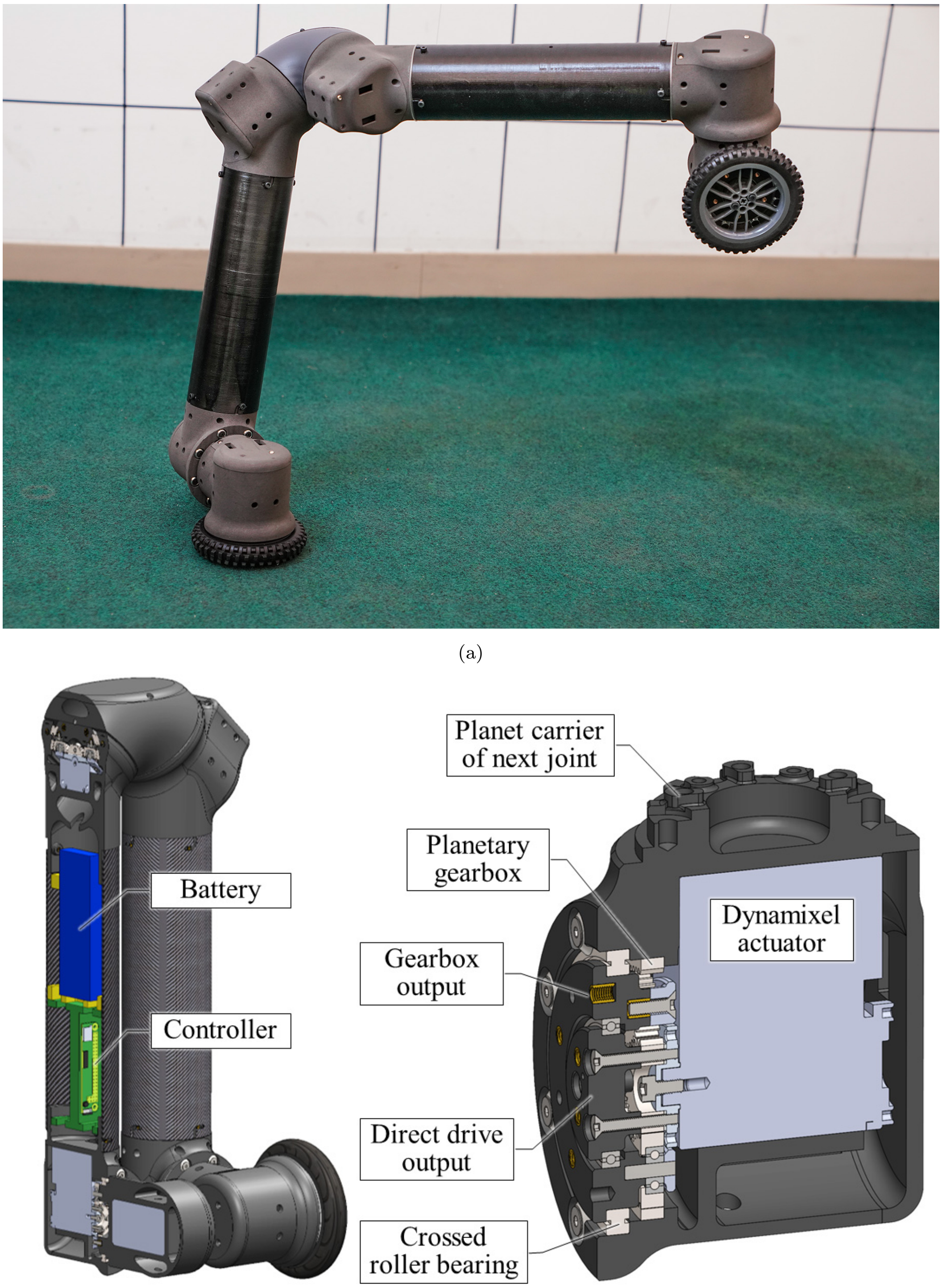
From these simulation experiments, it is obvious that with sub-optimal control strategies the required torque and speed increases. Although this should be resolved to improve features like lifting capacity, these shortcomings can be taken as a safety factor when determining motor torque-velocity characteristics. Further optimizations can be made in the future. With the torque and velocity curves for each of these tasks, a motor can be chosen to achieve the minimal tasks needed for last mile delivery for LIMMS. With the joint configurations and torque requirements a prototype can be built.

2.4 Hardware Prototypes

2.4.1 Prototype Iteration # 1

Based off of the kinematics analysis in Section 2.2 the first iteration of LIMMS prototype was manufactured. The design principle was to develop a prototype capable of testing all necessary functions separately as quickly as possible. This allows for further analysis to be done for much more complex characteristics such as torque requirements and link lengths, after which another design iteration can be done with more insights into the system before the final design. As such, this prototype's end effector could only either attach the wheel or the latch but not both at once. When the wheel is attached, it is directly connected to the output shaft increasing it maximum velocity. On the other hand, the latching mechanism is mounted to the output carrier of planetary gearbox with a gear reduction increasing its torque. Despite having area for a battery and computer, for testing those were only used for weight analysis. Most components were either off-the-shelf or easily manufactured 3D printed parts. In total it weighed 4.14 kg and could stretch up to 0.75 m. When folded, as depicted in Fig. 2.1b on the left, the prototype can be contained within a 0.43 m × 0.22 m × 0.18 m box.

Fig. 2.5a shows the hardware of the first iteration. The battery and controller are housed in carbon fiber tubes. Structural components used at the joints were manufactured with a rapid prototype method called Selective Laser Sintering (Fuse 1 SLS printer, Formlabs) using nylon 12 material. Design in this way allows



(a)

(b)

Figure 2.5: The first iteration for a prototype of LIMMS. This unit has been designed for testing and evaluations to inform future design choices for the next iteration. Therefore, some features are missing, e.g. the latch. (a) LIMMS hardware prototype. (b) Section view of LIMMS prototype using commercial off-the-shelf actuator with custom gearbox.

for a quick turnaround new designs for further testing and experimentation.

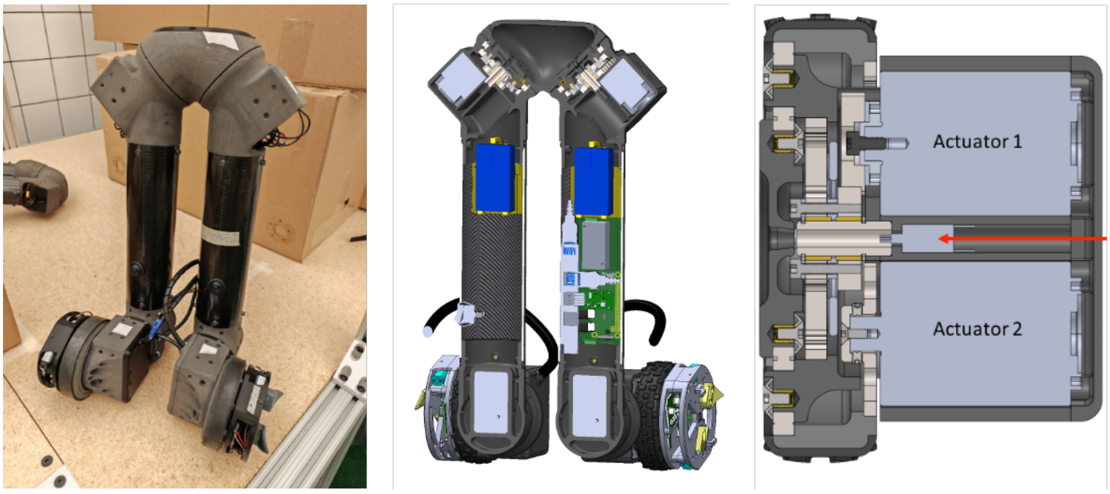
The commercial off-the-shelf actuators used at the joints were DYNAMIXEL XM540-W150-R motor from ROBOTIS. An extra external speed reduction was required for these motors since they slightly differed from the target torque computed in simulation. A 3.5:1 reduction ratio planetary gearbox was used to make up for the difference as shown in Fig. 2.5b. The modified joint now achieved 31 Nm of peak torque with a 2 rad/s max velocity. To increase the structural rigidity cross roller bearings (CRBT505A, IKO) were used.

Each end of LIMMS contained two outputs on the same surface, meaning that the wheels and latches could only be tested independently for different modes. Whenever the latch was used, the torque requirements were much larger than velocity, so the output with the 3.5:1 reduction was used. On the other hand, whenever the wheel was attached much larger velocities were needed, and the wheel was connected directly to the motor.

2.4.2 Prototype Iteration # 2

The second iteration of the LIMMS prototype weighed a bit heavier at 4.7 kg. When extended it could reach a length of 1.06 m, and when folded it reached 0.53 m. The wheel speed was improved from the 70 RPM in the first iteration to 475 RPM in this design. The batteries were split across both sides as seen in Fig. 2.6b. This is in line with our design of the configuration such that no particular end was favored for a specific task. A single on board computer was also integrated. This prototype allowed for wireless communication, whereas the first prototype had to be wired for power and computation.

When running tests on the first iteration of the prototype, it was obvious that the wheels were too slow. Ideally, a custom end effector would be built with a through shaft at which point the gear reduction ratios could be set for latching



(a) (b) (c)

Figure 2.6: Second iteration of LIMMS prototype. (a) Depicts the hardware for the fully assembled prototype. (b) An inside view of the wiring management and electronics inside the tubes. The batteries shown in blue are distributed across each half to keep dynamics approximately the same for either base. (c) The dual actuated end effector for latch and wheel. The red arrow points to a slip ring used for power and communication transmission.

and wheels without interference. This, though, is considered outside the scope of the current phase of the project. Instead overall torque capacity was sacrificed to increase the wheels' velocity. This was achieved by placing an additional actuator at each end to actuate the wheel itself as seen in Fig. 2.6c. To reach 475 RPM, a gear inside the off-the-shelf actuators was replaced by a custom manufactured one. Despite introducing another actuator, the joint at the end effector and the base joint will still be referred to as joint 6, q_6, and joint 1, q_1, respectively. The wheel and latch could be decoupled with this design with the cost of a heavier ends for LIMMS.

Having two actuators at the end effector makes the wiring for power and communication a bit more challenging since the wheel will be rotating over a full rotation. The latching mechanism also has two of its own actuators for a locking mechanism as well as a camera for visual feedback. A slip ring was incorporated into the design, as seen by the red arrow in Fig. 2.6c, to transmit power and communication without limiting the number of rotations of the wheel or latch.

From Fig. 2.6b and Fig. 2.6a, there are wires that are still exposed resulting in a constraint in the number of rotations certain joints can achieve. Preferentially, every joint would be able to rotate without limit. Practically, though, it is only necessary that the wheels be able to do that. A compromise was reached to balance fast design iterations with that of design goals for high-level optimization. The wheels and latches at the end effectors could rotate infinitely. Joints 2 and 5 have wires extruding and can only rotate slightly over a full rotation. Joints 3 and 4 in the elbow have actuators offset such that wires can pass through without a slip ring. Extra wire is provided such that roughly four full rotations can be made collectively between the two joints.

2.4.3 BEAR LIMMS

The BEAR LIMMS prototype is similar to that of the first iteration of the LIMMS prototype except everything is altered for high velocity dynamic motions. The prototype is much lighter, approximately 2.5 kg. The BEAR motors used for the prototype have a gear ratio of 9, and its speed constant is 27.3 RPM/V. Theoretically, this prototype can move very quickly but at the cost of torque. Its peak torque for 15 s at each joint is 3.5 Nm. In some joint configurations the motors overheat and fail after extended periods. This prototype, however, is designed to move fast and study what can be done at high speeds for future extensions of LIMMS, thus, increasing the number of packages that could be moved.

BEAR Motors An additional benefit of BEAR motors is that its low-reduction-ratio gearboxes allow for backdrivability as well as high transmission transparency. Its backdrivability and low reflected inertia can help deal with high impacts that can occur when robots move highly dynamically. The low-reduction ratio also reduces the friction losses and other non-linearities to the output. This gives rise to the ability to estimate the torque output directly without an external force

(a) (b)

Figure 2.7: (a) Shows 3 different in-house developed backdrivable electromechanical actuators for robotics, also known as BEAR actuators. (b) CAD of the BEAR LIMMS prototype.

torque sensor, high transmission transparency [33, 34]. With these capabilities indirect force control can be implemented on BEAR LIMMS to help with its dynamic motions.

2.4.4 Conclusion

With these prototypes we were able to implement and test sub-components as well as the full system. Each prototype played a role in the development process of LIMMS. The first iteration was used to verify weight tests and different modes in LIMMS. The second iteration tried to improve upon that by adding another motor to wheel faster and removing the tether to make things wireless. This allowed for testing of the full system untethered. The BEAR LIMMS were used to run fast latching tests and demonstrate the potential for use of Jacobian to carry much heavier loads than intended for.

CHAPTER 3

Latching Mechanism

During a single last mile delivery, LIMMS will transform between different modes frequently, e.g. lifting a package requires manipulation mode and moving a package quadruped. Even simply switching between these modes would give rise to numerous latching and unlatching actions, especially if LIMMS needs to move packages from the back of the delivery vehicle. Scaling this up to the countless packages delivered on a daily basis and the demand for ever increasing speedy delivers, the action of latching and unlatching needs to be done quickly and reliably.

The latching mechanism must be able to interface with cardboard boxes, assumed in Chapter 2, as well as stiff wall-mounted latching patterns in delivery trucks. Many mechanisms exist that try to attach and detach similar to LIMMS. Applications range quite widely from tool changers proposed in [35] to space shuttle docking solutions in [36–38] to modular, self-configuring robots [39–45]. The underlying method for achieving successful attachment and detachment of two different objects also ranges widely: spring-loaded pin locks, magnetic locks, and pneumatically actuated locks to list a few. For many of these technologies, the mechanism relies on intricate designs or gendered connectors to help with alignment such as ball plungers [46], opposing grip claws [47], and floating platforms [15] among others.

For the application of last mile delivery these mechanism have distinct shortcomings when compared to the LIMMS latching mechanism introduced in this chapter. Most of the mechanisms described above offer reliable solutions for their particular use case, but they tend to be overly complex which either increases the mechanisms weight, e.g. additional actuation to assist with alignment, or increases their spatial footprint, e.g. large geometric grooves for passive self-alignment. Despite sacrificing being light-weight for these features, the number of initial conditions for which latching will be successful or *area of acceptance* is still quite small [48]. As a result, relative to the actual mechanism's size, the target area when aiming for alignment tends to be small, requiring very precise controllers to position and orient when mating the surfaces. This results in the process being very slow.

In addition, there are several operational restrictions LIMMS has during last mile delivery in which more common approaches fail. The most obvious issue being that one mating surface is limited to being a cardboard box without any complex expensive to manufacture components, since the boxes will be considered single use. Any modifications done to the box for the purposes of latching need to have a small spatial footprint and be light weight to reduce energy expenditures and, thus, delivery cost. The latching mechanism at the end effector of LIMMS should similarly have a small volume with respect to affecting the overall available space within a box, e.g. the latch should not protrude too deeply into the box with respect to its mating surface. The mechanism should also be light weight, since extra weight at the end effector will increase the energy consumption and decrease its carrying capacity and locomotion capabilities.

Given these restrictions for the latching mechanism, LIMMS integrates a female latching pattern into the sides of the cardboard box. The simplest version of this design cuts planar features and folds them into the box, creating flaps that can assist with alignment, depicted in Fig. 3.2 and Fig. 3.3 (*right*). Another version of the female latching patterns that is also introduced in this section simply cuts the flaps, creating small holes for the latch to engage with as seen in Fig. 3.6. This is low cost, does not add any weight to the box, and can be easily added to preexisting manufacturing processes.

Typically, the slowest portion is alignment of the mating surfaces. This re-

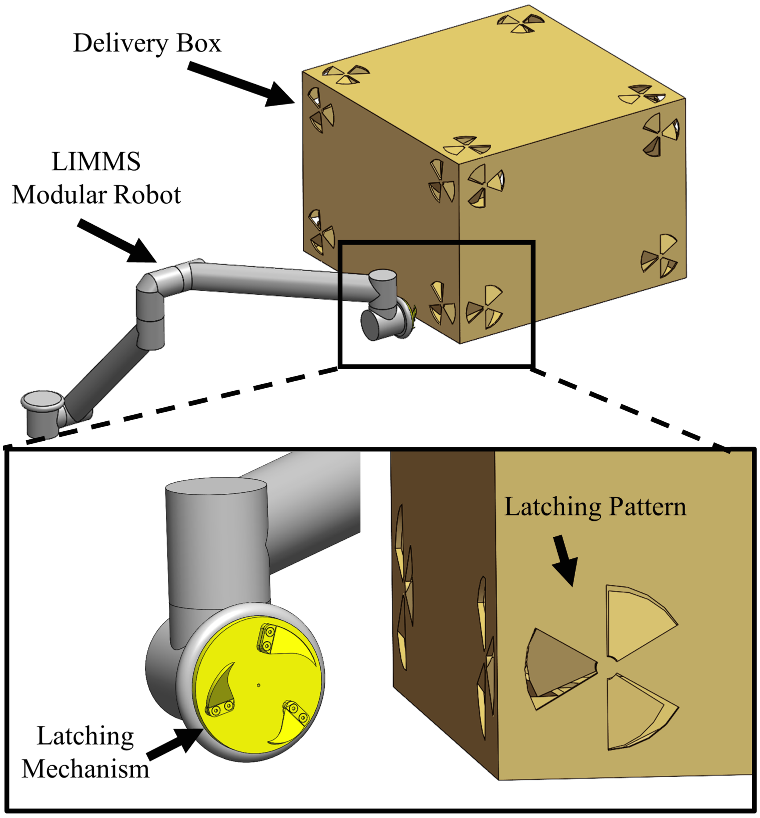


Figure 3.1: Depicts a 3-blade latching mechanism on LIMMS module with a box with corresponding latching patterns near the corners of the box.

quires a fine tuned control capable of rejecting large disturbances and noise. Considering how variable packages and the environment could be for LIMMS, developing a self-aligning rotational mechanism that admits a large region for successful latching with minor assumptions would be ideal. This reduces the burden placed on the controller and, ultimately, achieves faster alignment.

A lightweight radial design with radial symmetric blades can mechanically admit such an area. By rotating the last joint, the blades rotate and can engage with the female pattern on the box. As the blades contact the edges of the pattern, it rotates and generates a self-correcting force that passively centers itself, assuming the joints of LIMMS is less stiff than the last joint. Since this only depends on blades for alignment, mass at the end effector is negligibly affected the addition of the blades.

In this section we introduce a framework for these types of self-aligning mechanisms. Under certain assumptions, the geometry and mechanics can be analyzed to inform designs that maximize misalignment tolerance. The analysis is limited to radially symmetric blade patterns with point contact models for the tips that engage with the mating surface of the box. For any position on the 2D mating plane and a random orientation of the blades parallel to that surface, design parameters are derived to ensure the highest chance of successfully latching over the largest area. The larger the area of acceptance is the less control effort is needed on the alignment phase of latching. Though this framework uses many assumptions, it can be extended to other designs and weaker assumptions.

Most of the detailed misalignment sensitivity analysis was carried out for 2-blade and 3-blade latch designs, since it is clear by observation from experiments shown in Fig. 3.14 that total area for successfully latching exponentially decays. A metric defined by summing over the area of distribution of successful latching in the cardboard (X-Y) plane can then be used to calculate optimal design specifications for the male and female latching components. With this, a latching mechanism with a 2-blade design is shown to have a misalignment tolerance of 3 times the blade tip radius, which is a much lager distance than common designs with 3 or more blade-like contacts can achieve. The following summarizes the contributions in this section:

1. Introduced a class of self-aligning mechanisms,
2. Characterized their mechanics and likelihood of misalignment for the 2-blade and 3-blade designs,
3. Formulated an error tolerance metric by which optimal design specifications can be determined, and
4. Verified results in simulation and hardware.

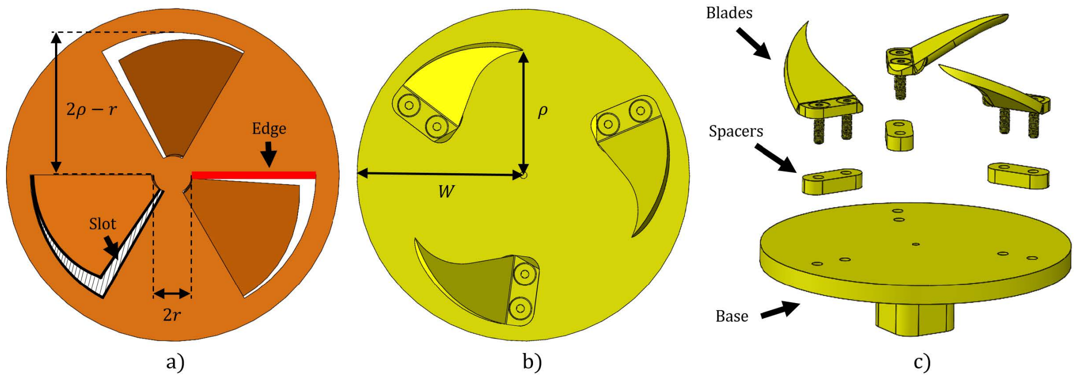


Figure 3.2: (a) Latching pattern cut from cardboard with red lines denoting engagement edges where the blades contact when fully engaged, (b) Latch assembled configuration, and (c) Components of the latch in blowout view.

Using the theory and framework developed in this section, a latch was developed for LIMMS that far outperformed other existing 6-DOF latching and methods. The underlying maths highlights the significance of key characteristics for passive alignment that have been largely overlooked in most designs. For example, many similar existing mechanisms and literature mentioned above have 4 contact points, such as hooks, pins, blades etc., whereas the models from the proposed framework suggests less contact points can be exponentially better. This makes intuitive sense as more contact points means more constraints, which cuts down on the number of solutions. For LIMMS, we show that a 2-blade design is preferable for alignment. Findings, such as these, may even be beneficial for non-rotational mechanisms.

3.1 Problem Setup

The framework as well as latching mechanism should be considered independent from the LIMMS application. Therefore, the proposed methodology can be taken for a class of self-aligning mechanisms. The problem statement of interest can be summarized with the following criteria:

- Self-alignment at a fixed position is unique up to a finite number of rotational symmetries,

- Actuation only occurs about the latch's axis of rotation,
- Design parameters should be optimal with respect to any two of the following constraints: space, strength of mating surface material, and error tolerance.

The design process developed and introduced in Section 5.4, derived in Section 3.3, is what the final bullet point's parameters is referring to. Choosing any two parameters with respect to the latch's total workspace W, the radius of the center non-hole portion of the pattern r which is dependent on the strength of the material, and Ψ a new metric for quantifying misalignment tolerance, fully defines the set of optimal design parameters.

For convenience a few terms used throughout this section need to be mentioned to avoid confusion. The *hole pattern* as seen in Fig. 3.2 (*left*) refers to the female component of the mechanism on the box, and *latch* as seen in Fig. 3.2 (*middle* and *right*) refers to the male component with symmetrically spaced blades though it can refer to any feature that engages with the hole pattern. The non-hole portion at the hole pattern's center shown in Fig. 3.2 (*left*) is referred to as *center island*. Different combinations and variations of these terms may arise.

3.1.1 Assumptions

Most of the mathematical proofs and theoretical concerns only apply to self-alignment and deriving optimal designs for the latch and hole pattern parameters. These derivations depend on a few minor assumptions that can be relaxed under certain circumstances, given by:

- The latch rotates with a counter-clockwise torque.
- The holes of the latching pattern are radially symmetric pieces of a circle with colinear edges extending to R.

- The latch face and hole pattern face are always parallel with one another and interact primarily in-plane with negligible out-of-plane effects.
- The forces applied to the latch in-plane during engagement are minimal.

Most of these assumptions can be relaxed in a form or changed. For example, in the first bullet the analyses does not change based on the direction of rotation. The resulting patterns are mirrored though. There exist patterns where the analyses is similar for shapes that differ in the hole patterns, as seen in Fig. 3.2 (*left*) outlined in the second bullet. The third assumption can be extended to handle nonparallel cases but a set of looser assumptions are required as will be explored in the latter sections. Also from hardware tests run in Section 5.4, the mechanism can successfully latch despite clearly having out-of-plane misalignment. The last point is also not a strict condition. From the hardware experiments, it is clear that the robotic arm applies forces in X and Y. Please see this video link Footnote 1 for experiments.

Several assumptions were made about the physical dynamics of the latch as the blades interact with the pattern's cut slots. It is assumed that the blades slide along the cardboard pattern's surface and fall into the engagement slots, ultimately resulting in a line of contact between the slot's engagement edge and the blade spacers, shown in red in Fig. 3.2 (*left*). Thus, it is also assumed that the shape of the slot itself ultimately does not matter for final engagement, other than the red engagement edges. These edges are radially symmetric intersecting at the center. For the 3-blade design, it is kinematically impossible to be greater than one latch radius away, since being greater than one radius away would result in one of the blades being constrained to the same hole or outside of the hole pattern region.

Due to these assumptions, in addition to this kinematic constraint, full cutouts for the 3-blade are not drawn and are instead treated as a line as seen in Fig. 3.4

and Fig. 3.12. This lack of dependence on overall slot shape is verified in practical testing. However, this is not the case for the 2-blade design. In [22] we used a conservative model for the 2-blade latching where only the final alignment edges when the blade is fully engaged were considered. We extend this to include all edges.

Furthermore, we make the assumption that there is minimal penetration into the hole pattern. Additionally, once a blade is inside a hole pattern it is constrained to be within a hole pattern. Including these two assumptions gives us a more realistic model and surprisingly substantially increases the total probability as seen in Section 3.3. Again, this phenomenon is not seen in the 3-blade assembly due to it being kinematically overconstrained.

In Section 3.3.4 derivations, it is assumed that the blade's radial location can be approximated as a single point, depicted by green dots in Fig. 3.3 (*right*) and Fig. 3.4. In the case of three or more blades, the blade spacer and the engagement edge will only contact at a single point until the latch is fully engaged due to the spacer's rounded shape. Additionally, the width of the spacer can be ignored since the blades have a triangular shape. If the blades are rotated off-center, the sloped surfaces of the blade will contact the pattern edges and force its overall location to adjust until it's aligned with the blade's tip. We tested a mechanism with three pins to verify that this assumption is valid as seen in the supplementary video[1].

In later derivations for Section 3.3, it is assumed that the shape of the blade does not matter beyond the requirements laid out in Section 3.3.4. For this reason only the point locations of the blade tips and where they start are considered.

In all derivations, the mating pattern surface is assumed to be much stiffer than the forces it is subjected to. This assumption needs verification based on the application and materials involved. In our case, the cardboard's strength is high

---

[1] Experiments, verification, and a brief explanation of this mechanism and associated publication [23] can be found online at: https://youtu.be/W5_3vF3nT28

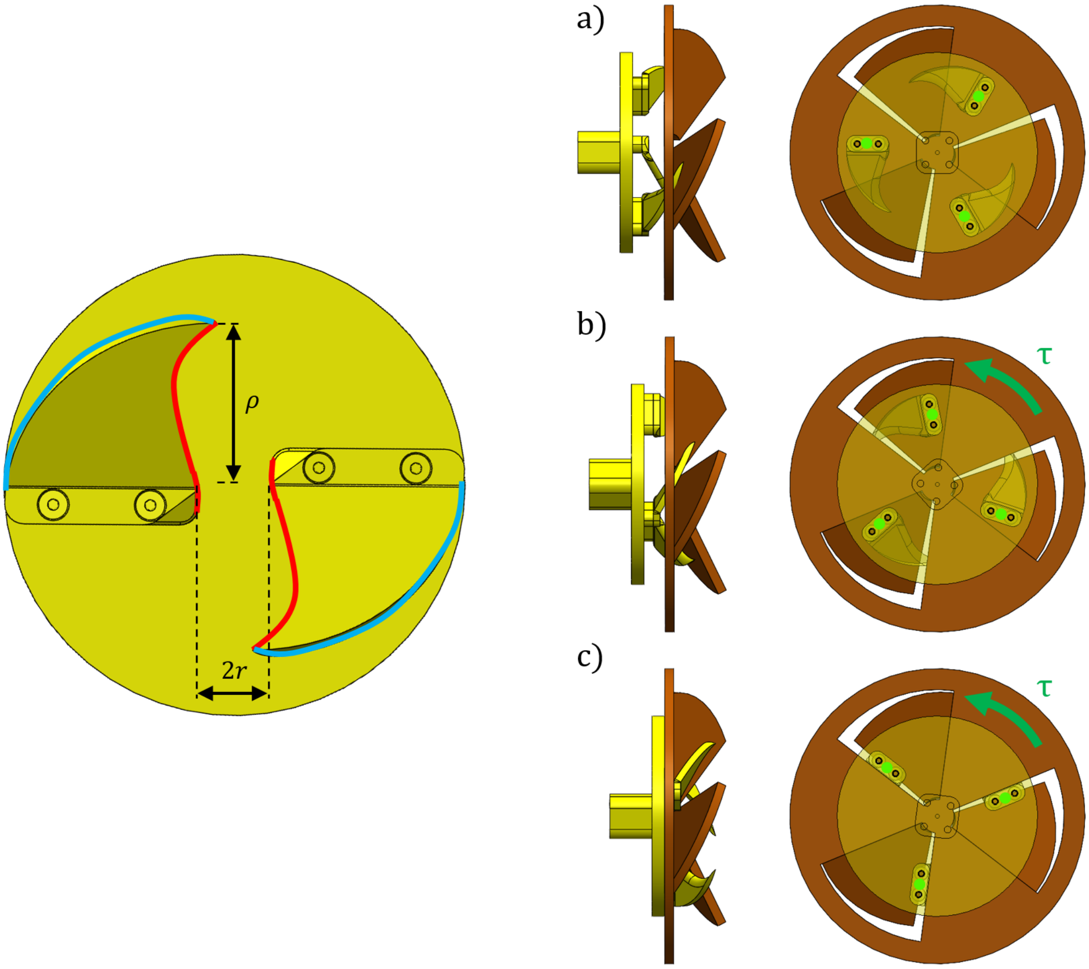


Figure 3.3: (*Left*) Two blade design showing inner (red) and outer (blue) edges that correspond to inner and outer alignment methods. (*Right*) Shows a sequence of a three blade design spinning to align.

enough for this assumption to be valid.

Finally, only the 2-blade and 3-blade cases for the latch were considered. Preliminary analyses showed that for more than three blades, the error tolerance is worse, and there are no additional benefits. For successful alignment, each blade must start its rotation in a separate angular region, since two blades cannot enter the same engagement slot. Since blades must be evenly spaced over 2π rads, increasing the number of blades effectively decreases the angular region available for each blade to start at for successful alignment. Also, a single blade latch is not able to meet the self-aligning requirements as it cannot both constrain the angle and the position of alignment when radially inserted into the mating surface.

3.2 Alignment Mechanics

This section analyzes the mechanism's sequence of operations and self-aligning motion when the latch is actuated to rotate about its center. For this analysis, it is assumed that the starting configuration for latching is with the blades lightly pressed against the surface of the cut pattern at some initial position (r', θ) and angle of rotation about the latch's center axis γ similar to the configuration in Fig. 3.3 (*right*, a).

3.2.1 Alignment with Two Blades

The 2-blade latch design consists of blades with an angular separation of π rads as shown in Fig. 3.3 (*left*). As the two blades rotate and travel along the surface of the cut pattern, the first blade that contacts the pattern's edge becomes the new pivot point that the latch begins to rotate about.

Once the second blade engages with an edge in the opposing hole, the latch moves along the two aligned edges at the latch's centerline. With only two blades, it would appear impossible to fully constrain the position and rotation, r' and θ, given only a rotational actuation about its center. However, the final constraint needed for a self-correcting motion comes from the blade profiles as the latch inserts its blades further into the the pattern.

This self-correcting motion comes from the contact force between the blade and the engagement edge as it rotates. This contact can occur at the blade's inner edge or outer edge. When the inner edge of the blade presses into the center island of the hole pattern, this is referred to as *inner alignment*, with the inner edge depicted in red in Fig. 3.3 (*left*). *Outer alignment* is when the blade edge furthest from the center pushes on the outer edges of the hole during engagement, depicted with the blue edge in Fig. 3.3 (*left*). As the latch continues to rotate, self-alignment occurs until the faces of the latch and hole pattern are touching. The latch then becomes fully constrained due to the width of the blade.

Inner Alignment Since inner alignment uses the inner edge of the blade, the blade needs to taper from its maximum allowable distance from the center W at the tip of its blade to r. Let $f(t)$ describe the inner edge contour, with $f(0) = (\rho, \theta)$ and $f(1) = (r, 0)$. As long as this contour follows the property that $||f(t)||$ is monotonically decreasing, then alignment will be successfully achieved.

Outer Alignment For outer alignment, no center island is needed, and the blade uses its outer edge of the hole pattern to align. In this case, the blade tapers out from its minimum distance from the center ρ at the blade's tip to W when fully aligned. This contour has a $f(t)$ where $f(0) = (\rho, \theta)$ and $f(1) = (W, 0)$. If the contour obeys the property that $||f(t)||$ is monotonically increasing, then alignment will succeed.

3.2.2 Alignment with Three Blades

The process of aligning with 3 blades can be broken down into four stages: no blade engaged, then one engaged, followed by two, and then finally all three. Note that it is not required to start at the first stage, although it is the most likely case. The blade locations are modeled as points, shown in green in Fig. 3.3 (*right*) and Fig. 3.4 and mentioned in Section 5.2.

The behavior of the first 2 stages closely follow the mechanics for the 2-blade case, where the first blade that contacts the edge of an engagement pattern becomes the new pivot point that the latch begins to rotate around. The behavior of this pivot point and the direction of velocity can be explicitly stated using instant

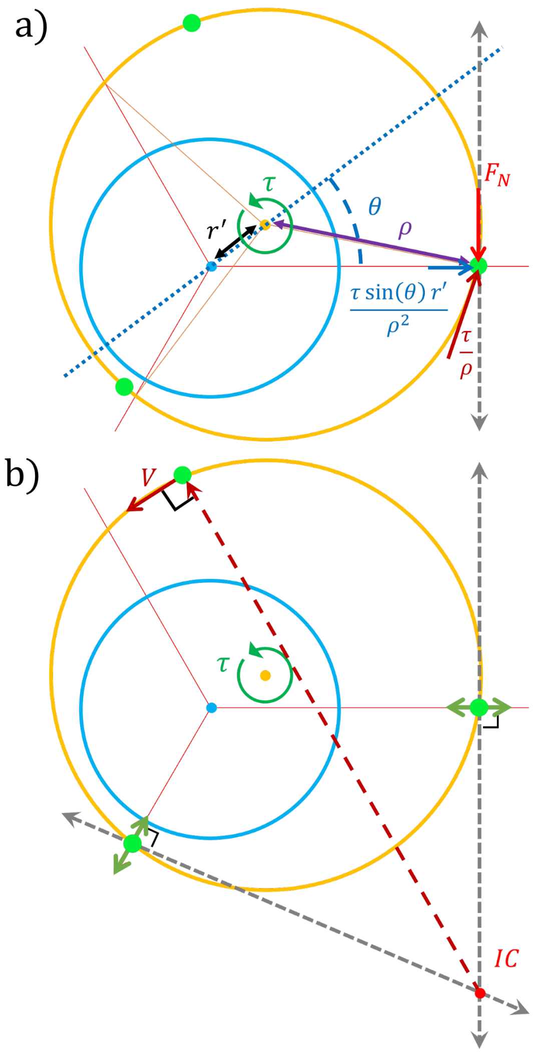


Figure 3.4: Visual representation of 3 blades (bright green dots on orange circle) aligning with the hole pattern center island (blue circle). a) First, a single blade engages the cutout edge (red) shown by the green dot on the far right. b) A second blade engages, resulting in a V in the direction of alignment. The dark green arrows depict the edge of constraint in which the blades can move given the torque direction.

centers, IC.

From the starting configuration, the mechanism first spins until one point makes contact with an edge. As torque τ is continued to be applied, the resulting forces can be seen in Fig. 3.4a. The figure further shows that the IC is not fully constrained since there is only one line (dotted grey) perpendicular to hole pattern edge (red line). In other words, the IC would need another grey dotted line that intersects to define a fully constrained point. Assuming $\mu F_N \geq \frac{\tau \sin\theta r'}{\rho^2}$, the far right green dot in Fig. 3.4a will be a stationary point, making it the IC. The blade assembly (large orange circle) rotates like a wheel around that point until the blade (green dot) contacts with the second edge. Even if slipping occurs, the constraint line (dark green) would cause the IC to be on the grey line, which would lead to the second point engaging (green dot).

Once two points are in contact, the points are only free to slide along the edge of the cutouts as shown in Fig. 3.4b. The IC for these two velocities appear in the lower right of the figure indicated by a red dot. The direction of the resulting velocity V is perpendicular to the line through the IC. This will continue until the third and final blade engages.

Finally, the three points provide three independent constraints, fully defining the latch's position and orientation given the axis of rotation and direction of τ. In this state the F_N from the edges are equal and opposite to those caused from τ, so the latching mechanism cannot move and is fully constrained at the target alignment. Fig. 3.5 shows a simple demonstration with an early prototype of a 3-blade design.

Hardware Experiments 3-Blades Our initial prototype, as seen in Fig. 3.1, has 3 blades that rotate into a cutout pattern of 3 triangular holes. The blades are sloped to pull the mechanism closer to the mating surface as the assembly is rotated. The spacers, as seen in the blowout view of Fig. 3.2, are used to establish contacts with the edges of the triangular patterns to constrain the blade to linear movements along the edge face as the latching sequence progresses. The goal is that once all spacers have made contact with their respective mating edges, the assembly and the cutout patterns should be concentrically aligned and mechanically constrained.

One possible alignment and mating sequence is shown in Fig. 3.5 where the mating surface is made of clear acrylic to better illustrate the process. The latching mechanism approaches the mating pattern in Fig. 3.5a but in gross misalignment as only one of the sloped blades successfully enters a triangular cutout. The entire assembly is rotated clockwise until the sloped blade makes contact with the edge of the hole (highlighted in red). Upon contact in Fig. 3.5b, further rotation produces two movements: 1) clockwise rotation of the latching mechanism about the point of contact, bringing the second blade into alignment with the triangular cutout in the bottom left and 2) moving the mating surface along the slope of the blade, pulling the latching mechanism closer to the mating surface.

Subsequent rotation causes the edge to contact with the spacer at the bottom of the blade as in Fig. 3.5c. In this state, the edge of the triangular cutout and the flat face of the spacer are flush and held together by the clockwise torque of the assembly. Due to the angle between the triangular patterns, the latching assembly is constrained to travel along the contact edge, moving in the upper right hand direction. Meanwhile, the second blade has now moved into the second triangular cutout, pulled the mating surface in, and the spacer has made contact with the second triangular edge.

With these two linear constraints, the final clockwise rotation will align the assembly, causing the last blade and the last spacer to make contact and fix the mating pattern to the assembly face as in Fig. 3.5d. With this the alignment process is complete and LIMMS is attached to the mating surface. It is important to note that while the above explanation was broken into steps, the entire procedure with the hardware occurred at a continuous speed without stopping.

Another important distinction is that with this prototype, the assembly must still apply torque to maintain the contact. The main goal of the prototype was to verify the effectiveness of the alignment procedure. To prevent radial movement and lock the latch in place, hooking blades that extend from the latching mechanism through the mating surface is used and discussed in latter portions of

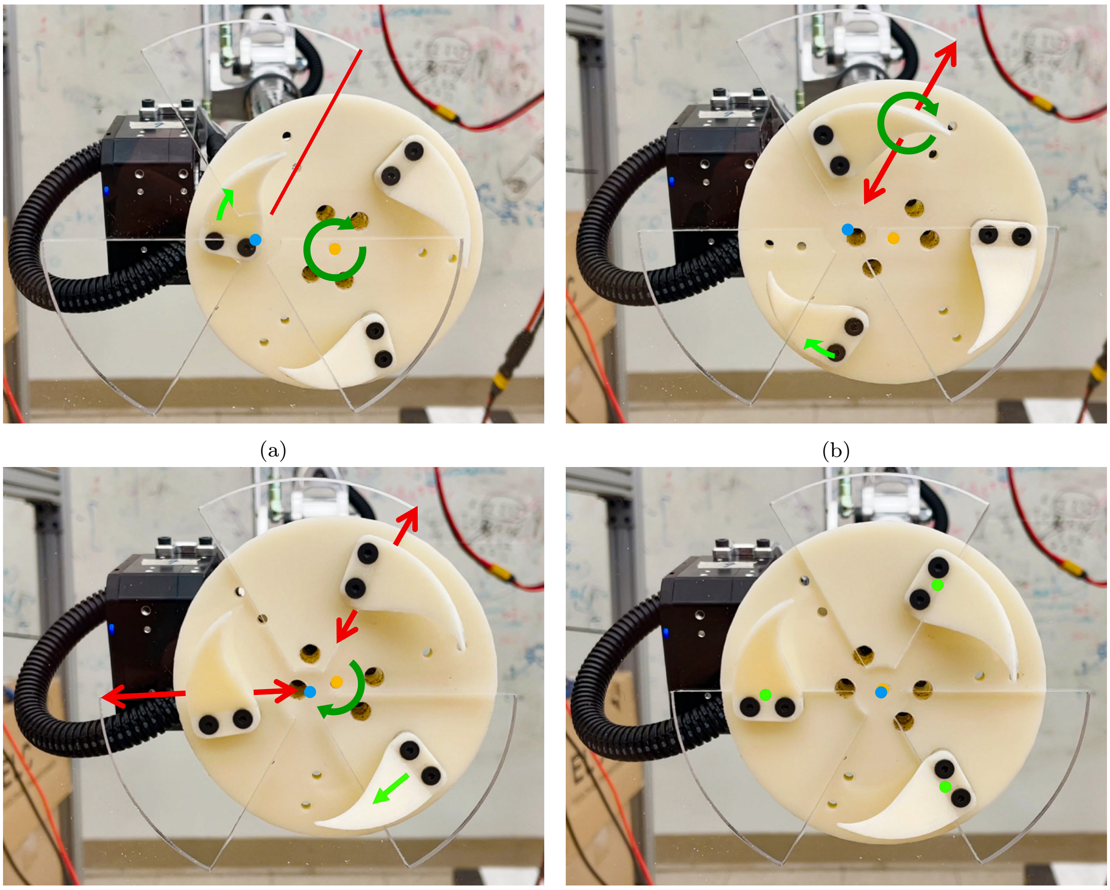

(a)
(b)
(c)
(d)

Figure 3.5: Early LIMMS latching prototype self-aligning with initial misalignment into a sheet of acrylic with three cutouts: (a) Initial position with latch center (orange) rotating (dark green) until blade moving in direction (neon green arrow) until hits mating pattern edge (red) centered at (blue), (b) Contact blade becomes a point in which the mechanism can pivot about until the second blade engages, (c) Two blades are constrained to moving along the red lines with arrows until the last blade makes contact, and (d) All blades are in contact (neon green dots) fully constraining the mechanism in an aligned position (orange and blue dots are concentric).

this chapter. Thus, LIMMS will be free to use the end actuator to rotate while maintaining the fixture to the surface.

In addition to the experiment in Fig. 3.5, testing with cardboard mating surfaces was conducted and was successful for initially misaligned configurations. A key design feature of the 3 blade design is to place loads only on the radially outward edges of the hole pattern. This prevents the center *island* connecting all three holes on the mating surface from experiencing direct loads and causing damage to the pattern. Based on the payload, the center *island* can be increased

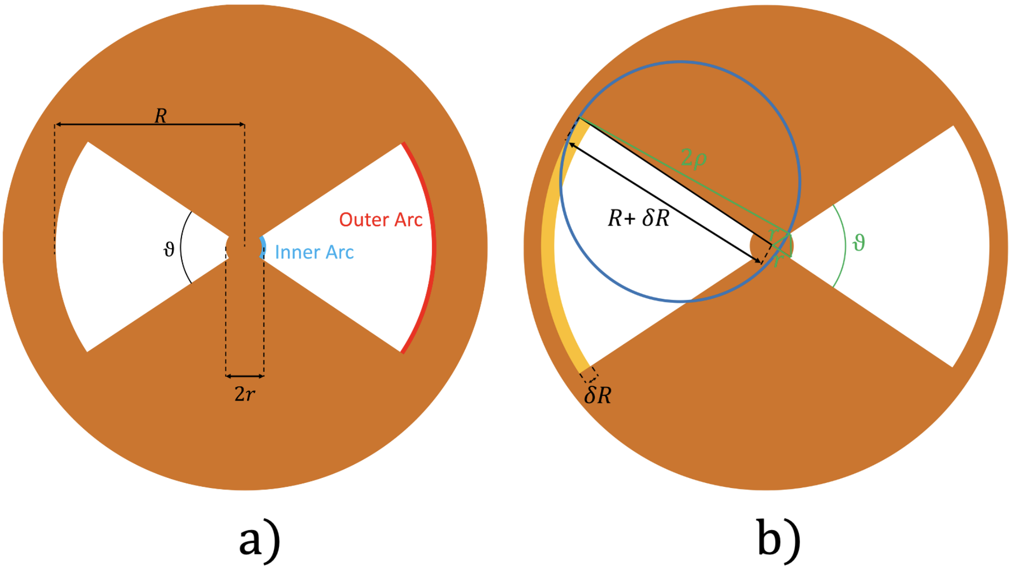


Figure 3.6: a) Diagram showing all the parameters that define the latching pattern for the two blade case. b) $R + \delta R$ indicates the maximum length that R can be. The features in green are used to calculate the length.

in size to ensure structural stability. For the purposes in the experiment, an 8 mm radius was sufficient.

In Section 5.4, Fig. 3.16 has several visual frames for the 2-blade design experiments. It is reserved until later chapters to contrast the difference in capabilities after all the theory has been derived.

3.3 Alignment Tolerance Analysis

One of the key features of our latching mechanism is its robustness to misalignment. Given an initial position (x, y) or (r',θ) in polar, and a random orientation γ of the blade assembly about its center axis, a probability distribution over the area of the cutout mating surface can be derived. In [22] we demonstrated the superiority of the 2-blade latch by comparing a conservative model of its probability distribution of latching successfully with that of the 3-blade latching mechanism in closed form. As an extension of [22], this analysis includes all four edges as show in Fig. 3.6a that fully define a hole in the pattern. Note that this figure depicts many of the parameters to be used in this section. In addition in contrast

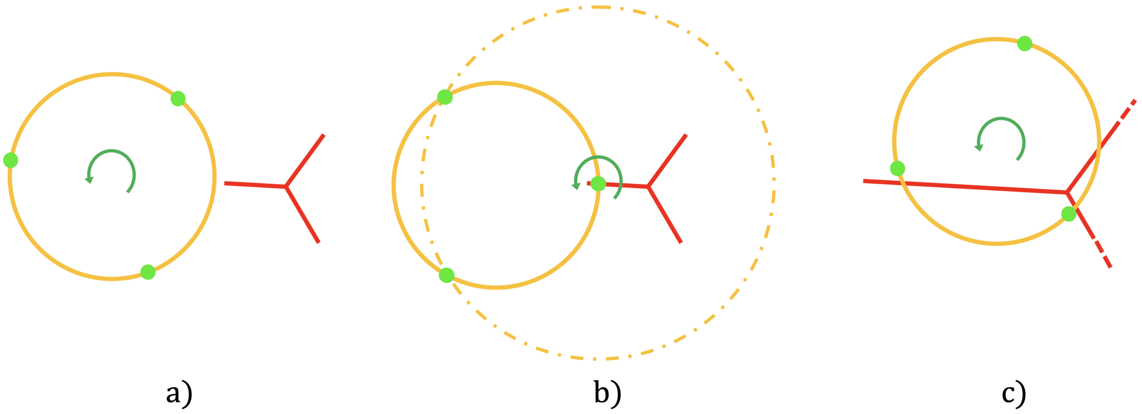


Figure 3.7: Failure types: a) Type 0 where no edges engage with the pattern, b) Type 1 where a single blade engages but the resulting trajectory causes the other blades to spin outside of the pattern, and c) Type 2 where 2 blades end up in the same region or 2 blades constrain the latch to a failure case.

with our previous analysis we highlight the importance of ϑ, the angle from which the edges expand radially. In the 3-blade case the probability distributions stay the same as it cannot take advantage of the edge defined by the outer arc, since it would simply be constrained to only failure cases. However, the 2-blade case does not have this issue, and its overall probability drastically increases. In this section we first prove that a set of optimal parameters exist and show how to obtain them for the 2-blade case. Then we show how suboptimal parameter choices can affect the probability distribution. Finally, the derivations for the probability distributions for the 3-blade design are used to draw contrast to the robustness of the 2-blade design.

To draw a fair comparison between inner and outer alignment the maximum area for the latch is defined by radius W, island size r, pattern angle by ϑ, and the integral of the probability distribution for successfully latching. Successfully latching means that each blade engages edges within different holes of the pattern. The number of failure modes varies based on parameter choices. The types of failure modes are defined as follows:

- Type 0: No blades engage with any edge,
- Type 1: One blade engages with one edge, while the other cannot engage

with any edge, and

- Type 2: Two blades engage edges within the same hole.

3.3.1 Two Blades

The probability distribution for the 2-blade design depends on certain parameters for both the pattern and the latch. The pattern is defined by r, R, and ϑ, and the blades are defined by ρ and W. The parameters ρ^*, R^*, and ϑ^* can be converted to only depend on r^* and W^*. In this section the optimal parameters for 2-blade inner align are derived for R^* in a bounded set. Outer align derivations follow similarly.

3.3.2 Inner Align

Proposition 1. For 2-blade inner align: $\rho^* = W^*$.

Proof. By definition of W it follows: $\rho \not> W \;\forall\; \rho, W$. Therefore, we consider if $\rho < W$. There exists some scaling between R and ρ such that simply increasing both by this factor would increase the overall maximum area of successfully latching and the total probability mass. Therefore, $\rho^* = W$ and vice versa following the same logic for W^*.

$\square$

Lemma 1. *For 2-blade inner align:*
$R^* \in [2\rho - r, \sqrt{\frac{1}{2}(r^2 \cos 2\vartheta - r^2) + 4\rho^2} - r(\cos\vartheta - 1) - r]$.

Proof. If $R < 2\rho - r$, then simply increasing R increases the total probability of successfully latching. As R increases the number of Type 0 failures strictly decreases since the maximum distance of engaging an edge increases. Some Type 0 failures get converted to Type 1 assuming small enough R. As R increases, Type

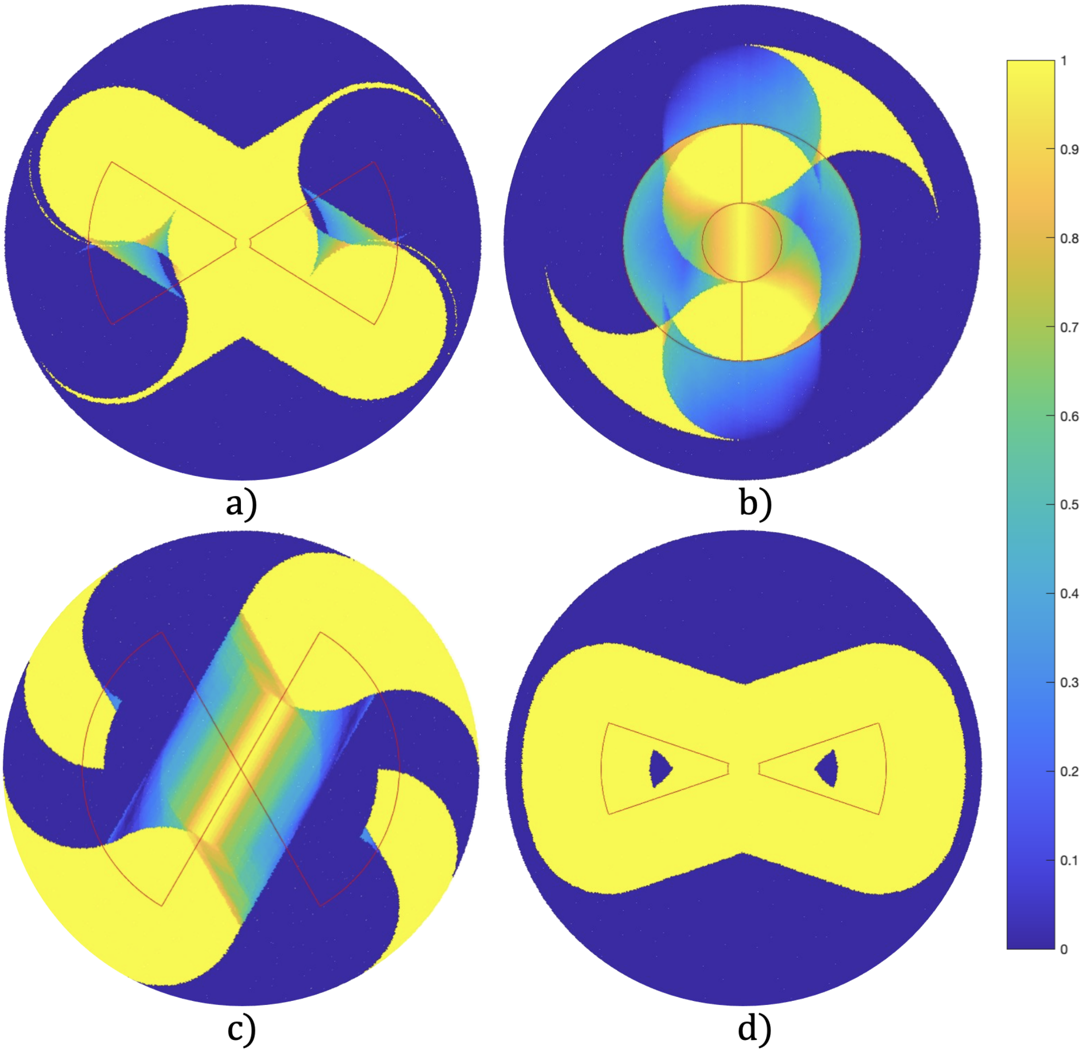


Figure 3.8: a) Probability distribution when R is bigger than the upper bound from Lemma 1. b) Probability distribution when ϑ is almost π. c) Probability distribution when $r = 0$ and $\vartheta = \frac{2\pi}{3}$. The area of high probability is mostly out to the corners, which is not ideal for controllers. d) Simulation showing how the probability distribution decreases when $\vartheta < \vartheta^*$.

1 failures strictly convert to successes. Type 2 failures cannot exist in this case since $2\rho \gg R$ and the blades' circular trajectory cannot fit within the pattern.

The upper limit $\sqrt{\frac{1}{2}(r^2\cos 2\vartheta - r^2) + 4\rho^2} - r(\cos\vartheta - 1) - r$ is the furthest distance away an engagement point can be in order to successfully latch as seen in Fig. 3.6b. This point is defined by the furthest corner of a hole and the corresponding nearest small corner of the opposing hole. This distance at most is 2ρ. This is then used to calculate the theoretical upper bound. If $R > \sqrt{\frac{1}{2}(r^2\cos 2\vartheta - r^2) + 4\rho^2} - r(\cos\vartheta - 1) - r$, one blade engaging on any part of the larger arc of the hole pattern, as seen in Fig. 3.6a, will result in a Type 1 failure since it can no longer reach the other hole. Decreasing R would serve to reduce those Type 1 failures. □

For our purposes, we define R^* as the following: $\widehat{R}^* = 2\rho - r$, where $\widehat{R}^*$ is a close approximation with bounded error on the true R^*. When $r = 0$, $R^* = \widehat{R}^*$ since it is independent of ϑ and r. Furthermore, since $\widehat{R}^*$ is the lower bound on R^*, the probability distribution for one half of the hole pattern (since it's rotationally symmetric by π) is a convex hull. This property provides continuous area without any break near the center as seen in Fig. 3.8a. From a controls perspective having any low probability around or near the axis of pattern engagement would be detrimental.

Lemma 2. *For* $r \in (0, \rho)$*,* $\vartheta^* \in (\frac{\pi}{3}, \pi)$*. As* $r \to 0$*,* $\vartheta^* \to \frac{\pi}{3}$*.*

Proof. First, we establish that the largest distance between any two points on the edges of a single hole d: $d = 2\rho \implies \exists \vartheta^*$. If $d > 2\rho$, Type 2 failures strictly increase since there now is a length in which two blades can fit into the same hole. As d increases more successes are converted to Type 2 failures. Similarly, if $d < 2\rho$, the number of Type 0 and Type 1 failures strictly increase since the trajectory of a latch can either completely encompass one of the hole patterns or engage with one hole pattern and encircle the other without making contact.

At $d = 2\rho$, with probability zero it will be in an initial condition where it fails, since there are only single points of failure along a continuous spectrum of real numbers. Therefore, $d = 2\rho$ is a necessary condition for the optimality for ϑ^*.

If $r = 0$, then $R^* = 2\rho$ by definition. Then the longest length $d = 2\rho$. That means $\vartheta \in (0, \frac{\pi}{3})$ and is not unique. We take ϑ^* to be the maximum value since increasing ϑ increases the overall potential to convert Type 0 and Type 1 failures to successes as the total area has increased, meaning $\vartheta^* = \frac{\pi}{3}$ from it being an isosceles triangle. The other bounds are its physical limits by definition.

□

Theorem 1. *There exists a unique $\vartheta^* \in (\frac{\pi}{3}, \pi)$ for every R^*, and ϑ^* is monotonically increasing with respect to r:* $\exists!\ \vartheta^*\ \forall\ R^*,\ r \in (0, \rho) \implies \vartheta^* = 2 \arcsin \frac{\rho^*}{R^*}$.

Proof. From Lemma 2, when $r = 0$, ϑ^* is chosen to be as large as possible to increase the total number of potential successes. The argument for the remaining interval of r follows similarly to the analysis presented in Lemma 1.

For $\vartheta < \vartheta^*$, then $d \leq 2\rho$ where the total number of failures either decrease or remain the same when ϑ is increased. Type 0 failures decrease as the area of the hole is larger. Type 1 failures remain unchanged or decrease since the possible number of intersections in opposing holes have increased. There will still be no change in Type 2 failures as $d \leq 2\rho$ except for potentially a few instances with zero probability. A probability distribution for this case is shown in Fig. 3.8d.

Now that ϑ is monotonically increasing in r and is unique, we show that this definition is optimal. For $\vartheta > \vartheta^*$ the overall probability distribution decreases. By adding a small angle, $\delta\vartheta$ to ϑ^* as shown in Fig. 3.10. We can analyze what happens when $\vartheta > \vartheta^*$. To complete this proof, a key insight was that adding δr introduced a half circle of probabilities centered at δr that changed from 1 to 0 as shown in Fig. 3.10. At the same time, adding $\delta\vartheta$ will convert part of a radius of probability from 0 to 1. An under-approximation of the probability loss δP_l and over-approximation of the probability gained δP_g are as follows:

$$\delta P_l = \delta r \pi \rho$$
$$\delta P_g = \frac{(2\rho)^2}{2}\delta\vartheta - \frac{r^2}{2}\delta\vartheta \ .$$

These regions are shown in Fig. 3.9 and Fig. 3.10b. Based on this insight the relationship between $\delta\vartheta$ and δr is critical and defined as:

$$\delta r = \frac{2\rho sin(\delta\vartheta)}{2sin(\delta\vartheta + \vartheta)} \ .$$

Using this relationship we find:

$$\delta P \leq 2\delta\vartheta\rho^2 - \frac{\delta\vartheta r^2}{2} - \frac{4\delta\vartheta\rho^2\pi}{2\delta\vartheta + \vartheta} \ . \tag{3.1}$$

The derivation for these relationships is derived below.

Derivation of the δr and $\delta\vartheta$ Relationship All definitions for angles and side lengths are provided graphically in Fig. 3.10. We note that $\gamma = \vartheta$ as they form an isosceles triangle with β. Then we can calculate β using $\beta + 2\vartheta = \pi$ and similarly β' using the new triangle defined by $\vartheta + \delta\vartheta$:

$$\beta = \pi - 2\vartheta$$
$$\beta' = \pi - 2(\vartheta + \delta\vartheta) \ .$$

Let $\delta\beta = \beta - \beta'$. Since the red-green-blue triangle is isosceles as both the red side and green side are 2ρ, ω is defined as:

$$\omega = \frac{\pi - \delta\beta}{2} \ .$$

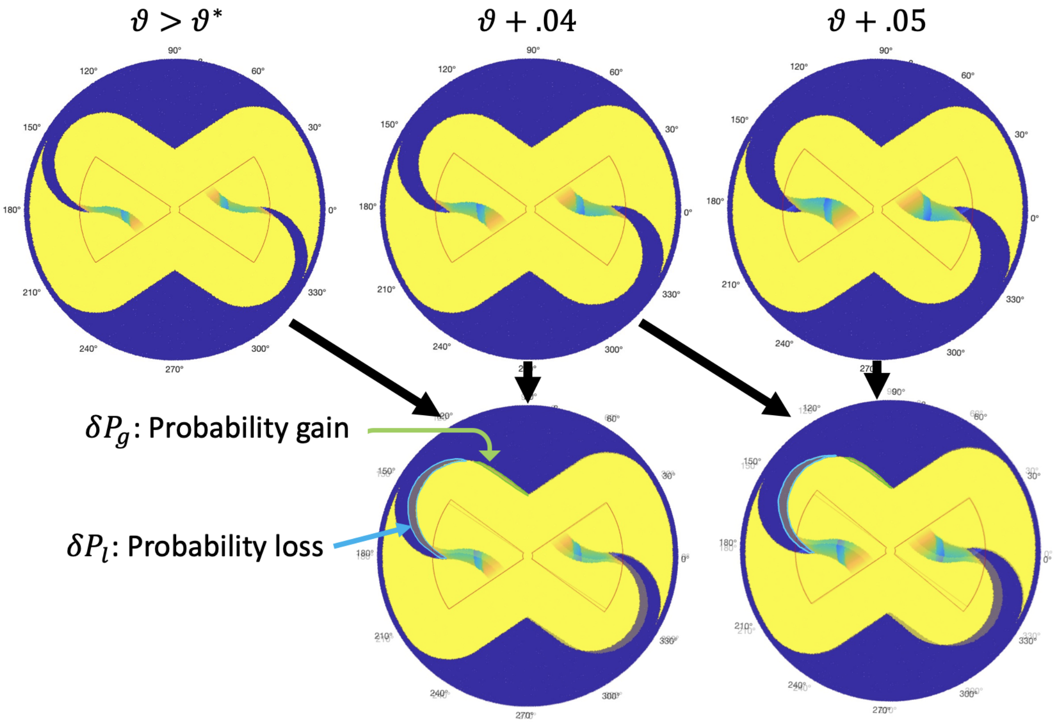


Figure 3.9: Simulations showing a progression of increasing ϑ. Note how when they are overlaid the regions described in Theorem 1 show up.

Next δg can be computed using the law of sines:

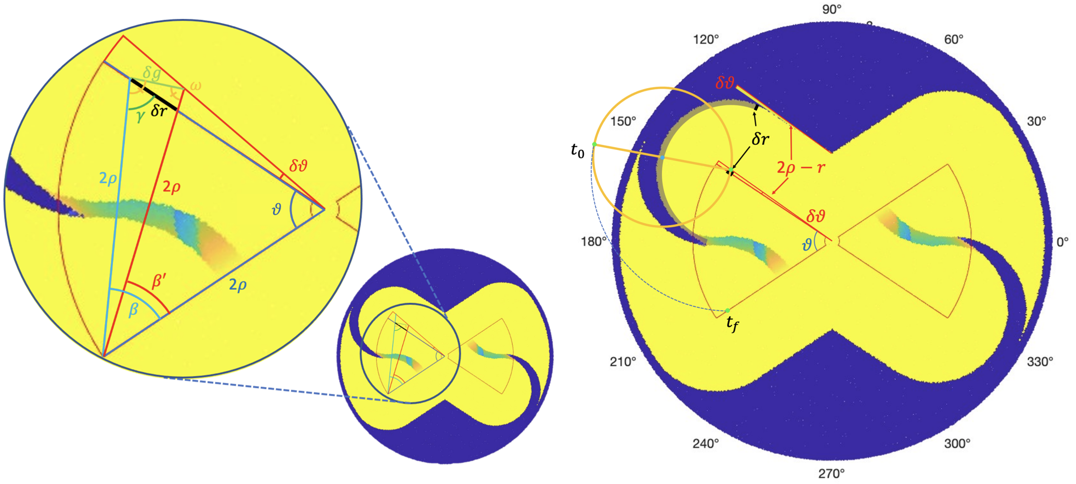
$$\delta g = sin(\delta\beta)\frac{2\rho}{sin(\omega)} \ .$$

Finally δr can be calculated using the green-black-red triangle and the law of sines:

$$\delta r = \frac{sin(\omega)\delta g}{sin(\pi - (\omega - \gamma) - \omega)} \ ,$$

Where the denominator is the angle opposite $\delta\gamma$ in the green-black-red triangle.

Derivation of δP_l and δP_g The insight for defining δr can be seen in Fig. 3.10b. If the circle defined by the path of the blade tips passes through δr then the first point will engage such that the second point will also end up in the same cutout, resulting in a Type 2 failure. One example of this is the orange circle with the green points indicating the current and future locations of the blade tips in Fig. 3.10b.



a) b)

Figure 3.10: a) Geometry used for relating $\delta\vartheta$ to δr. The red line with angle B' and the blue line with angle β is defined to have length of 2ρ. b) By perturbing ϑ^* by $\delta\vartheta$, a net loss of probability mass occurs, despite an overestimation of the probability gained, shown by the little triangle sticking out on the upper left of the probability defined by $\delta\vartheta$ and $2\rho - r$, and an underestimation of the probability loss, shown by the greyed out yellow arc in the upper left.

If δr is sufficiently small, this area can be calculated as $\delta r \pi \rho$, which is shown in the shaded region of the figure. At the same time some area is being converted from probability 0 to probability 1 as now a blade tip can reach the upper edge of the cutout removing some Type 0 failures. This area is shown as the yellow sliver sticking out of the probability mass next to the shaded region. This area can be calculated by subtracting two triangles: the smaller one defined by length r and width $r\delta\vartheta$ from the larger one defined by length 2ρ and width $2\rho\delta\vartheta$. This is an upper bound on the area of probability mass added as the yellow sliver sticking out clearly should not have been converted from probability 0 to 1 since it is at least ρ away from δr. δP_l is defined as the probability region that goes from $p = 1$ to $p = 0$.

Decreasing Probability The right-hand side of Eq. (3.1) is negative over the following sets of parameters defined by physical limitations:

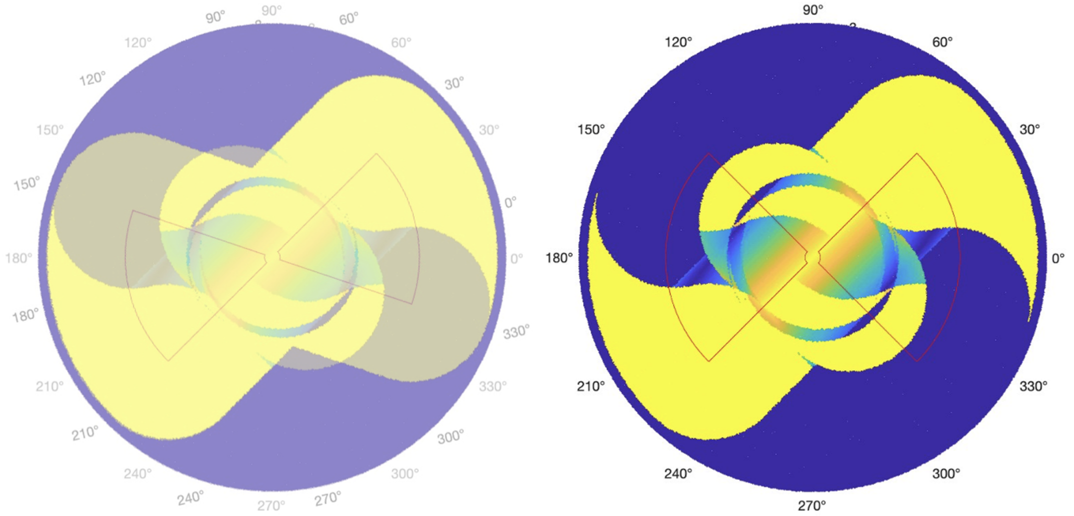
$$\{r, \rho, R | 0 \leq r < \rho \leq R \leq 1\}$$
$$\{\vartheta,\ \delta\vartheta\ | 0 < \vartheta < \pi,\ 0 < \delta\vartheta < (\pi - \vartheta)\}\ .$$

Now when substituting the appropriate upper bounds to maximize the right-hand side, we prove it is always negative on these sets:

$$\begin{aligned}
2\delta\vartheta\rho^2 - \frac{4\delta\vartheta\rho^2\pi}{2\delta\vartheta + \vartheta} & \qquad \text{(Substitute minimum } r\text{)} \\
2 - \frac{4\pi}{2\delta\vartheta + \vartheta} & \qquad \text{(Factor out } \delta\vartheta \text{ and } \rho^2\text{)} \\
2 - \frac{4\pi}{2\pi - \vartheta} & \qquad \text{(Substitute maximum } \delta\vartheta\text{)} \\
2 - \frac{4}{2 - \epsilon} & \qquad \text{(Substitute } \epsilon\pi > 0 \text{ for } \vartheta\text{)} \\
2 - \frac{4}{2 - \epsilon} < 0\ , & \qquad \forall\ \epsilon \in (0, 1)\ \text{(Bounds on } \vartheta\text{)}\ .
\end{aligned}$$

In addition, there is also a similar line of decreased probability for an arc-like area centered about the bottom corner of the pattern. It is also caused by an increase in the likelihood of Type 2 failures. This line, however, does not add areas of zero probability, as there are still starting angles that lead to successful alignments due to the rotation direction of the latch. This covers where δr increases with $\delta\vartheta$.

Once ϑ is sufficiently large, increasing $\delta\vartheta$ cannot increase δr, as the entire upper edge of the slot already results in failure. In this case, no new area is gained or lost on the upper edge or half circle around the upper corner, but the area around the lower corner still decreases as the likelihood of Type 2 failures is still increasing. Therefore, the change in probability brought about by a change



a)

b)

Figure 3.11: a) Independent probability distributions suboptimal R and suboptimal ϑ overlaid. Red hole pattern is of suboptimal R. The hole pattern for suboptimal ϑ is the same as the figure on the right. b) Probability distribution for combined suboptimal R and suboptimal ϑ. By rotating the superimposed hole pattern of suboptimal R over suboptimal ϑ from the graph on the left, the overlapping regions compose to make the figure on the right.

in angle from ϑ to $\vartheta + \delta\vartheta$ is always negative.

Intuition In Fig. 3.9, $\delta\vartheta$ is much larger than the one used in the argument for visualization purposes, which results in regions being shaped slightly differently. However, the regions of interest and arguments still follow.

The final piece of the argument relies on the independence of R^* and ϑ^*. To see this we superimposed two distributions, one with a suboptimal R and another with a suboptimal ϑ in Fig. 3.11. By rotating the superimposed image of the hole pattern of suboptimal R over the hole pattern of suboptimal ϑ, the overlapping regions compose to form the combined suboptimal R and ϑ graph.

From this ϑ^* can be defined as the largest angle such that the largest distance between any two points in a single hole pattern is exactly 2ρ. With zero probability does the Type 2 failure occur, and it is uniquely defined. Given the problem constraints the two points are exactly the points that define the ends of the outer arc. It is then trivial to solve for ϑ^*.

□

We found the approximation of R^* with $\widehat{R}^*$ for inner align did not affect performance noticeably in our practical experiments in Section 5.4. We consider deriving the exact R^* to be outside the scope of this work.

3.3.3 Outer Align

The propositions, lemmas, and theorems from the previous inner align section also apply to outer align. The arguments are the same in both cases since the model has been simplified down to contact points kinematically interacting with edges. The only difference is W because the outer align uses the outer edges to self-align and, therefore, will be larger than ρ. Additionally, as a design principle, r is always 0 for outer alignment. This is because r arises from a mechanical constraint dictated by the material and application. Since outer alignment uses the outer edge, it does not need any strength for the center island. In fact, having a center island creates an additional constraint, resulting in an adverse effect. Even if r is small, assuming the island can hold, one should just use inner align and scale the latch and pattern for the same W. Therefore, for outer align $r = 0$ is the only case considered.

Lemma 3. *For outer alignment* $R^* = W$, *and* $\rho^* = \frac{R}{2}$.

Proof. By definition to fully constrain the latch using outer align: $R = W$. If $R > W$, the latch cannot be fully constrained with the pattern. If $R < W$, there is more than one way to be constrained. Therefore, $R^* = W$ to uniquely constrain the alignment of the latch and pattern. By following the proof Lemma 1 with $r = 0$, the bounds converge to a single point $\rho* = \frac{R}{2}$, implying $R^* = 2\rho$. □

Unlike in the inner align case, we can derive the optimal set of parameters without approximating R^*. Since kinematically, the analysis of outer align follow exactly the same as the inner align section above but instead of $\widehat{R}^*$ being used the true R^* is used. We now can claim that these are the exact optimal parameters for outer align.

Theorem 2. *For outer align, the summation over the probability distribution for ϑ^* and R^* is greatest for a given ρ when $r = 0$.*

Proof. By Lemma 1

$$P(\vartheta \neq \vartheta^*, R \neq R^* | r, \rho, W) < P(\vartheta \neq \vartheta^*, R^* | r, \rho, W)$$

And by Theorem 1

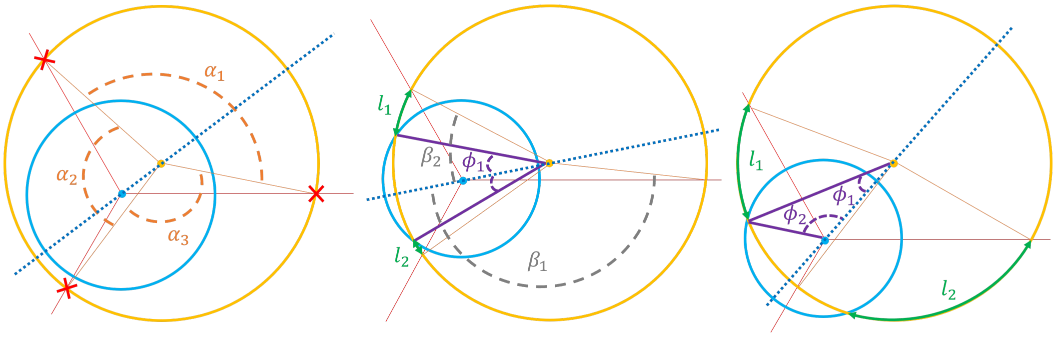
$$P(\vartheta \neq \vartheta^*, R^* | r, \rho, W) < P(\vartheta^*, R^* | r, \rho, W)$$

Therefore, the probability is the greatest for ϑ^* and R^*. □

Now that we have proved optimal parameters, we derive the probability distribution for those parameters. These distributions shown in Fig. 3.14 are calculated by taking points ρ away from the edge of the hole pattern.

3.3.4 Three Blades

For the 3-blade case, the blades must be evenly spaced for radial symmetry, so they must be located $\frac{2}{3}\pi$ apart. These probabilities are rotationally symmetric. In order to achieve successful latching in the 3-blade case, each blade needs to start in separate regions outlined in Fig. 3.4 and Fig. 3.12 and not be blocked by the center island. This is because each blade must be able to enter an engagement slot after rotating, and no two blades can physically enter the same slot. If the center of the latch is positioned such that an orientation exists where each blade is located within its own region (and will enter its own slot), then a non-zero probability



a) b) c)

Figure 3.12: The three alignment cases for the 3-blade design: a) Case 1 where there are no intersections, b) Case 2 where both intersections are either completely in α_2 or α_3, and c) Case 3 where one intersection is in α_2 and the other is in α_3.

exists that latching will succeed. If the probability distribution is defined over a single region, it can be rotated by $\frac{2}{3}\pi$ to get the full distribution.

The boundary with zero probability for a single region, i.e., $0 \leq \theta < \frac{2\pi}{3}$, is defined. For latching to be possible, when one blade is engaged and the assembly begins to rotate, the second blade must be able to reach the other side of the island to engage: $|r'| \leq \rho$.

Two additional constraints appear due to the center island area blocking any blades from entering the engagement slots, which would also result in zero probability. Fig. 3.12b and Fig. 3.12c show how the center island creates new boundary constraints given by: $y^2+(x-r)^2 \leq \rho^2$ and $(y-r\sin(\frac{2\pi}{3}))^2+(x-r\cos(\frac{2\pi}{3}))^2 \leq \rho^2$.

Case 1 $|r'|$ is sufficiently small such that the trajectory of rotation about the blade assembly's center, shown in orange in Fig. 3.12a, does not intersect the center island indicated by the blue circle. In this case, the arc length of interest is the smaller of α_2 and α_3, since it defines the first blade to violate a boundary. However, there are also blades located $\frac{2}{3}\pi$ away in the other regions. The feasible arc lengths of these blades need to be considered as well. The minimum of this is divided by $\frac{2}{3}\pi$ to give the probability.

Case 2 The trajectory when rotating about the assembly's center intersects the center island exactly twice as seen in Fig. 3.12b and Fig. 3.12c. Depending on where the intersection occurs, the probability distribution function $p(\cdot)$ varies. If the intersections are both in the same region, then it would look similar to Fig. 3.12b. In this scenario, the arc lengths of l_1 and l_2 are of interest, as well as how much of the other region is available for the blade tip, which is $\frac{2}{3}\pi$ away engaging. This is done for both l_1 and l_2, always taking the minimum of it and its offset. Once this is done all of the resulting arc lengths are summed and divided by $\frac{2}{3}\pi$ to give the probability.

Case 3 The intersections are split across sectors as shown in Fig. 3.12c. This case is similar to Case 1 where there are no intersections. The only exception is when α_2 and α_3 are blocked by the center island.

The probability function for the 3-blade case is not continuous since it must take into account different cases and take minimum values. In order to present the closed form expression of p, several intermediate variables are provided to simplify the expression, also depicted in Fig. 3.12:

$$\begin{aligned}
\beta_1 &= \frac{\pi}{3} - sin^{-1}\left(\frac{r' \sin(\theta)}{\rho}\right) - \theta \\
\beta_2 &= \frac{\pi}{3} - \sin^{-1}\left(\frac{r'}{\rho \sin(\frac{2\pi}{3} - \theta)}\right) + \theta \\
\beta_3 &= \frac{\pi}{3} - sin^{-1}\left(\frac{r' \sin(\frac{2\pi}{3} + \theta)}{\rho}\right) - \theta \\
\alpha_1 &= (2\pi - (\beta_2 + \beta_1)) \\
\alpha_2 &= \beta_3 + \beta_2.
\end{aligned}$$

For 3 blades p is as follows:

Case 1: $r' \leq \rho - r$ (0 or 1 intersections with the island)

$$p = \min(\alpha_2 + \alpha_3 - \tfrac{2\pi}{3}, \min(\alpha_2, \alpha_3))$$

Case 2: $r' > \rho - r$ (2 intersections with the island)

(a) $\theta - \phi_2 < \frac{4\pi}{3}$ (2 intersections in α_2)

$$l_1 = \beta_2 - \phi_1$$

$$l_2 = \beta_1 - \alpha_3 - \phi_1$$

$$p_1 = \min(\max(\alpha_2 + \alpha_3 - \tfrac{2\pi}{3}, 0), l_1)$$

$$p_2 = \min(\max(\alpha_2 + \alpha_3 - l_1 - 2\phi_1 - \tfrac{2\pi}{3}, 0), l_2)$$

$$p = p_1 + p_2$$

(b) $\theta + \phi_2 > \frac{4\pi}{3}$ (2 intersections in α_3)

$$p_1 = \min(\max(\alpha_2 + \alpha_3 - \tfrac{2\pi}{3}, 0), l_1)$$

$$p_2 = \min(\max(\alpha_2 + \alpha_3 - \tfrac{2\pi}{3} - l_1 - 2\phi_1, 0), l_2)$$

$$p = p_1 + p_2$$

Case 3: $r' > \rho - r$ (1 intersection in α_2 & other in α_3)

$\theta + \phi_2 \leq \frac{4\pi}{3}$ and $\theta - \phi_2 \geq \frac{4\pi}{3}$

$$p_1 = \max(\min(l_1, l_2) + 2\phi_1 - \tfrac{2\pi}{3}, 0)$$

$$p_2 = \min(\alpha_2 + \alpha_3 - \tfrac{2\pi}{3}, \min(l_1, l_2))$$

$$p = \max(p_2 - p_1, 0)$$

3.3.5 Alignment Tolerance Metric Ψ

With probability p now defined, the performance of different latch designs can be compared based on their alignment tolerance. Ψ is a scalar quantity that can help compare relative designs by summing over p:

$$\Psi(W, \rho, r) = \int_{D(0,Wc)} p(r', \theta | r, \rho, W) dA \tag{3.2}$$

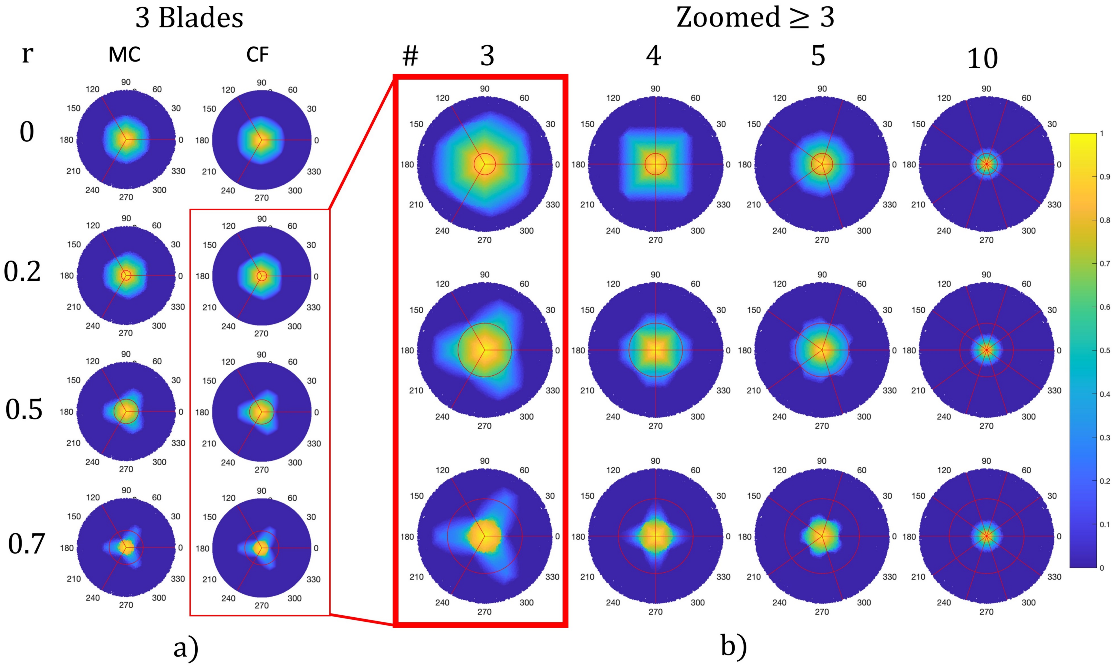


Figure 3.13: Depicts p where brighter colors represent higher probabilities of success: a) Monty Carlo (MC) experiments and closed-form (CF) expressions for the 3-blade design, with r values of 0, 0.2, 0.5, and 0.7. b) Zoomed-in figures of our MC experiments for 4, 5, and 10 blades with r values of 0.2, 0.5, and 0.7, where the 3-blade case was added to show how increasing the number of blades decreases the misalignment tolerance.

where $p(r', \theta | r, \rho, W)$ is defined above, and c is a constant based on the application. It is constrained to $c \leq 3$, as this will capture the entire distribution. Due to the complexity and piece-wise nature of p, Ψ can most easily be computed numerically. Ψ is not unitless to allow it to maintain a physical meaning for design comparisons. Comparisons are made on a relative scale for similar designs, but there are nuanced specifications in which looking at both p and Ψ could be even more beneficial.

Although not a complete one-to-one comparison since it is not unitless, we show the comparison of Ψ values for the 2-blade and 3-blade cases since the differences are in such contrast. This is used to give some intuition for what the values of Ψ should look like for certain cases. Ψ in general is non-negative and less than one. For 3-blade case where $W = \rho = 35$mm, $c = 3$, and $r = 8$mm, $\Psi = 0.0430$. For the 2-blade case with the same parameter values as in the 3-blade one, $\Psi = 0.449$. This is approximately ten times better than the 3-blade case in terms of values of

Ψ.

The best case offset from the center comes from the 2-blade case using inner alignment when r approaches 0. When r goes to 0, it becomes $3\rho^*_{in}$. Although r can never truly equal zero in a practical design, it can be reduced to nearly zero if the hole pattern incorporates a stronger material such as metal or plastic to reinforce the center island.

3.3.6 Monte Carlo vs Theoretical

For the 3-blade cases, our derivations are validated by comparing Monte Carlo simulations (MC) to our closed-form expressions (CF) as shown in Fig. 3.13. The left image, Fig. 3.13a, shows the two probability density functions side by side for comparison. The two are the nearly identical, verifying our CF results for the 3-blade case. Fig. 3.13b shows a zoomed-in view of the bottom 3-blade patterns from Fig. 3.13a and increases the number of blades in the pattern. It can clearly be seen that the larger the number of blades in the pattern, the less robust it is for handling misalignment. Inspecting Fig. 3.14 indicates that the 2-blade design has a larger overall area of feasible positions with a higher probability of success than any of the other designs with more than 2 blades.

3.4 Design Case Study

Design Framework Using Ψ defined in Eq. (3.2), three different design methods can be outlined based on known constraints for a particular application. These can be alignment tolerance Ψ^*, a minimal r based on material properties, or a maximum W based on the size constraints for the mechanism.

The designer must first decide if 2 or 3 blades work better for the application. The 2-blade design has a better Ψ but uses r to align, while 3-blades has a worse Ψ but does not use r for alignment. The flow charts shown in Fig. 3.15 show the

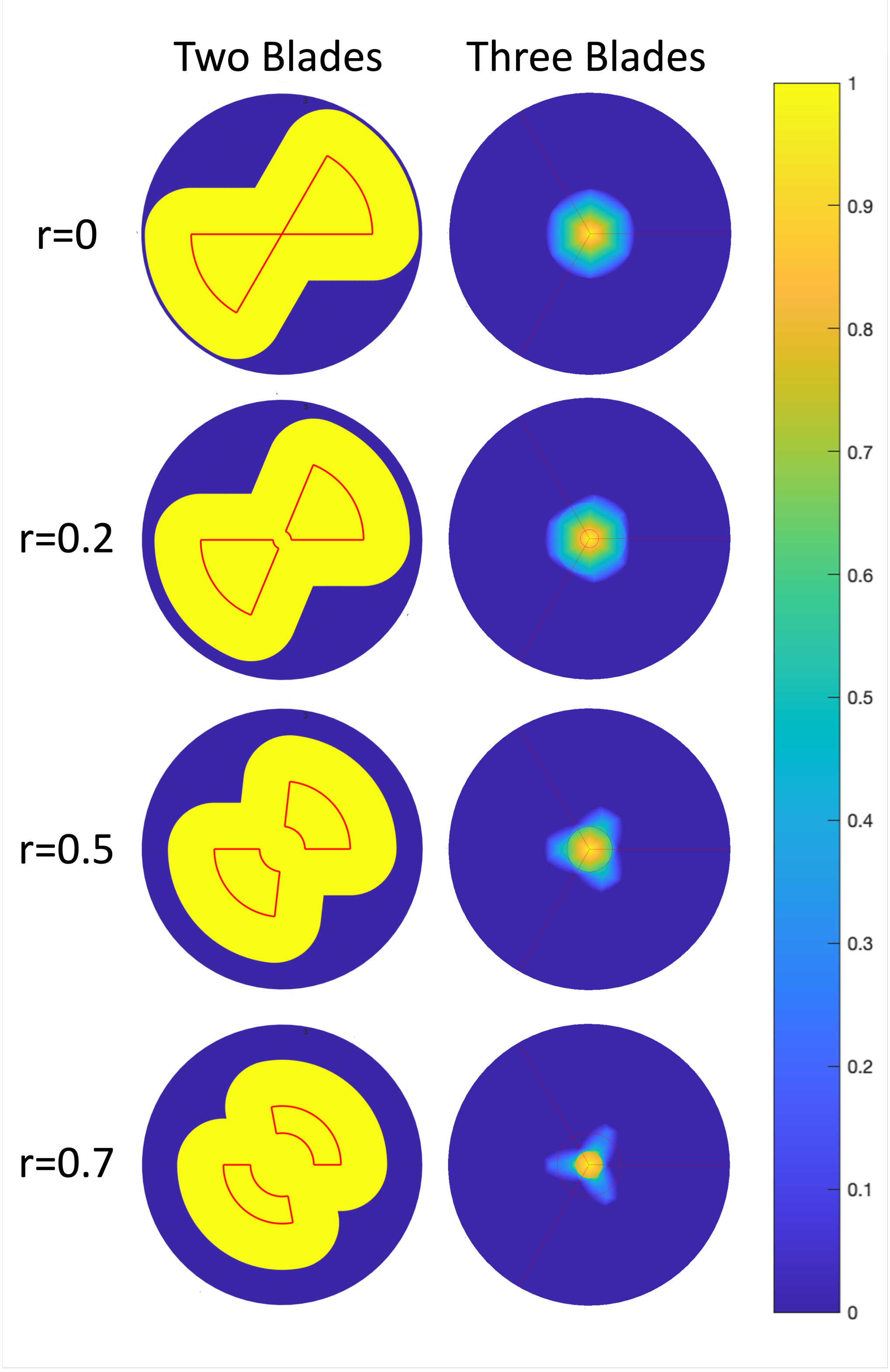


Figure 3.14: Varying 2-blade probability distribution on the left versus 3-blade design on the right. As one can clearly see, the 2-blade design is much more robust than the 3-blade design.

process for picking the best design given 2 of the 3 parameters. For the 3-blade design, inner align is similar to that of the 2-blade's.

Using Fig. 3.15a, a viable design for both 2-blade and 3-blade mechanisms for use on LIMMS was developed. We first conducted experiments and determined $r \geq 8$ mm based on the strength of cardboard. Since corrugated cardboard is an orthotropic composite material, patterns were cut with the flutes oriented the same way to eliminate effects from the material anisotropic properties [49].

Verification Experiments To verify the results of our design methodology, both the 2-blade and 3-blade latches were built and tested using two different mating surfaces. The first surface tested was cardboard, which is required for the last mile delivery problem. The cardboard chosen for experiments was C-flute cardboard with a thickness of 4.4 mm, chosen to best simulate the average properties of common shipping boxes. The second surface was acrylic, which is transparent and has a lower coefficient of friction. This was to explore the option of integrating a plastic insert with the cardboard pattern for reinforcement and to study the mechanical behavior of the latch as it rotates. A set of stills from the 2-blade testing on acrylic can be seen in Fig. 3.16. In this figure, the blue point denotes the center of the hole pattern, and the orange point marks the center of the latch. The red lines indicate the edges of the hole pattern that contact the blades. In Fig. 3.16a, the hole pattern and the latch are significantly misaligned. As the latch is rotated, the blade tip enters the slot, and the latch self-aligns with the hole pattern.

The blade mechanisms were attached to YORI, a 5-DoF robotic arm [50], to simulate the mechanism's behavior when attached to the LIMMS platform. YORI was operated using a simple PID controller with minimal gain tuning. In these tests, the latching mechanisms were able to align and hold as expected, as shown in Footnote 1.

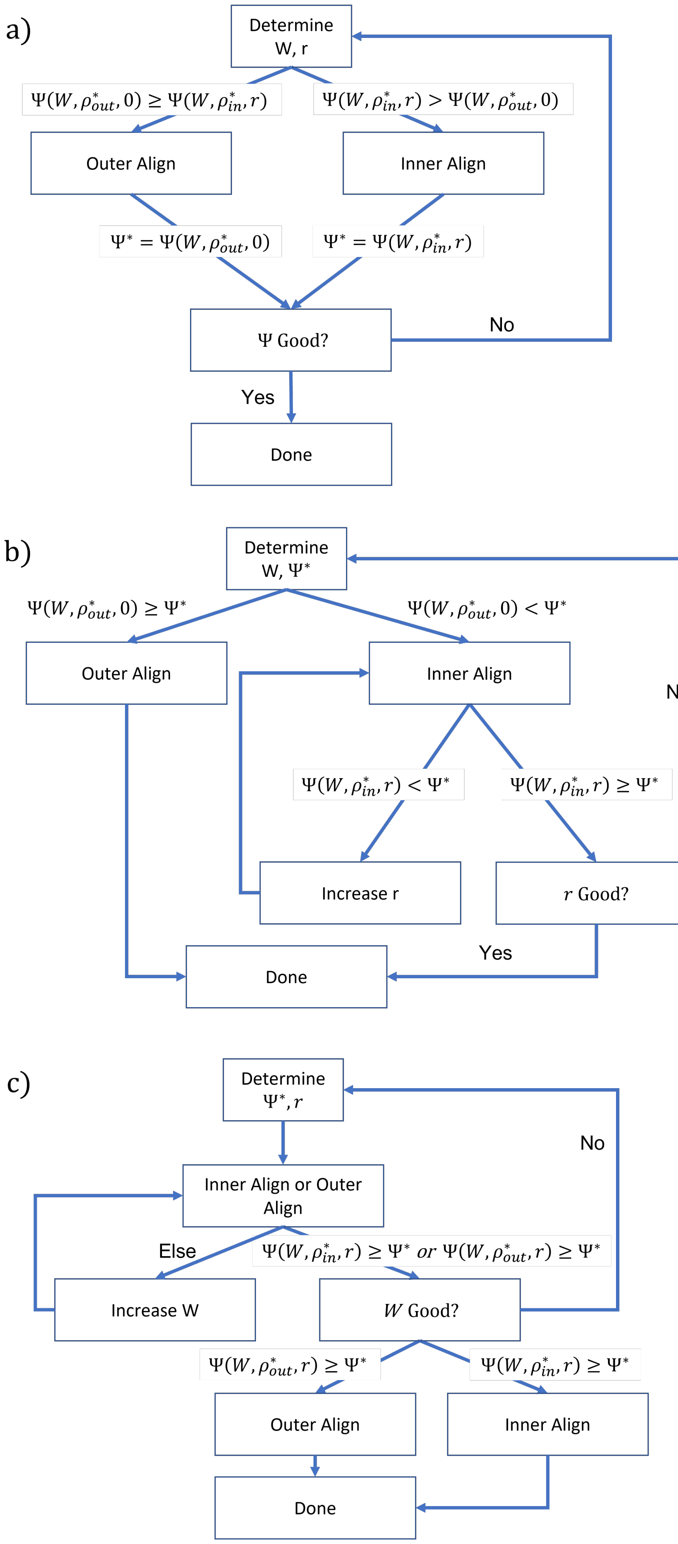


Figure 3.15: Design process derived for 3 separate desired design specifications using defined metric Ψ. The process is split based on starting parameters: a) W and r, b) W and Ψ^*, and c) Ψ^* and r.

Performance Comparison The 2-blade LIMMS latch demonstrated superior alignment capabilities compared to existing latching and alignment methods. While the scale and mechanical approach varies widely between examples, we can still compare alignment performance by designating the connector diameter D_c and connector thickness T_c, and characterize misalignment distance as a ratio of these geometries. The LIMMS prototype latch uses parameters $\rho = 35$ mm, $W = 40$ mm, $D_c = 80$ mm, and $T_c = 36$ mm from the base to the blade tips.

It is important to note that most existing latching methods use male and female hole patterns with identical diameters. For LIMMS, however, the connector diameter D_c does not connect into a hole pattern of the same diameter. The hole pattern diameter D_h is 140 mm, since the hole pattern on the cardboard does not have the same size constraints as the connector. This significantly increases the area of acceptance for successful latching. Additionally, for the following comparisons, we define displacement in the X or Y direction as parallel to the mating surfaces, while displacement in Z is the perpendicular distance between the mating surfaces.

Most existing literature does not include an analysis of tolerance to misalignment, but the few examples that do can only tolerate a very small amount of misalignment. For example, the HiGen connector, when misaligned in X or Y by $0.035D_c$, experienced a $\sim$65% drop in success rate [40]. It also has a tolerance range of $0.33T_c$ in Z with a $\sim$50% drop in success rate. Comparatively, our data and experiments show that the LIMMS prototype latch can be misaligned by up to $1.31D_c$ and $0.83T_c$ with a 100% success rate. This is a tolerance increase of 37x and 2.5x respectively.

For angular misalignments, the LIMMS latch's capabilities far exceed that of existing latches, with a 100% success rate at any angle about the Z-axis, even when considerably displaced in X-Y. This is thanks to the blade design, which features point contacts that can engage at any location in the slots. By comparison, the

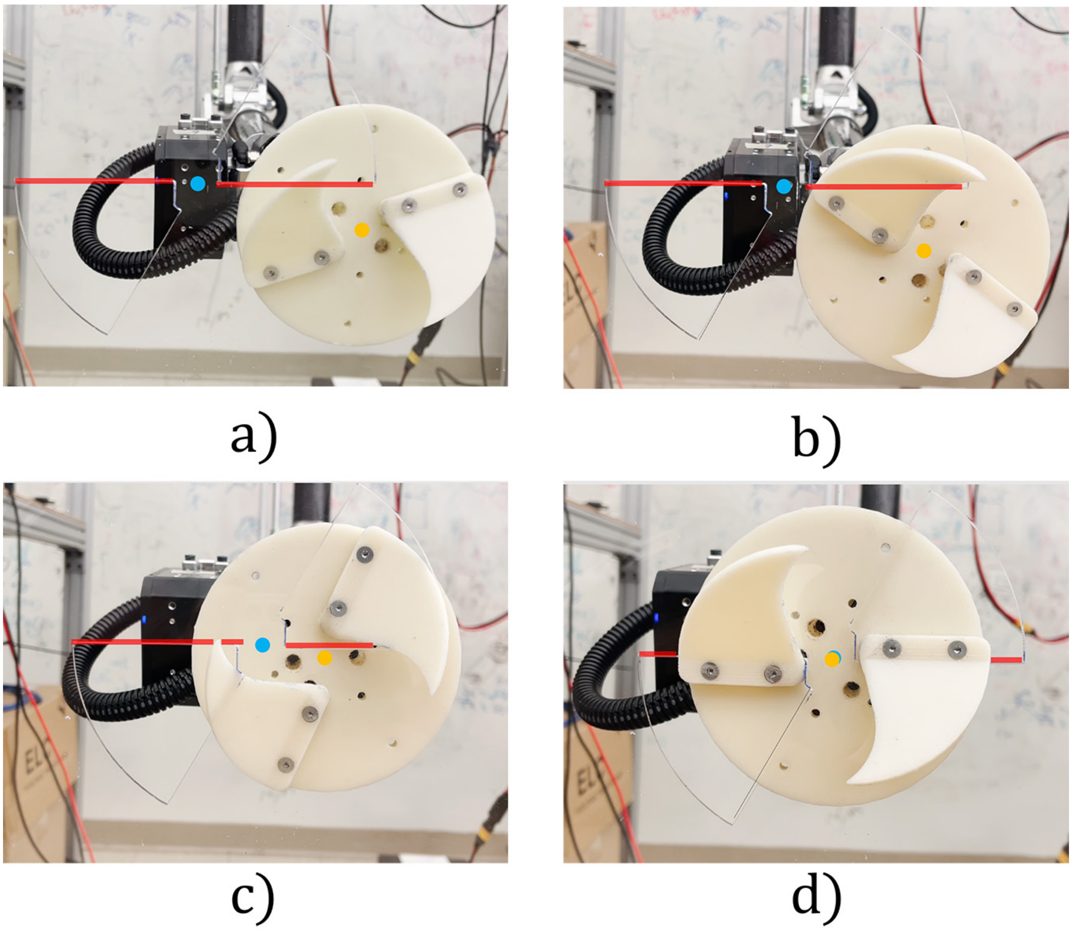


Figure 3.16: Two-blade latching with an acrylic mating surface. The blue point is the center of the hole pattern, and the orange point marks the center of the latch. The red lines depict the hole pattern edges that contact the blades.

HiGen connector experiences a ∼75% drop in success when displaced up to 12°. Other non-radial designs exhibit even lower tolerances, up to 1.7° [47].

The latch prototype used in these experiments does not represent the fully optimal case. These performance metrics can be enhanced even further by changing some aspects of the design. The area of acceptance in X-Y could be increased by setting $\rho = W$, which would allow the blade tips to engage even when displaced by a distance equal to the entire connector diameter. The prototype could reach a theoretical misalignment tolerance of $3\rho^*_{in}$ or $1.5D_c$ with these changes. While not the focus of this work, tolerance to Z-axis and angular misalignments can be improved by increasing the blade length or angle of attack.

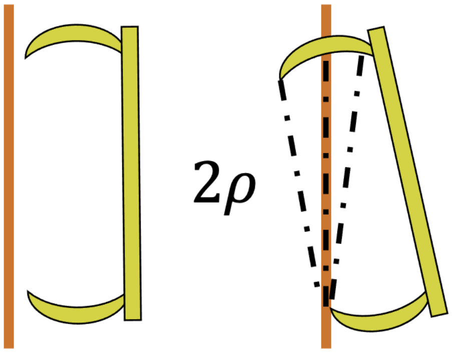


Figure 3.17: Brown represents the pattern, and the yellow a 2-blade latch. (*Left*) depicts the original constraint where both the latch and the pattern need to stay parallel. (*Right*) shows the concept of being able to have a nonparallel orientation but maintain the 2ρ distance needed to keep the optimal distance between blades.

3.4.1 Relaxing Assumptions

One of the largest limiting assumptions for the 2-blade case that was made for the analysis above was that the latch and pattern both had to be parallel with each other. This in practical terms is very limiting. To relax this assumption we introduce two weaker assumptions in its place. The first is that only either inner align edges make contact or outer align edges make contact during the process of latching, and all other contacts with respect to latching are negligible. Under certain use cases this assumption may hold true. Controllers can be adjusted accordingly as well to further ensure this. The other assumption is that the latch always can make progress into the pattern surface once both blades are engaged. This means that it can overcome any sort of friction forces or locking that could occur. As long as the two blades engage with their respective edges, the blades rotating motion will self-align and lock the mechanism in place.

To ensure that the analysis from before still holds, we design the edge of contact to have constant distance with the tip of the opposing blade. That constant distance will be 2ρ. Given an orientation offset where only at full engagement the base plate of the latch touches the pattern's surface, the first time both blades touch the surface of the latch pattern the distance will be 2ρ as shown by the

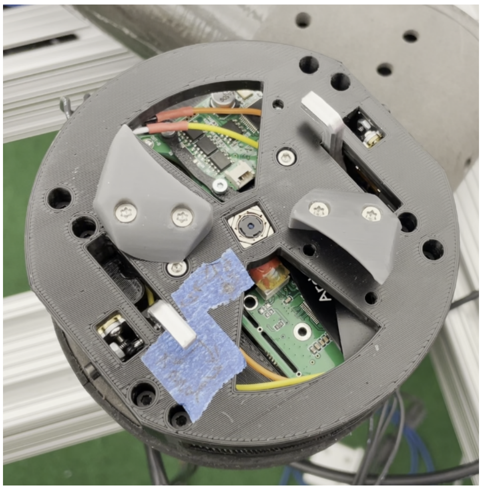


Figure 3.18: Hardware prototype of the latch with locking mechanism shown by the reversed blades machined out of aluminum. There is a camera at the center for perception. Each reverse blade is driven by a separate motor with a worm gear making it difficult for backdrivability.

dotted lines at the surface of the pattern in Fig. 3.17, maintaining the relationships derived in previous sections. By adding this last design feature, a mechanism is developed to latch and self-align quickly while allowing for large misalignment distances as well as nonparallel orientations. We were able to get the latch to slam into the pattern at high speeds and still fully engage when it was over a full latch pattern away. By doing this BEAR LIMMS prototype, as shown in Fig. 2.7b, could switch bases in approximately 2 seconds, whereas other similar systems can take up to the order of tens to hundreds of seconds.

3.4.2 Locking Mechanism

The last component missing from this latching mechanism is locking the mechanism in place once it is fully engaged with pattern. To do this we decided to use two blades that rotate upwards through the slots, shown in Fig. 3.18. The reason reverse blades were chosen were to be able to adjust slightly in any deformation of the cardboard box that may expand the hole in the pattern. It could continue to rotate further to further engage or press into the surface of the box. The blades are made of aluminum in our prototype and have small individual motors that actuate the blades. Also there are worm gears to make backdrivability very difficult. A camera was added to the center of the prototype for perception.

3.4.3 Conclusion

In this section we analyzed and showed that two blades were much better than one. Optimal parameters were then derived to get a relationship between the design parameters of the latch and pattern. This was then extended to nonparallel relationships between the latch and pattern surfaces by only considering either inner or outer align edges of contact. By holding a constant distance between the blade edges and the opposing tips, the previous optimal parameters hold. Once this was done then a hardware prototype was built with a locking mechanism and camera.

CHAPTER 4

Kineto-Static Duality

In this section covers a Jacobian method for finding trajectories where load on the end effector would be absorbed partially or primarily through the structure itself rather than through joint effort. This has a dual purpose of reducing the overall torque requirements for the system and decreasing the amount of energy consumed over time. This method is formalized as *kineto-static duality* where deriving LIMMS statics relationship with differential kinematics gives way to the following relationship [51, 52]:

$$\tau = J^T(q)\gamma_e, \tag{4.1}$$

Where J is the Jacobian of LIMMS give a pose q, and γ_e is the end effector forces and wrenches. The Jacobian connects the relationship between the motor torque outputs τ and γ_e. Though quite simple in form, Eq. (4.1) will allow LIMMS for certain trajectories to carry loads much greater than what one would imagine given LIMMS motor specification. As far as the contributors are aware this section's contributions can be summarized as follows:

1. Analyzed LIMMS kineto-static duality,
2. Discovered LIMMS unique areas of structurally load-bearing poses, and
3. Experimentation where BEAR LIMMS, designed for speed and not strength, moves approximately 10 times its own weight.

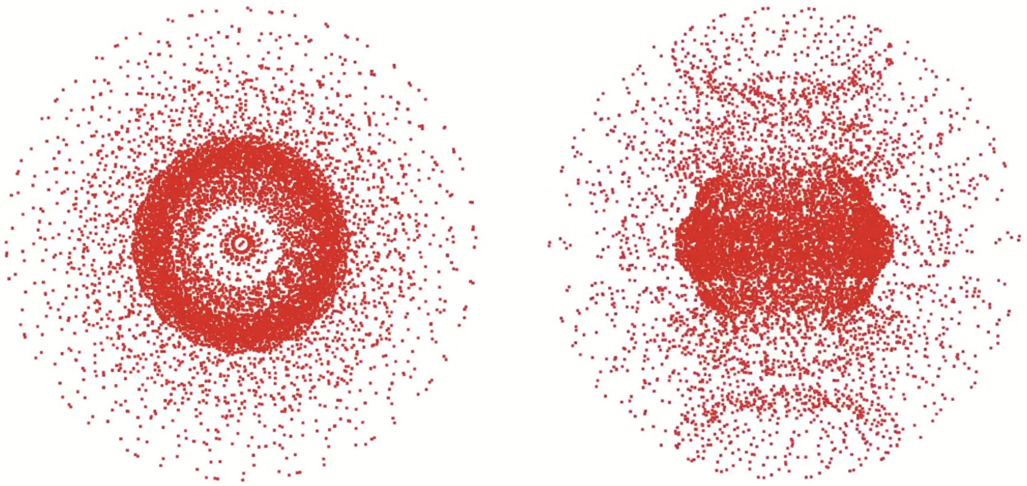


Figure 4.1: Two different views of small singular values close to zero of the Jacobian of LIMMS. Each red dot represents a sampled point in the workspace of LIMMS. The Jacobian was then calculated, and then its singular values. The space that it covers reaches the most outer and inner parts of the workspace of LIMMS.

4.1 Jacobian

Eq. (4.1) uses J^T to give way to a relationship between τ and γ_e. LIMMS is a 6-DOF system. As long at the motor axes are not sequentially parallel, it should for the most part avoid singularities in J. Henceforth, J^T would rarely hit a singularity. Singularities in the Jacobian imply a lost in DOFs. A single motor or combination of them result in a redundancy. Usually, this happens somewhere over or under the base axis and directly aligned with that axis. LIMMS does indeed follow this pattern. It has 3 over the base and 3 under the base as seen in Fig. 4.3. What is particularly peculiar are the curves near those points. On top of that the area near these regions are poorly conditioned meaning that the Jacobian is close to a singularity or have very small eigenvalues or singular values.

Fig. 4.1 depicts a sampling of small singular values close to zero of the Jacobian of LIMMS. These small singular values, not only have some sort of structure or pattern clearly seen by the rings in 3D space, they reach almost the entire workspace of LIMMS. They are abundant and dense in certain regions. By observing this structure, we can leverage the kineto-static duality.

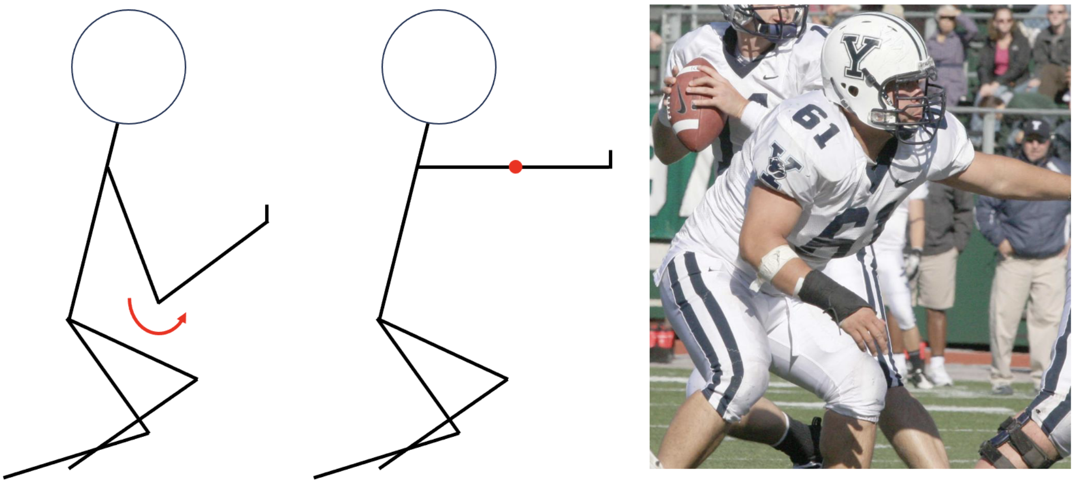


Figure 4.2: Depicts an analogy of the kineto-static duality using American football. Comparing the stick figure on the left and in the center, the only difference is one has its elbow bent and the other has it straight. In American football, depicted on the right an offensive lineman has to protect the person with his hands. The stick figure with his elbows locked out can hold larger forces or players for longer periods of time because the muscles in the arm do not have to hold the weight of the player.

Even being close to small eigenvalues or singular values can be useful. The nullspace of J^T in Eq. (4.1) implies that the load is being absorbed by the structure. loads purely along the nullspace of the Jacobian would imply that the robot can carry the load as long as its structure does not fail. For most systems the structure tends to be much stronger than the actuators. If nullspaces are rare amongst robotic poses, poses with small singular values can be used instead. Loads in the direction of small singular values or eigenvalues would indicate there is load sharing between the actuators and structure. Depending on how small those values are, the actuators may feel effectively a negligible load.

Intuition An analogy of what is being proposed is shown in Fig. 4.2 using American football. The picture on the right shows the author playing the position of offensive lineman. The job of this position is to keep the opposing players from hitting the player with the ball, so they use their hands to keep opposing players from getting to the players with the ball. The stick figure on the right and center depict two poses by an offensive lineman. The left most one has its elbow bent. The center has its elbow straightened out. The left figure has to use its arm muscles more to effectively maintain the opposing player. The player in the center though, if already in contact with a player, can sustain much larger forces for longer lengths of time since most of the force is generated from much larger muscles in the hips and legs. The muscles in the arms do not have to work as much.

Locked elbows and straight arms would correspond to a singularity in the Jacobian and a loss of DOF. On the other hand the figure on the left would have a corresponding singular value not close to zero. This section proposes finding trajectories or heuristics to generate trajectories capable of lifting much heavier weight than otherwise capable, reduce torque requirements, and save energy. Specifically, during the LIMMS hardware implementation this helped reduce the load felt by the motors allowing for increased mobility.

Data Collection Methodology To achieve this end, we sampled a radial slice of the workspace of LIMMS give a particular end effector orientation as seen in Fig. 4.3. To evenly sample, the workspace was used first, and then using its IK we can translate it into joint space. As discussed in Chapter 2, numerical IK had to be used. Due to this at some points it converged while in others it did not. There are even very interesting clusters where it seemed to converge quite well. At each of these points we calculated the corresponding Jacobian. This then was multiplied by a unit vector pointing in the direction of gravity. From their the maximum of all the torques was used to color the point with red corresponding with small max torque values and blue with large ones. The base of LIMMS is to the center left. If the figure and its mirror counterpart were combined, LIMMS would sit at the center.

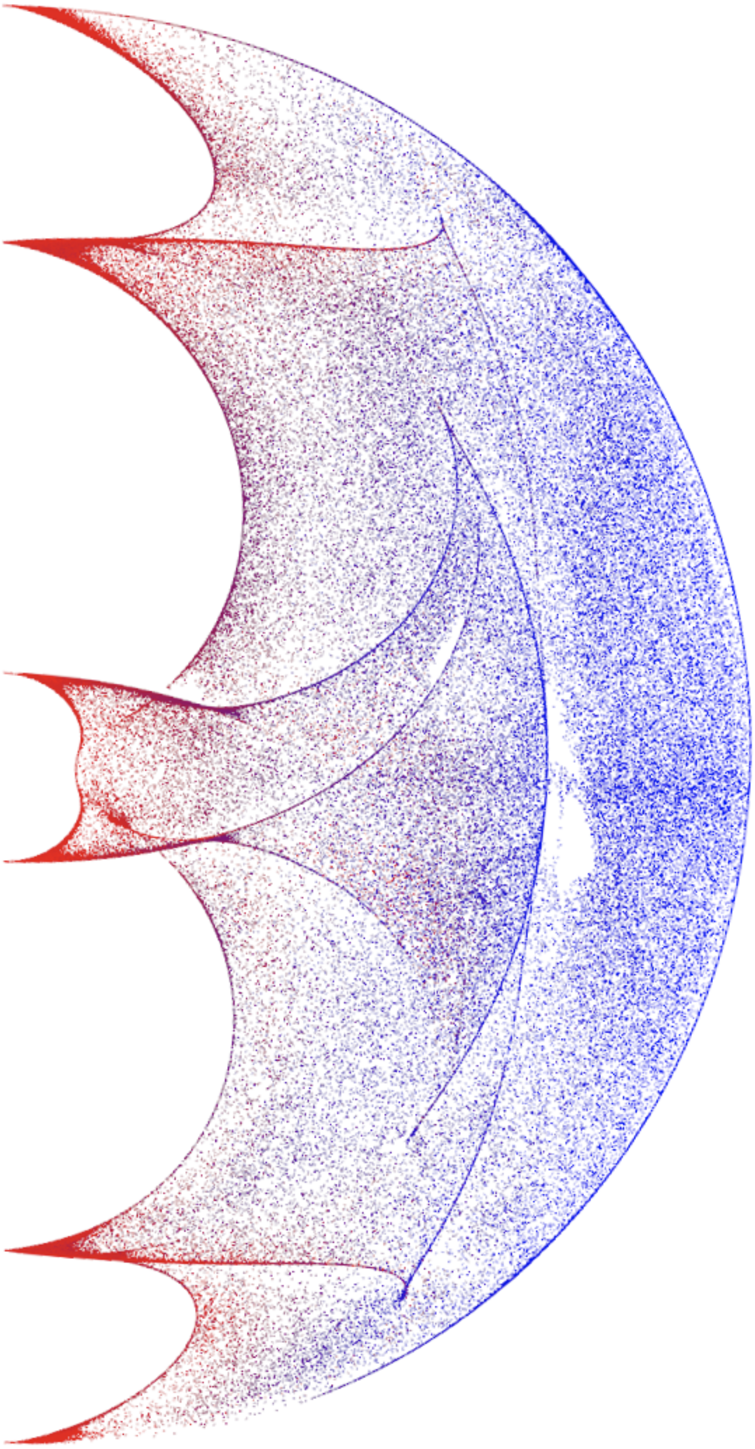


Figure 4.3: Depicts a radial slice of LIMMS workspace showing the max torque required by LIMMS to sustain a unit vector of force in the direction of gravity at the end effector for certain poses. LIMMS base would be left center and any colored point would correspond to a point LIMMS could reach. The color red indicates a small max torque value, while blue indicates a large max torque values. These values were calculated using Eq. (4.1).

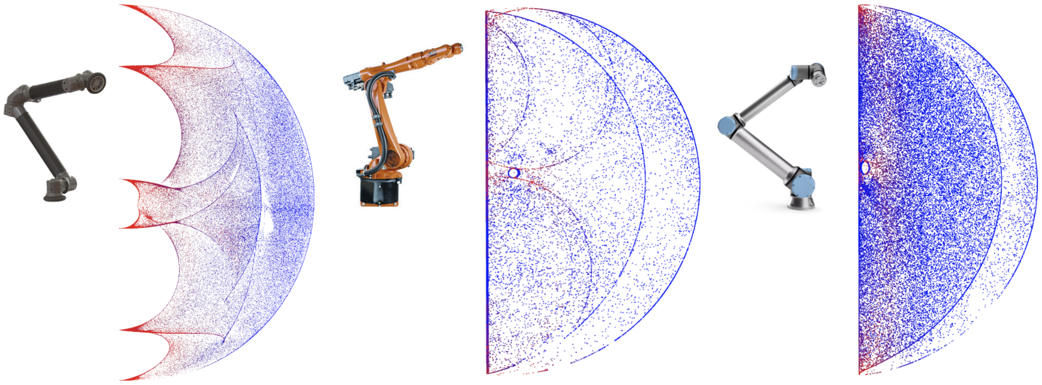


Figure 4.4: Three robots and the cloud of max torques calculated similarly to Fig. 4.3 side-by-side for comparison. The robots and cloud sizes are not to scale but have been enlarged to have roughly the same semi-circle of radial slice of sampling. The left is LIMMS. The center is Kuka KR5 Arc, and the right is UR10e. Note how much more red there is for LIMMS. However, our techniques can still be applied here.

Observations There are interesting features of Fig. 4.3. First, note that the closer to the center, the redder the samples are, and the further out the bluer. This is not surprising, but the amount of red is quite astounding compared to some of its peers in Fig. 4.4. Additionally, there are several nullspace points indicated by the sharp edges. Around those regions there are also curved manifolds. The edges of these manifolds can be considered as physical constraints placed its kinematics. In other words, the physical structure keeps this constraint from being violated. This implies that the nullspace of the Jacobian is perpendicular to the hyperplane tangent to the boundary. As the points approach the sharp edges, the redder it gets. The surfaces also get more horizontal, meaning the nullspace is more vertical or aligned with the direction of gravity. The contour of the inner edges are much more curved than some of its peers in Fig. 4.4. This curved manifold allows for much more variation in the direction of nullspace at the edges as opposed to a straight vertical wall. This would allow for different use cases of this technique to handle loads or forces in a wide range of directions. Finally, there are long red lines that extend from the second sharp point from the top and bottom points. This implies that along that shelf line moving inner or out could be done with much less torque.

We hypothesize these characteristics have to do with the 45 degree opposing joint elbow in the LIMMS design, as most loads on the end effector do not cause a torque purely in the plane of rotation causing some of the forces to be absorbed into the structure. As a side note, on the hardware implementation, due to the massive loads, the 3D printed part link structures would flex. The direction of flex usually is in the direction that reduces the torque output for the joints. This needs more research to say any definitive statement though.

This method differs from the robot equation of motions in the sense that it is not balancing the weight of the load with nonlinear terms. If a sampling method was done for the robot equation of motion, it would have to be resamples for every different load. Whereas, the Jacobian has a linear relationship meaning the analysis still holds for any vector in the same direction. For most robots the largest force motor torque has to deal with is gravity, which is why we chose it for our analyses. Secondly, balancing points can be very thin and difficult to track. Whereas, for LIMMS there is a large region in which this method can be done, and Fig. 4.4 implies that this can be a much more general method applicable to many systems.

4.1.1 Results

Finally, we tested this using BEAR LIMMS from Chapter 2. This prototype is light weighing ~ 2.5 kg and is not built for high torque tasks. For this test we tried different poses and weights at the end effector as seen in Fig. 4.5. In Fig. 4.5 (*left*) it was given a pose where the small singular value of the Jacobian was not aligned with gravity. It could not hold the weight, 7.5 lbs. Therefore, we had to catch it. In Fig. 4.5 (*right*), we picked a pose where the small singular value was aligned with gravity. It held ~ 75 lbs. It even moved the weight with its bottom motor. This show how powerful this method could be if applied properly. Eventually, at this weight the 3D printed structures cracked as expected giving

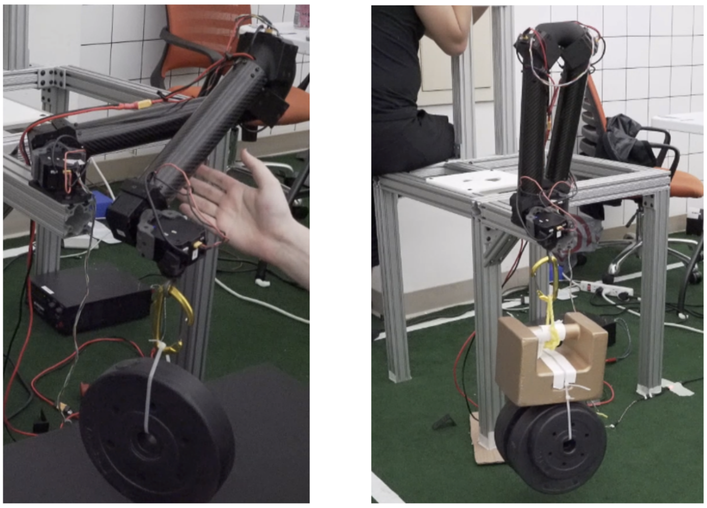


Figure 4.5: Shows the difference between poses where the small singular values are in the direction of gravity. The BEAR LIMMS prototype was used for this. It is about ~ 2.5 kg and is meant for high speed not torque output. (*Left*) shows a pose where the Jacobian's small singular value is not aligned with the direction of gravity. It fails and falls only holding 7.5 lbs. (*Right*) is a snapshot of BEAR LIMMS rotating its bottom base to move ~ 75 lbs. This experiment highlights how much of a difference this technique can make.

how much load was on the structure.

4.1.2 Conclusion

In this section we used kineto-static duality to develop a method for generating trajectories that bore most of the load on the structure as opposed to the motors. Our analysis showed that for LIMMS regions near the vertical center line could bear more weight. There were also curved manifolds in which the Jacobian nullspace varied allowing for a much wider arrangement of nullspaces. As opposed to the typical nullspace point seen in robots, LIMMS has a nullspace manifold that changes in direction along those curves. We then used this to generate a trajectory where we had BEAR LIMMS hold and move ~ 75 lbs, which is astounding for high speed low torque actuators.

CHAPTER 5

Planning

In this chapter a motion planner for LIMMS is presented for multi-agent, multi-modal package delivery. Coordinating amongst multiple LIMMS, when each one can take on vastly different roles, can quickly become complex. For such a planning problem we first compose the necessary logic and constraints. The formulation is then solved for skill exploration and can be implemented on hardware after refinement. To solve this optimization problem we use alternating direction method of multipliers (ADMM). The proposed planner is experimented under various scenarios which shows the capability of LIMMS to enter into different modes or combinations of them to achieve their goal of moving shipping boxes. Practically, this planner would be done offline to gather solutions depending on the scenario. It can then be used by a function approximator to predict solutions or even be used as warm start to solve the actual problem online via Bender's decomposition [53].

Self-reconfigurable robots, such as LIMMS, present challenging mechanical design, motion planning, and control problems. As the number of robots scales up, issues arise in communication, coordination, and decision making partially because it can no longer be assumed to have accurate global information. Modular self-reconfigurable robot systems have attracted the attention of researchers [18,54,55]. The ability to transform relatively simple robot modules to realize various forms and functions can be applied in areas such as aerial or space robotics [14,15] and more recently last-mile delivery [56].

LIMMS introduces a challenging motion planning problem with each LIMMS having multiple DOFs. This requires kinematic constraints which were rarely done

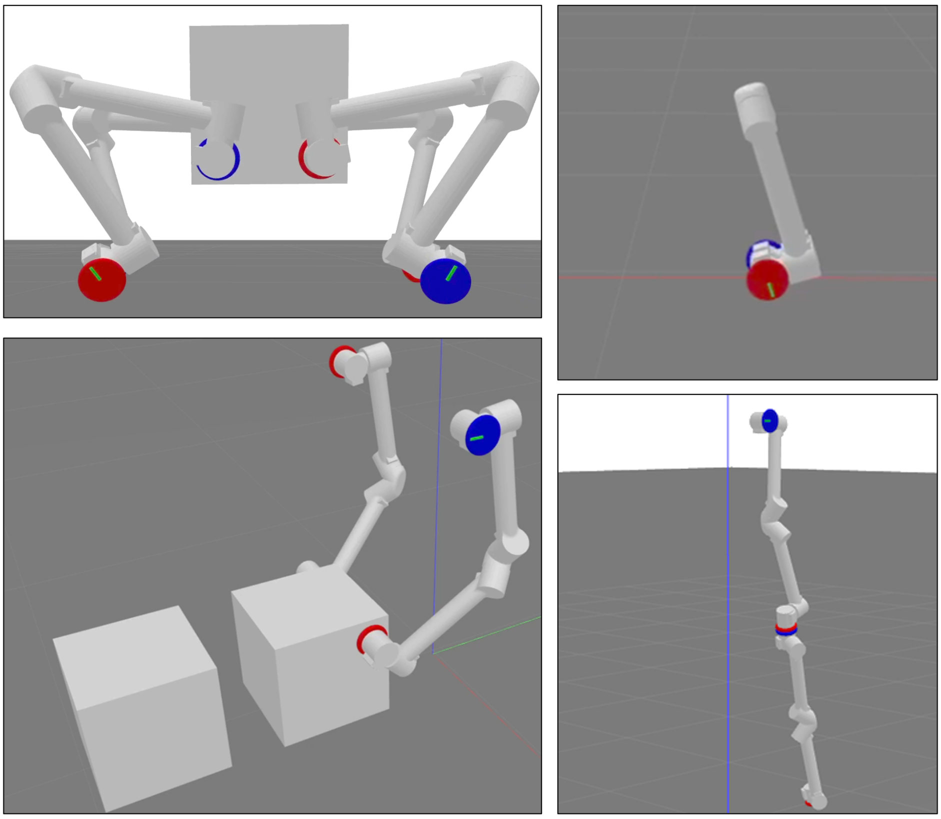


Figure 5.1: Shows the various operational modes for LIMMS which will be considered for the optimization formulation. *Upper Left*: four LIMMS attached to a box in quadruped mode. *Upper Right*: a single LIMMS self-balancing to move. This is called free mode. *Lower Left*: two LIMMS anchored to surfaces in manipulator mode. *Lower Right*: LIMMS attached to each other. This can be used in both quadruped and manipulator mode.

in previous works [57, 58]. Moreover, each LIMMS operate in different modes: a wheeled robot, a leg, or an arm. The delivery package can also be transported in various ways: manipulation or locomotion via acting as the body for a quadruped robot. This introduces complicated mixing of discrete decisions and continuous constraints.

Various approaches have been explored to resolve the reconfiguration planning problem, such as graph search methods [57], reinforcement learning [59], or optimization-based approaches [60]. We adopt optimization-based methods for LIMMS. Among the optimization-based methods, mixed-integer programs (MIP) are useful for discrete decision-making within multi-agent systems [61, 62]. However, the LIMMS motion planning problem includes nonlinear constraints such as kinematics. Therefore, the problem becomes a mixed-integer nonlinear (nonconvex) program (MINLP). MINLPs are known to be computationally difficult when the problem scale is large. Aside from directly applying commercialized solvers, e.g., SCIP, there are generally two approaches that transform the MINLP problems into MIPs using linear approximations of the nonlinear constraints, or into nonlinear programs (NLPs) using complementary constraints to replace binary variables. Unfortunately, as the problem scales up, nonlinear constraints yield a large amount of piecewise linear approximations making the problem very slow [63], and complementary constraints tend to cause infeasibility without a good initial guess [64].

In this section, we implemented the alternating direction method of multipliers (ADMM) to solve the MINLP problem, inspired by [65, 66]. ADMM decomposes the problem into two sub-problems: an MIP problem and an NLP problem. By decomposition, the problem scale of both MIP and NLP is reduced and becomes more tractable. The logic constraints about the connections of LIMMS are formulated into MIP while the nonlinear kinematics constraints are formulated into NLP. We establish general rules for such a system to operate under and use ADMM to explore possible approaches to resolve the given scenario, showing the feasibility of our proposed method as well as the richness of the system. To summarize, this sections contributions are as follows:

1. Solved a unique type of problem that incorporates multi-agent locomotion and manipulation,
2. Proposed a formulation that can be generalized to solve such types of multi-agent motion planning problems, and
3. Solved the proposed formulation with ADMM and demonstrated it on hardware.

This section was taken from [24]. The author's main contribution on this work was in forming logical constraints needed for the optimization formulation and built the environment to showcase our results. First two authors ran hardware experiments and wrote [24] collaboratively.

5.1 System Description

One of the crucial aspects of LIMMS is its latching mechanism, which allows it to attach either end to anchor points or itself, and it relaxes some of the strictness presented in the optimization formulation. Latching will be used frequently since it is how LIMMS transitions to different modes and grab objects for manipulation. The current prototype proposed in [22] consists of a radially symmetric multi-blade design. Fig. 5.1 (bottom right) depicts the latch prototype. By the geometry of the blade and the anchor point hole pattern, the latch mechanically self-aligns when it rotates. The mechanism is designed to maximize robustness to misalignment in position and orientation about its center axis, see [22]. This has a dual effect of easing control effort and allowing for the latches to pull the box or itself into the desired position and orientation without being fully positioned. If LIMMS is kinematically constrained and cannot fully reach the target position and orientation, there is slack created by the latch's mechanical design. In this sense when latching to an anchor point, the alignment does not need to strictly satisfy its constraint.

Modes of Operation Through latching, LIMMS can enter many different modes to complete its tasks. For our proposed planner we only consider three different modes as these will be sufficient for delivering packages. However, LIMMS has the potential for many other modes, some of which are described in [56]. Fig. 5.1 depicts three modes considered in our optimization formulation: four LIMMS attached to a box are in quadruped mode, LIMMS attached to walls or surfaces are in manipulator mode, a single LIMMS which wheels around or move like a snake is in free mode. The last sub-figure in the lower right shows two LIMMS attached to the end of each other. This can be viewed as a sub-skill accessible to all modes.

5.2 Problem Formulation

In this section, we demonstrate the optimization formulation for LIMMS motion planning. The objective function of the optimization problem is to minimize the distance from the box center position to the target box position, i.e. deliver the box to the goal. The constraints consist of two parts: logic, which is formulated into constraints using integer variables, and kinematics or dynamics, which is formulated into linear or nonlinear equations of motion. As a result, the problem is an MINLP, which will be separated into an MIP and NLP to be then solved with ADMM. Assume there are B boxes and L LIMMS. Each box has $S_B = 4$ anchor points at the center of each face. The optimization is run from $t = 1, ..., T$ time steps. We use upper case letters to indicate constants such as the number of boxes B, the number of anchor points S_B, and use the lower case letters to index the quantities such as $b = 1, ..., B$, $s_b = 1, ..., S_B$. i is used to index the binary variables. Upper case letters are also used as variable names. For example, $z_{B,i}$ are binary variables associated with boxes. We state the assumptions made for this formulation:

1. The box does not rotate and the momentum is assumed to be balanced. This simplifies the dynamics constraints. In practice, this minimizes the damage to the contents in boxes during shipping.
2. Multi-body dynamics of LIMMS are not enforced. This is to simplify the constraints.

All binary and continuous variables are summarized in Table 5.1 and Table 5.2,

	Var	Dim	Description
Binary	$z_{B,i}$	$[B, T]$	Mode for box b at time t. $i = \{1 : \text{stable object}, 2 : \text{free object}, 3 : \text{manipulated}, 4 : \text{quadruped}\}$
	$z_{L,i}$	$[L, T]$	Mode for limb l at time t. $i = \{1 : \text{free}, 2 : \text{arm}, 3 : \text{add arm } 4 : \text{leg}, 5 : \text{add leg}\}$
	$z_{S,i}$	$[B, S_B, L, T]$	Mode of connection for anchor point s_b on box b to limb l at time t $i = \{1 : \text{empty}, 2 : \text{to arm}, 3 : \text{to leg}\}$
	$z_{W,i}$	$[S_w, L, T]$	Mode of connection for anchor point s_w on wall to limb l at time t $i = \{1 : \text{empty}, 2 : \text{to arm}\}$
	$z_{Ac,i}$	$[L_{pr}, L_{po}, T]$	Mode for connection s.t. limb l_{po} connects as an additional limb to l_{pr} $i = \{1 : \text{not connected}, 2 : \text{connected}\}$
	$z_{Lc,i}$	$[L_{pr}, L_{po}, T]$	Mode for connection s.t. limb l_{po} connects as an additional leg to l_{pr} $i = \{1 : \text{not connected}, 2 : \text{connected}\}$
	δ_{La} δ_{Lg}	$[J, L, P, T]$ $[J, L, P, T]$	Joint j of limb l stays within convex region p at time t
	δ_{Ba}	$[B, C, P, T]$	Corner c of box b stays within convex region p at time t
	δ_{Bg}	$[B, P, T]$	Box b stays within ground region p at t

Table 5.1: Table of binary optimization variables

respectively. Binary variables $\delta_{Ba,i}$, $\delta_{Bg,i}$, $\delta_{La,i}$, $\delta_{Lg,i}$ pertain to enforcing collision avoidance with the environment, and the corresponding $\lambda \in [0, 1]$ variables are for convex combinations. Note *pr* is short for previous, and *po* is short for post.

5.2.1 Integral Logic Constraints

Mode for Boxes We define 4 modes for each box represented by 4 binary variables: $z_{Bi}[b, t], i = 1, ..., 4$ for the mode of box b at t. Mode 1 is stable object mode, where the box is supported by the ground. Mode 2 is free object mode

	Var	Dim	Description
Continuous	$\mathbf{p}_B$	$[B, T]$	Position of center of box b at time t
	$\mathbf{R}_B$	$[B, T]$	Orientation of box b
	$\mathbf{c}_B$	$[B, C, T]$	Position of corner c for box b at time t
	$\mathbf{p}_L$	$[J, L, T]$	Position of joint j of limb l at time t
	$\mathbf{R}_L$	$[J, L, T]$	Rotation matrix of joint j of limb l
	$\mathbf{f}_L$	$[B, S, L, T]$	Contact force at anchor point s of box b from limb l at time t
	λ_{La} λ_{Lg}	$[B, C, V, P, T]$ $[B, C, V, P, T]$	Convex combination variable for corner c of box b by combining vertex v from patch p
	λ_{Ba} λ_{Bg}	$[B, C, V, P, T]$ $[B, V, P, T]$	Convex combination variable for corner c of box b (or box b itself for ground) by combining vertex v from patch p
	$\mathbf{a}$	$[L_1, L_2, T]$	Normal vector of separating plane for l_1 and l_2
	b	$[L_1, L_2, T]$	Offset of separating plane for l_1 and l_2

Table 5.2: Table of continuous optimization variables

where the box is in the air subject to gravity. Mode 3 is manipulated object mode where the box is connected to a manipulator. Mode 4 is quadruped mode where the box is used as a robot body. We currently only allow quadruped robot for walking. This can be relaxed to incorporate more solutions such as simultaneously bipedal walking while manipulating boxes as in [12]. At each t, a box is subject to 1 mode, such that: $\sum_{i=1}^{4} z_{Bi}[b, t] = 1 \ \forall b, \ \forall t$.

Mode for LIMMS We define 5 modes for each LIMMS represented by 5 binary variables: $z_{Li}[l, t]$, where $i = 1, ..., 5$ indicating the mode of LIMMS l at t. Mode 1 is free (wheeled) mode, where the corresponding LIMMS unit moves on the ground like a Segway robot. Mode 2 is manipulation mode, where LIMMS connects to one connection site on the wall and may connect to one box to manipulate it. Mode 3 is add arm mode, where a LIMMS can connect to another LIMMS to extend the length of the arm for a larger workspace. Mode 4 is leg mode, where LIMMS connects to a box and serves as a leg. Mode 5 is add leg mode, where it can connect to another LIMMS to extend the length of the leg similar to mode 3. At any time step, LIMMS can only be in one mode, such that: $\sum_{i=1}^{5} z_{Li}[l,t] = 1\ \forall l, \forall t$.

Mode for Anchor Points Each box has 4 anchor points on each side face. We define 3 modes for each anchor point by 3 binary variables $z_{Si}[b, s_b, l, t]$, where $i = 1, ..., 3$, denoting the connection mode for anchor point s_b on b to l at t. Mode 1 is empty mode, where s_b on b is not connected to l. Mode 2 is arm mode where l connects to s_b as an arm. Mode 3 is leg mode where l connects to s_b as a leg. Their summation has to be 1 at each time step: $\sum_{i=1}^{3} z_{Si}[b, s_b, l, t] = 1\ \forall b,\ \forall s_b,\ \forall l,\ \forall t$. In addition, at each time s_b can connect to no more than 1 LIMMS, while a given LIMMS can connect to no more than 1 s_b at its base point. This introduces two more constraints: $\sum_b \sum_{s_b} z_{Si}[b, s_b, l, t] \leq 1\ \forall l,\ \forall t$ and $\sum_l z_{Si}[b, s_b, l, t] \leq 1\ \forall b,\ \forall s_b,\ \forall t$. s_b on b is associated with a physical connection. If latching is enforced, the position and orientation of the base of LIMMS is constrained:

$$\begin{aligned} \mathbf{p}_L[j=0, l, t] &= \mathbf{p}_B[b, t] + \mathbf{R}_B[b, t]\mathbf{o}[s_b] \\ \mathbf{R}_L[j=0, l, t] &= \mathbf{R}_o[s]\mathbf{R}_B[b, t] \end{aligned} \tag{5.1}$$

Where $\mathbf{o}[s]$ is the constant offset vector from the center of the box b to the anchor point s_b. $\mathbf{R}_o[s_b]$ is the constant rotation matrix from the box frame located at the geometric center of the box to the s_b frame located at anchor point s_b. This conditional equality constraint can be enforced through big-M formulation such that if $z_{S,i} = 1$, Eq. (5.1) is enforced.

Each anchor site on the wall s_w has two modes. Mode 1 is empty mode where s_w is empty, and mode 2 is manipulation mode where l connects to s_w as an arm. Multiple s_w can exist on the ground. Two binary variables $z_{Wi}[s_w, l, t]$, where $i = 1, 2$ are used to represent those modes. The associated mode constraints and physical connection constraints are similar to the anchor points on the boxes. We

Table 5.3: Table of logic rules

	#	Logic Rule Description	Mathematical Formulation
Quadruped	1	Box b in quadruped mode, all 4 s_b is connected to leg mode LIMMS.	$z_{B,4}[b,t] = 1 \implies \sum_l z_{S,3}[b, s_b, l, t] = 1 \ \forall s_b$
	2	Anchor point s_b on box b in leg mode, box b is in quadruped mode.	$z_{S,3}[b, s_b, l, t] = 1 \ \exists l, \ \exists S_B \implies z_{B,4}[b,t] = 1$
	3	Anchor point s_b is connected to LIMMS l as a leg, l is in leg mode.	$z_{S,3}[b, s_b, l, t] = 1 \implies z_{L,4}[l,t] = 1$
	4	LIMMS l in leg mode, it's connected as a leg to one anchor point.	$z_{L,4}[l,t] = 1 \implies \sum_s \sum_b z_{S,3}[b, s_b, l, t] = 1$
Manipulation	5	b in manipulated mode, at least 1 s_b is connected to 1 arm mode l.	$z_{B,3}[b,t] = 1 \implies \sum_l \sum_{s_b} z_{S,2}[b, s_b, l, t] \geq 1$
	6	s_b is connected to l in arm mode, l is in arm or add arm mode.	$z_{S,2}[b, s_b, l, t] = 1 \implies z_{L,2}[l,t] = 1$ or $z_{L,3}[l,t] = 1$
	7	l is in arm mode, l is connected to one s_w on the wall or ground.	$z_{L,2}[l,t] = 1 \implies \sum_{s_w} z_{W,2}[s_w, l, t] = 1$
	8	s_w on wall or ground, s_w is connected to l, l is in arm mode.	$z_{W,2}[s_w, l, t] = 1 \implies z_{L,2}[1,t] = 1$
	9	LIMMS l is in add arm mode, it's connected to one other LIMMS.	$z_{L,3}[l,t] = 1 \implies \sum_{l_{pr}} z_{Ac}[l_{pr}, l, t] = 1$
	10	LIMMS l_{pr} is connected to l_{po}, l_{pr} is in arm or add arm mode, l_{po} is in add arm mode.	$z_{Ac}[l_{pr}, l_{po}, t] = 1 \implies \begin{cases} z_{L,2}[l_{pr}, t] = 1 \\ z_{L,3}[l_{po}, t] = 1 \end{cases}$
	11	LIMMS l_{pr} is connected to l_{po}, l_{pr} cannot connect to any box.	$z_{Ac}[l_{pr}, l_{po}, t] = 1 \implies z_{S,2}[b, s_b, l_{pr}, t] = 0$
Free	12	Box b is in stable or free object mode, all anchor points s_b are empty.	$z_{B,1}[b,t] = 1 \ or \ z_{B,2}[b,t] = 1$ $\implies z_{S,1}[b, s_b, l, t] = 1 \ \forall s_b \ \forall l$

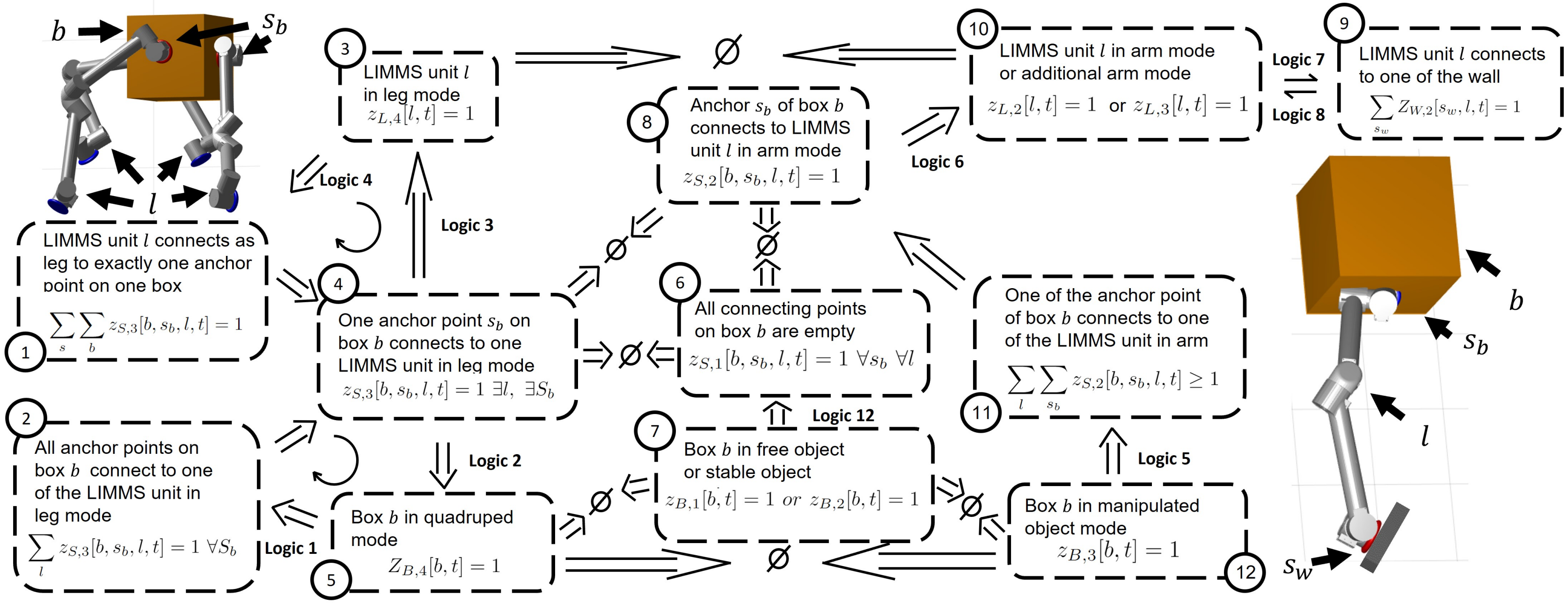


Figure 5.2: Implications of logic 1-12. Left half is associated with robot mode while right half is associated with manipulation mode. Arrows are labeled with specific logic rules. If not, the implication is mathematically correct. The symbol $\Rightarrow \emptyset \Leftarrow$ means mutually exclusive.

also define binary variables for additional connections between LIMMS as arms or legs: $z_{Ac,i}[l_{pr}, l_{po}, t]$ or $z_{Lc,i}[l_{pr}, l_{po}, t]$, $i = 1, 2$. If $z_{Ac,i}[l_{pr}, l_{po}, t] = 1$, LIMMS l_{po} connects as an additional arm to LIMMS l_{pr} at t. Similar for $z_{Lc,i}$.

Logic for Boxes as Robot Bodies We define 4 logic rules for any box detailed in rule $1-4$ in Table 5.3. They constrain the boxes in quadruped mode and LIMMS units connected to it. In short it is to ensure that several things happen simultaneously: 1) b is used as the robot body, 2) four LIMMS are its legs, and 3) connections happen between them. On the other hand, if no box is used as a quadruped body, no LIMMS should be used as legs. This is enforced through formulating the rules into a loop as shown in Fig. 5.2. If a single LIMMS is used as leg, it should connect to 1 anchor point on 1 of the boxes due to Logic 4, and the corresponding box should be in quadruped mode due to Logic 2.

Each logic can be formulated as constraints between integer variables through big-M formulation. For example, Logic 1 can be written as $1 - M(1 - z_{B,4}[b, t]) \leq \sum_l z_{S,3}[b, s_b, l, t] \leq 1 + M(1 - z_{B,4}[b, t])$ where M is a large constant (usually 10^5). Other constraints follow similarly.

Logic for Boxes as Manipulated Objects We define 7 logic rules for boxes as objects and are manipulated by LIMMS as arms. They again constrain the modes for the box and LIMMS connected to it. From a high-level perspective, this is to ensure that the following happens simultaneously: 1) box is being manipulated by 1 or more LIMMS, 2) 1 or more LIMMS operate in arm mode, 3) connections occur between the arm and box, and 4) arm is connected to 1 anchor point on the wall or ground. While most of the logic rules are single directional, the bidirectional rules are enforced by formulating implicit loops. For example, there is no explicit rules enforcing a box to be in manipulated object mode (arrow from 8 to 12) if LIMMS arm is manipulating it (block 8 in Fig. 5.2). However, block 8 is mutually exclusive with block 4 and 6. This means that if 8 happens, 4 and 6 cannot happen, which means block 5 and 7 cannot happen (by reversing the direction of implication). This in turn indicates that 12, which is the complement of 5 and 7, has to happen. Note that we did not include the constraints for additional arm or leg mode in Fig. 5.2, but similar arguments carry over.

Logic for Boxes as Stable or Free Objects Logic 12 in Table 5.3 is defined for boxes as stable or free object modes which states that if a box is stable or free, all its anchors are empty.

5.2.2 Continuous Constraints

Kinematics Kinematics constraints are imposed for each LIMMS through a series of linear constraints and bilinear constraints in the same fashion as [67]:

$$\begin{aligned}
&\mathbf{p}[j+1,l,t] = \mathbf{p}_L[j,l,t] + \mathbf{R}_L[j,l,t]\mathbf{p}_{j+1,j} \\
&\mathbf{R}_L[j-1,l,t]\mathbf{z}_{j-1,j} = \mathbf{R}_L[j,l,t]\mathbf{z}_{j,j} \\
&\mathbf{R}_L[j,l,t]\mathbf{R}_L[j,l,t]^T = \mathbf{I} \\
&\mathbf{R}_L[j,l,t] \text{ represents a right handed frame}
\end{aligned} \tag{5.2}$$

Where $\mathbf{p}_{j+1,j}$ is the constant position vector of the next joint as seen in the frame of the previous joint, and $\mathbf{z}_{j-1,j}$ is the constant orientation of the next joint as seen in the frame of the previous joint, where ours are $\mathbf{z}_{j-1,j} = [0,0,1]^T$.

Collision Avoidance with Environment To enforce constraints such that LIMMS and boxes does not collide with the environment, we model the environment into discrete convex regions. All boxes and LIMMS have to stay within the convex regions during the process. We also need to discriminate if the LIMMS or box is making contact with the ground. This introduces additional binary variables $\delta_{Ba,i}$, $\delta_{Bg,i}$, $\delta_{La,i}$, and $\delta_{Lg,i}$. Note subscript g stands for ground, and a stands for air. If LIMMS and boxes are within a convex region, the joint points of LIMMS and corners of boxes are linear combinations of the vertices of the convex region:

$$\mathbf{p} = \sum_v \lambda_v \mathbf{V}_v, \quad \sum_v \lambda_v = \delta, \quad \lambda_v \in [0, 1] \tag{5.3}$$

Where $\mathbf{p}$, λ and δ are associated with either corner points of the box $\mathbf{c}_B$ or position of joints of LIMMS $\mathbf{p}_L$ as listed in Table 5.1 and Table 5.2. λ_v's represent the vertices of the convex region. If one region is not selected, all λ_v's are zero due to $\delta = 0$.

Collision Avoidance Between Agents To enforce collision avoidance for LIMMS-LIMMS and LIMMS-box contact, we use the formulation from [64] that uses separating planes. For convex polygons, the two polygons do not overlap with each other if and only if there exists a separating hyperplane $\mathbf{a}^T\mathbf{x} = b$ in between [68]. That is, for any point $\mathbf{p}_1$ inside polygon 1 then $\mathbf{a}^T\mathbf{p}_1 \leq b$, and for any point $\mathbf{p}_2$ inside polygon 2 then $\mathbf{a}^T\mathbf{p}_2 \geq b$. Our problem uses the following constraints:

$$\begin{gathered} \mathbf{a}^T\mathbf{p}_L[j, l_1, t] \leq b, \;\; \mathbf{a}^T\mathbf{p}_L[j, l_2, t] \geq b \\ \mathbf{a}^T\mathbf{c}_B[c, b, t] \leq b, \;\; \mathbf{a}^T\mathbf{p}_L[j, l, t] \geq b \; \forall t, \;\; \mathbf{a}^T\mathbf{a} \geq 0.5 \end{gathered} \tag{5.4}$$

Where $\mathbf{a}$ and b is the normal vector and offset for planes associated with the specific pair. 0.5 is just an arbitrary nonzero number that we choose, as $\mathbf{a}$ does not necessarily need to be a unit vector. With this method, we enforce collision avoidance purely through inequality constraints and avoid using complementary constraints such as [69].

Dynamics for Box The dynamics are required for the agent to generate strictly feasible motions. However, enforcing dynamics for each LIMMS is expensive given its high DOF. We only enforce dynamics for the boxes. This serves as two purposes. First, it allows the system to generate dynamic motions such as throwing or jumping. Second, it allows the system to select motion plans based on the box weight.

When the box is in stable object mode, the gravity is compensated for by the ground. Additionally, LIMMS can only apply reaction forces to the box when it is connected through the anchor points on the box. We define the reaction force on the end effector of a LIMMS l to the box b as $\mathbf{f}_i[b, s, l, t]$, where the index s indicates that the force is through the anchor point s_b. When LIMMS connects to the box as a leg, $\mathbf{f}_i$ serves as the contact force on the ground, while when LIMMS connects to the box as an arm, $\mathbf{f}_i$ serves as the contact force to grasp the box. The box dynamics are:

$$m\ddot{\mathbf{p}}_{box}[b, t] = \sum_{s=1}^{4} \sum_{l=1}^{L} \mathbf{f}_i[b, s, l, t] - m\mathbf{g}(1 - z_{B,1}[b, t]) \tag{5.5}$$

$$\sum_{s=1}^{4} \sum_{l=1}^{L} (\mathbf{p}_L[j = 6, l, t] - \mathbf{p}_B[b, t]) \times \mathbf{f}_i[b, s, l, t] = 0 \tag{5.6}$$

$\mathbf{f}_i[b, s, l, t]$ only exists when l is connected to the box:

$$\mathbf{f}_i[b, s, l, t] = 0 \;\text{ if }\; z_{S,1}[b, s_b, l, t] = 1 \tag{5.7}$$

When l connects to the box as leg mode, $\mathbf{f}$ represents the contact force from the ground. Therefore, $\mathbf{f}$ needs to satisfy the friction cone constraint:

$$\mathbf{f} \in Cone \;\text{ if }\; z_{S,i}[b, s_b, l, t] = 1 \text{ and } \delta_{Lg}[j = 6, l, \exists p, t] \tag{5.8}$$

Since LIMMS has a loading capacity, we enforce the max norm constraint on any contact force: $||\mathbf{f}|| \leq f_{max}$. Note that there is a contact moment across the latching. Missing this moment results in the box incapable of being manipulated with a single contact (moment balance will be violated). For simplicity, we fix the box orientation: $\mathbf{R} = \mathbf{I}$. This can be justified as in manipulated mode, the contact moment is sufficient to keep the orientation of the box, and in quadruped mode, the four ground support points is sufficient to keep the body orientation.

Support Polygon The balance of moment constraint should guarantee the stability of the quadruped body. However, there are already many nonlinear constraints such as kinematics. To simplify the NLP formulation, one approach is to enforce a simple stability constraint in replacement of the moment balance constraint. Since the rotation matrix is fixed, we can simply enforce that the end effector of LIMMS stays to one side of the body which guarantees that the body's center of mass lies within the support polygon of the foot.

Stability of LIMMS One drawback of our formulation is that we do not include the dynamics of LIMMS. As a compensation, there should be a constraint to guarantee the stability when LIMMS is on the ground and tries to reach an anchor point. We just assume that we have many latching points on the ground so realizing this motion is relatively simple. Therefore, when LIMMS is in free mode, we enforce that the base stays on the ground and give a speed constraint:

$$||\mathbf{p}_L[j, l, t+1] - \mathbf{p}_L[j, l, t]|| \leq \Delta\mathbf{P} \quad \text{if} \quad \delta_{Lg}[j = 0, l, p, t] = 1 \tag{5.9}$$

Continuity of Connection One suboptimal solution to avoid is having LIMMS frequently latch on and off an anchor point. We enforce equality constraints for binary variables within a range $z_{S,i}[t] = ... = z_{S,i}[t + n]$. We usually choose n

between 3 and 5, decided based on the speed of latching.

5.3 ADMM Formulation

Collecting the constraints defined previously, the problem to solve becomes:

$$
\begin{aligned}
&\underset{z,\ \delta,\ \mathbf{p},\ \mathbf{R},\ \mathbf{f},\ \lambda,\ \mathbf{a},\ b}{\text{minimize}} \quad f_{obj} \\
&\text{subject to} \\
&\textbf{Mixed integer constraints:} \\
&\text{Logic rules 1-12} \\
&\text{collision avoidance with the environment Eq. (5.3)} \\
&\text{Dynamics constraints: Eq. (5.5), Eq. (5.7), Eq. (5.8)} \\
&\text{Stability of LIMMS on ground Eq. (5.9)} \\
&\textbf{Nonlinear constraints:} \\
&\text{Kinematics Eq. (5.2)} \\
&\text{Collision avoidance between agents Eq. (5.4)} \\
&\text{Dynamics constraints Eq. (5.6)}
\end{aligned}
\tag{5.10}
$$

Where f_{obj} is a quadratic equation that minimizes the distance of the box to the goal position. The variables and constraints of problem Eq. (5.10) incorporate discrete and continuous variables with linear and nonconvex (bilinear) constraints. This results in an MINLP. The commercial solvers tend to perform non-satisfactory in this type of problem. There are generally two approaches that convert this type of problem: an MICP using convex envelope relaxations for nonlinear constraints [67] or conversion of the discrete variables into continuous ones through complementary formulation [70]. MIPs with convex envelopes tend to solve slowly when the problem scales up. Since there are many discrete variables in this problem, complementary formulations will be numerically difficult [66]. In this section, we adopt the ADMM. ADMM separates the problem into two sub-problems. Although those sub-problems have different constraints, ADMM iterates between sub-problems such that constraint 1 which may not appear in sub-problem 2 will be implicitly enforced as the iteration proceeds. In the end, sub-problems will reach a consensus meaning their solutions are close to each other. This procedure is detailed in Algorithm 1. In our problem, the logic rule constraints are resolved through MIPs, while the nonlinear kinematics and collision avoidance constraints are resolved through NLPs. Similar to [65], we first make copies $\mathbf{var}_2$ of the variables $\mathbf{var}_1 = [z, \delta, \mathbf{p}, \mathbf{R}, \mathbf{f}, \lambda, \mathbf{a}, b]$. Represent the feasible set of mixed-integer constraints through $0 - \infty$ indicator function by $\mathcal{I}_M$ and the nonlinear constraints by $\mathcal{I}_N$. The consensus problem between MIP and NLP is:

$$\begin{aligned} &\underset{\mathbf{var}_1\ \mathbf{var}_2}{\text{minimize}} \quad f_{obj} + \mathcal{I}_M(\mathbf{var}_1) + \mathcal{I}_N(\mathbf{var}_2) \\ &s.t. \quad \mathbf{var}_1 = \mathbf{var}_2 \end{aligned} \tag{5.11}$$

The constraints are moved to the objective function through the indicator function. Applying ADMM [71] to the Lagrangian $\mathcal{L}$ of Eq. (5.11) results in three iterative operations:

$$\mathbf{var}_1^{i+1} = \text{argmin}_{\mathbf{var}_1} \mathcal{L}(\mathbf{var}_1^i,\ \mathbf{var}_2^i,\ \mathbf{w}^i) \tag{5.12a}$$

$$\mathbf{var}_2^{i+1} = \text{argmin}_{\mathbf{var}_2} \mathcal{L}(\mathbf{var}_1^{i+1},\ \mathbf{var}_2^i,\ \mathbf{w}^i) \tag{5.12b}$$

$$\mathbf{w}^{i+1} = \mathbf{w}^i + \mathbf{var}_1^i - \mathbf{var}_1^i \tag{5.12c}$$

Where $\mathbf{w}$ is the dual variable of the Lagrangian of Eq. (5.11). In Eq. (5.12), Eq. (5.12a) solves the MIP problem:

$$\begin{aligned} &\underset{\mathbf{var}_1}{\text{minimize}} \quad ||\mathbf{var}_1^i - \mathbf{var}_2^i + \mathbf{w}^i||_{\mathbf{W}^k_{\text{MIP}}} \\ &\text{s.t. Mixed-integer constraints in Eq. (5.10)} \end{aligned}$$

In the next step, Eq. (5.12b), solves the NLP:

$$\begin{aligned} \underset{\mathbf{var}_2}{\text{minimize}} \quad & ||\mathbf{var}_2^i - (\mathbf{var}_1^{i+1} + \mathbf{w}^i)||_{\mathbf{W}_{\text{NLP}}^k} \\ & \text{s.t. Nonlinear constraints in Eq. (5.10)} \end{aligned}$$

And the next step, Eq. (5.12c), updates the dual variable $\mathbf{w}$. To finish one iteration, the weights for MIP, W_{MIP}, the weights for NLP, W_{NLP}, and the dual variable $\mathbf{w}$, are updated with line $6-7$ in Algorithm 1. Within one iteration, Eq. (5.12a), Eq. (5.12b), Eq. (5.12c) are solved in succession. This iterative procedure continues until the discrepancy between the MIP solutions and the NLP solutions $\boldsymbol{\theta} = \mathbf{var}_1^i - \mathbf{var}_1^i$ are lower than the user-set error threshold $\boldsymbol{\theta}_{th}$.

It is well known that ADMM has convergence guarantees for convex problems and can significantly improve the solving speed. However, for complex MINLPs, there is no convergence guarantee. In this problem, both MIP and NLP can be slow and expensive. We avoid explicitly placing complementary constraints to represent discrete modes as [65] did, since it hinders convergence for NLP. However, NLP does need some information on discrete variables as it needs to reason connections and turn variables such as $\mathbf{f}$ on or off accordingly. After solving the MIPs, we directly use the solutions of $z_{S,i}$, $z_{W,i}$ in the NLP step to enforce connections. This improves the precision of the NLP step and the overall precision of consensus. The price to pay is an increase in difficulty for NLP solvers to find solutions.

5.4 Results

We performed 5 numerical experiments to evaluate the performance of the proposed formulation and ADMM algorithm. For all experiments, the MIP formulation was solved with Gurobi 9, and the NLP formulation was solved through Ipopt on a Intel Core i7-7800X 3.5GHz × 12 machine. We solved all the scenarios

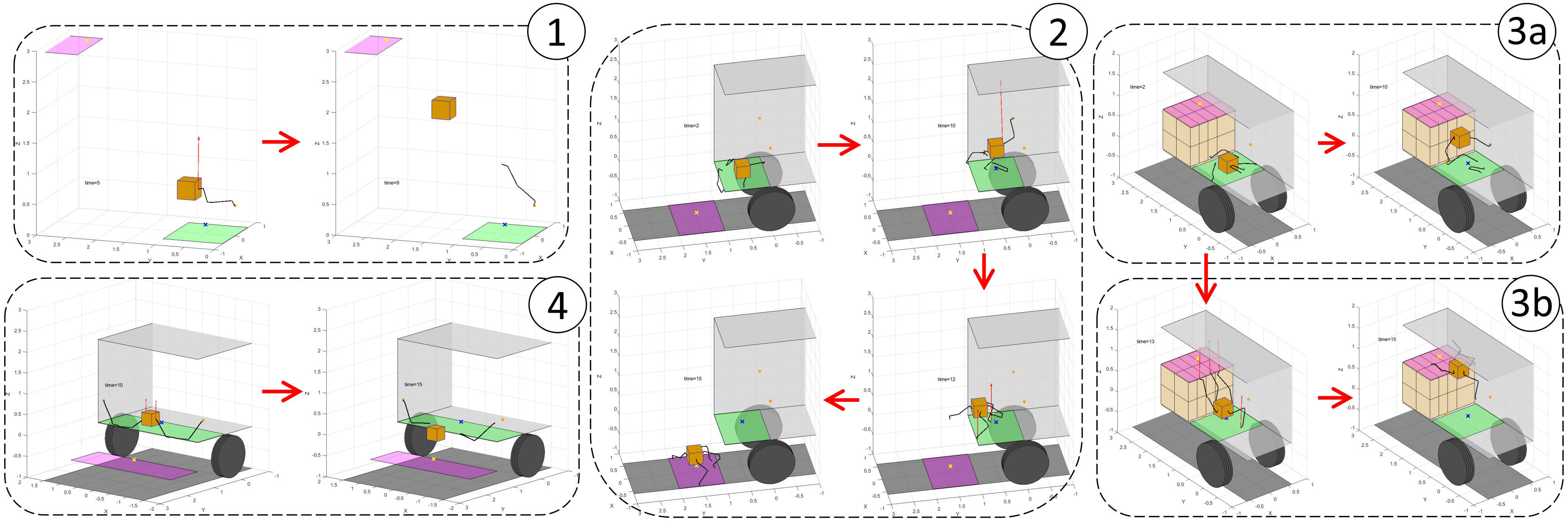


Figure 5.3: Results for 5 experiments. The green rectangle represents the initial region in which LIMMS is constrained in. LIMMS is trying to get the box to the magenta region, where LIMMS is allowed to move in as well. 1) LIMMS picks up a box and throws it towards the goal. 2) LIMMS lifts box in manipulator mode and switches to quadruped mode to jump out of the vehicle. 3a) LIMMS attempts to use manipulator mode to reach goal. 3b) LIMMS climbs onto a step using quadruped mode. 4) LIMMS sends the box towards the goal using dual double arm manipulation to enlarge the workspace.

Algorithm 1 ADMM for LIMMS

```
Input $\rho$, $\mathbf{W}^0_{\text{MIP}}$, $\mathbf{W}^0_{\text{NLP}}$, $\mathbf{w}^0$, $\mathbf{var}^0_2$, $\boldsymbol{\theta}_{th}$
Initialization $i = 1$
while $\boldsymbol{\theta} > \boldsymbol{\theta}_{th}$ and $i < i_{max}$ do
    Compute $\mathbf{var}^{i+1}_1$ via $Eq.$ (5.12a)
    Compute $\mathbf{var}^{i+1}_2$ via $Eq.$ (5.12b)
    $\mathbf{w}^{i+1} \longleftarrow \mathbf{w}^i + \mathbf{var}^i_1 - \mathbf{var}^i_1$
    $\mathbf{W}^{k+1}_{\text{MIP}} \longleftarrow \rho\mathbf{W}^k_{\text{MIP}}$,   $\mathbf{W}^{k+1}_{\text{NLP}} \longleftarrow \rho\mathbf{W}^k_{\text{NLP}}$
    $\mathbf{w}^{i+1} \longleftarrow \mathbf{w}^i/\rho$
    $\boldsymbol{\theta} \longleftarrow \mathbf{var}^i_1 - \mathbf{var}^i_1$
    $i = i + 1$
return $\mathbf{var}^i_2$
```

for 15 iterations. Other parameters we have chosen are: $\Delta t = 1sec$, $f_{max} = 20N$, max moving speed for LIMMS on the ground of $0.5m/s$, and a $30cm$ cube box. The 6-DOF LIMMS unit lengths are $5cm$, $7.4cm$, $33cm$, $7.8cm$, $33cm$, $7.4cm$, $5cm$, respectively for each link starting at one end. Fig. 5.3 shows the MATLAB visualization of the first 4 experiments. Animations of all tests can be seen in the online video link related to published work [24] [2].

Throwing In the first experiment as seen in Fig. 5.3 (1), we placed 1 LIMMS and 1 box in the scene and provided 1 anchor point on the wall. The goal is set higher than a LIMMS unit can reach. The solver gives a solution where the LIMMS unit connects to the wall and picks up the box, then throws the box to the goal. This simple test shows the ability of the solver to generate manipulation motions using dynamics.

Jumping In the second experiment as depicted in Fig. 5.3 (2), we placed 4 LIMMS and 1 box on a raised platform or inside a truck and set the goal position to be lower on the ground. In addition, 2 anchor points are provided on the wall. The planner provides the solution where 1 LIMMS latches itself onto the wall and lifts the box. Then the other 3 LIMMS connects to the anchor points on the

---

[2] Online video link to this sections related publication [24] showing planner under different scenarios: https://www.youtube.com/watch?v=RH9gMOK24L0

sides of the box which enters quadruped mode. This quadruped robot then jumps down the step to reach the goal. The solver automatically changes modes from manipulation to quadruped.

Weight Lifting In the third experiment as shown in Fig. 5.3 (3a) and Fig. 5.3 (3b), we investigate the behavior change due to a change in the weight of the box. We place 4 LIMMS and 1 box down a ledge and set the goal to be above the ledge. Two anchor points on the wall are provided. First, we set the box weight to be 0.5 kg. The planner provides a manipulation trajectory where two LIMMS connect to the wall and lift the box to move it to the goal. We then increase the box weight to 7 kg. In this case, if we force LIMMS to manipulate the object, the planner returns infeasible. The feasible trajectory returned by the solver set 3 LIMMS to be anchored onto the ledge and box, while 1 leg stays down from the ledge to push the body up onto the platform. As the box weight exceeds the capacity of dual arm manipulation, a quadruped motion is necessary to lift the 7 kg box.

Manipulation with Double-dual Arms In the fourth experiment in Fig. 5.3 (4), we again place 4 LIMMS and 1 box in the same scenario, below a desired platform or step. Two anchor points are provided on the wall. However, we enlarged the width of the truck such that 1 LIMMS cannot reach the box from the wall. In this case, the solver connected a second LIMMS as an additional arm to the first which is connected to the wall, allowing the arms to reach the box and perform dual arm manipulation.

Quadruped Walking with Refinement In the fifth experiment, we placed 4 LIMMS and a box on flat ground and set the goal to be at the same elevation but separated by a distance. LIMMS moved to the box and assembled a quadruped robot which then moved towards the goal. Furthermore, since there is no gait

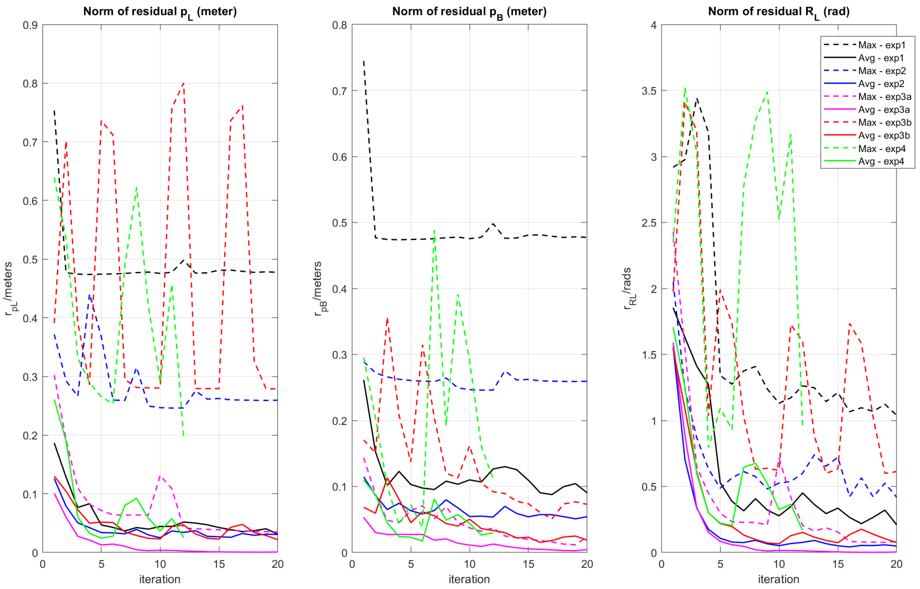


Figure 5.4: Convergence of mean and max residual r_{pB}, r_{pL}, r_{RL} for experiment 1-4. Solid lines denote mean residuals and dashed lines denote max residuals.

optimization in our problem formulation, we used an additional planner similar to [72] but included 6-DOF kinematics to provide the quadruped walking motion to the goal. This experiment demonstrated that, although the planner may give rough trajectories, they can be further refined with another planner to correct kinematics discrepancies and gait cycles. This readies it to be implemented on the hardware.

Box Lifting on Hardware In this experiment, one LIMMS is initially anchored to the ground. There is another anchor point on the wall. The objective is to lift the box higher. If LIMMS lifts the box with the other end anchored to the ground, the kinematics quickly becomes infeasible. The planner instead lets LIMMS first anchor to the box, then anchor to the wall from the box. The LIMMS can easily lift the box higher with the other end anchored to the wall. The hardware implementation is included in the video.

#	T	# of variables		# of constraints		Time in Minutes		
		MIP	NLP	MIP	NLP	T-MIP	T-NLP	T-Total
1	10	4857 cont. 670 bin.	1777	3623	eq. 582 ineq. 5206	0.88	7.17	8.05
2	15	25917 cont. 3150 bin.	8787	34545	eq. 621 ineq. 28990	129	24	153
3a)	15	25917 cont. 3150 bin.	8787	33102	eq. 1686 ineq. 28990	41	107	148
3b)	10	17277 cont. 2100 bin.	5857	22032	eq. 1131 ineq. 19180	5	50	55
4	15	25917 cont. 3690 bin.	8787	35924	eq. 606 ineq. 31342	19	244	263

Table 5.4: Solving time for experiment 1-4. Note: Far left column in ascending order is: 1 for Throw, 2 for Jump, 3a for Lifting with Arm, 3b for Lifting with Climb, and 4 for Double-dual Arm.

Convergence Define the residual to be the mismatch between the MIP and NLP solutions. The mean residuals for $\mathbf{p}_B$, $\mathbf{p}_L$ and $\mathbf{R}_L$ are the mean value of all the norms:

$$r_{pB}[i] = \underset{b}{\text{mean}}\ ||\mathbf{p}_{B,\text{MIP}}[b,i] - \mathbf{p}_{B,\text{NLP}}[b,i]|| \tag{5.13a}$$

$$r_{pL}[i] = \underset{j,\ l}{\text{mean}}\ ||\mathbf{p}_{L,\text{MIP}}[j,l,i] - \mathbf{p}_{L,\text{NLP}}[j,l,i]|| \tag{5.13b}$$

$$r_{RL}[i] = \underset{j,\ l}{\text{mean}}\ ||\text{Vec}(\mathbf{R}_{L,\text{MIP}}[j,l,i] - \mathbf{R}_{L,\text{NLP}}[j,l,i])|| \tag{5.13c}$$

The max residual is the maximum value of all the norms. Fig. 5.4 depicts the change of mean and max residuals as a function of time. ADMM generally showed decent average consensus after iteration 10, where r_{pB} or r_{pL} usually converges to cm-mm level and r_{RL} usually less than 0.1 rad. The maximum residual can sometimes be large. If we put the MIP solutions on the real hardware, we can run another kinematics refiner to solve the kinematics to ensure that the nonlinear constraints are strictly satisfied.

The number of variables, constraints and time cost for solving the experiments above are listed in Table 5.4. Generally, the NLP portion is the more challenging portion of ADMM, since it includes kinematic and collision avoidance constraints. To speed up the solving process, some linear constraints in can be moved into the MIP formulations. This will speed up the NLP solver but the residual may increase.

5.4.1 Conclusion

Here we have demonstrated an offline, mid-level planner capable of complex motions that coordinate between multiple LIMMS to achieve a common task. In some instances even a combination of discrete modes appeared. In general this is a difficult problem to solve given its a nonlinear MIP. Solving this offline allows for a discovery of motion primitives given a task. Data from these offline trajectories given a problem state can be used online using a machine learning algorithm and function approximators. Recent research may suggest that these nonlinear MIPs could be solved online using prior solved problems such as the ones from our offline generated database [53].

CHAPTER 6

Conclusion

6.1 Implementation

The main goal for this project was to demonstrate that this idea was feasible. We built and assembled a wireless LIMMS with the second iteration of the prototype. In order to improve wheel speed another motor was added. This increased the difficulty of quadruped mode seen in Fig. 6.1 (*top left*) due to the increase in weight. We actually used heuristics from Chapter 4 to get quadruped to stand up. Fig. 6.1 (*top right*) shows a workaround that had to be done in order to get LIMMS quadruped mode to climb a step due to the weight. In future iterations a custom end effector is needed where one motor has two output shafts, one to the wheel and one to the latch. Fig. 6.1 (*bottom left*) shows a new mode that was not present in the original concept. This mode has *three wheels* in the sense of distinct contact points. This is useful when returning back to the truck and there is an obstacle. Fig. 6.1 (*bottom right*) is another mode not in the original concept. This mode has all four wheels aligned with the middle joint connected with the other LIMMS unit. This mode drives very smoothly and should be used on flat surfaces. A passive wheel should be added at the middle joint.

LIMMS ended up completing each sub-task individually but had trouble completing the entire task in one take. Overall, the setup time for the entire system slowed progress time. It is understandable given it is an early prototype. However, for future development reducing time for battery changes and calibration would improve productivity. Despite the extra setup time, it is very possible to get a working prototype to complete the entire task from start to finish, and our

Figure 6.1: Frames taken from the LIMMS demonstration moving a box out of the truck and delivering it to a location and returning. (*Top Left*) shows LIMMS quadruped mode rolling down the ramp. (*Top Right*) depicts a workaround that had to be made to compensate for the extra motors to get LIMMS to climb a step. (*Bottom Left*) illustrates a three wheeled mode used to return when overcoming obstacles. (*Bottom Right*) shows an aligned four wheel mode used to traverse flat surfaces to return.

limited demonstration proves that.

6.2 Miscellaneous

In this section we cover some of the additional contributions made by the author to the robotics community not covered yet. In 2019 a novel parachute mechanism was introduced in [73]. Later that year we published an optimization paper for vertical climbing of a six-legged robot in [74]. In 2020 we published work for creating *safe* linear quadratic regions in function approximators in [75]. In 2022 we presented an optimization formulation for bookshelf organization problem with obstacle avoidance in [76]. In the later half of that year we also published a warm-start method to solve MIPs online in [64]. Finally, we went to RoboCup 2023 a robotic soccer competition. We built an additionally state of the art adult-sized humanoid robot in 6 months. The author also developed a key localization method that surpassed the competitors.

6.3 Summary & Final Thoughts

In this manuscript, LIMMS was introduced as a modular, multi-modal robotics concept for last-mile delivery. Based on real-world data, the design requirements and assumptions were laid out and tested. The LIMMS kinematics and design were then shown to have a better workspace than a traditional 6-DOF robotic arm and not suffer from asymmetric torque imbalances.

LIMMS fills a niche between the general purpose and task-specific robots. While it is suited mainly for manipulation tasks, it still remains adaptable for application on most other delivery tasks. The compact design allows multiple LIMMS to replace larger, complex systems. This comes at a cost. For instance, the wheeled delivery robot may be more energy efficient than the LIMMS quadruped unit. However, the wheeled robot is effectively constrained to a specific type of terrain. LIMMS is not. For a humanoid delivery robot, the arms and legs of the robot are constrained by the physical structure of its back and hip widths. This may be a disadvantage for very heavy or awkwardly shaped payloads as those spacing constraints limit the methods in which to carry the box and how many humanoids work together on a single box. LIMMS can potentially scale up to carry heavier loads than a humanoid because it does not have the same physical spacing constraints. Moreover, when scaling the number of LIMMS units it can squeeze into a number of places and attach itself on walls or even roofs.

The generalizability of LIMMS additionally opens up new avenues of optimization for logistics never before seen. As long as there are attachment points, LIMMS can latch its base link to it, meaning that for warehouses there is a potential to not only optimize traffic of automated systems on the ground but vertically as well. Mixing and matching all of the modes can bring about much more efficient delivery logistics. For example, the wheels on LIMMS could be combined with manipulator mode to form a dynamic mobile conveyor belt system to move stacks of boxes. Sub-tasks such as queuing up the next delivery or other portions of logistics can be handled by LIMMS.

We also analyzed a class of self-aligning rotational latching mechanisms and manufactured prototypes for use on LIMMS. The probability distributions for both 2-blade and 3-blade designs were formally derived and verified using MC and CF. By inspection of Fig. 3.13, it is clear that the 2-blade design outperforms the others in terms of self-alignment. As the number of blades increased, the rate of success decreased overall. From the probability distribution, a new metric Ψ was introduced to compare relative designs and quantify alignment tolerance. This metric also demonstrated that the 2-blade latching mechanism achieves self-alignment more consistently than the other designs. Depending on the specific application, using both Ψ and the probability distribution may be preferable when designing the latch as there may be subtle trade-offs between robustness in (x, y) and θ. Our analysis also showed that the 2-blade latch can be misaligned as far as $3\rho_{in}^*$ away and still align successfully with the hole pattern.

Our analyses were formalized as a design process for this type of latching mechanism to find optimal design specifications for specific constraints as laid out in Fig. 3.15. We then used this methodology to manufacture both a 2-blade and 3-blade latching mechanism. The theory was then tested using these mechanisms with a robotic arm holding a single off-center position with PID control. The results of these experiments with two different surface materials were consistent with our theory, and the prototype demonstrated superior alignment capabilities compared to existing methods.

To increase the strength of LIMMS when loads over torqued the actuators, we used theory from kineto-static duality. By following trajectories in regions where the structure absorbed loads, the weight capacity increased effectively reducing the overall torque requirements for LIMMS. Analyzing the a radial slice of the max torque needed to compensate for gravity force, it was discovered there was a curved manifold near the vertical center. This differed from other robotic manipulator peers who all had completely flat vertical centers. Having a curved manifold at the inner edge admits a continuum of directions of nullspaces in the Jacobian. This can, therefore, be used in many different situations. This method also maintains a linear map meaning the same analysis holds for differing weights in the same direction, whereas the equation of motions do not. This method was then demonstrated on the BEAR LIMMS prototype to move ~ 75 lbs although failing to even hold its own body weight in some positions. This method should be regarded as a general method even if it is used as a heuristic for optimizing a trajectory.

Finally, an optimization based motion planner was presented for the multi-agent modular robot system LIMMS. We demonstrated solving the proposed formulation with ADMM. The results show how LIMMS autonomously coordinates between different modes and generates trajectories of the system under different situations. With proper refinement, the trajectories can be implemented on the hardware. It is worthwhile to mention that due to the separating plane for collision avoidance, the NLP part of the problem takes a long time to solve. In the previous work [64], we used data-driven methods to solve those constraints fast online among numerous of other methods to transform offline data to solve problems online.

We have introduced LIMMS and built the necessary tools required to get a bare minimum demonstration to work. Throughout the process we have researched and developed technologies related to latches and surface engagement, energy and torque reductions in systems, and mid-level multi-agent planning. More research and development will be needed in the near future to make this idea become a reality but with this work we are close. To an extent even some sub-tasks or use cases may be feasible with what was presented here.

APPENDIX A

Supplementary Materials for Chapter 3

Probability Distributions

Several interesting higher resolution CF probability distribution functions are shown in this section for 2 and 3 blade designs on the next page. In the three blade designs, there are subtle details, and there does not seem to be one consistent pattern as the island (red circle) scales. The red lines in the figure show the separation of regions that indicate symmetry. Please note that in this figure the relative scale between figures was distorted in favor of more details.

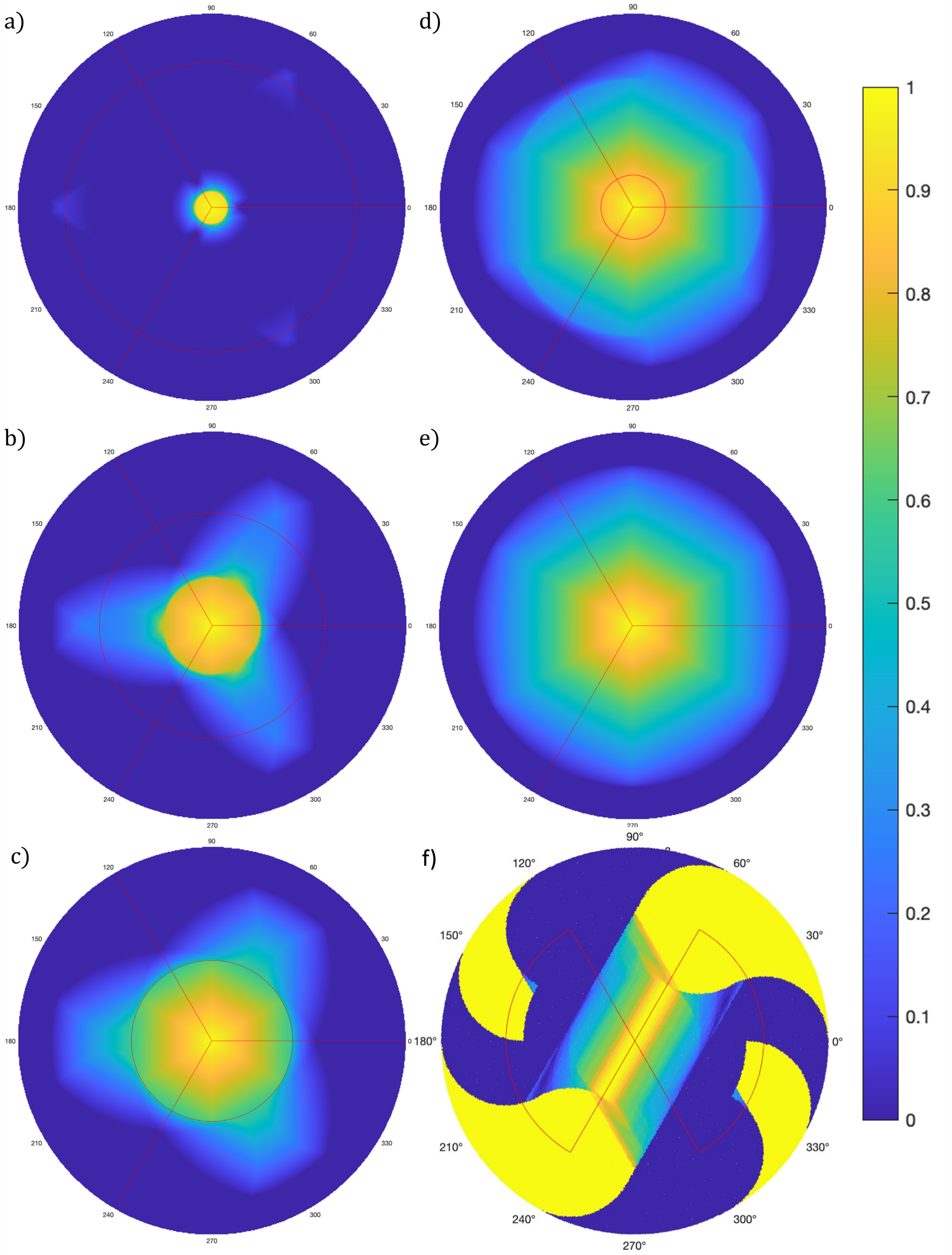


Figure A.1: Probability distribution functions: a) 3-blade r=0.9, b) 3-blade r=0.7, c) 3-blade r=0.5, d) 3-blade r=0.2, e) 3-blade r=0, and f) 2-blade $r \to 0$ with $\vartheta > \vartheta^*$.

References

[1] D. Jennings and M. Figliozzi, "Study of sidewalk autonomous delivery robots and their potential impacts on freight efficiency and travel", *Transportation Research Record*, vol. 2673, no. 6, pp. 317–326, 2019.

[2] T. Zhu, G. Fernandez, C. Togashi, Y. Liu, and D. Hong, "Feasibility study of limms, a multi-agent modular robotic delivery system with various locomotion and manipulation modes", in *2022 19th International Conference on Ubiquitous Robots (UR)*. IEEE, 2022.

[3] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, et al., "Anymal-a highly mobile and dynamic quadrupedal robot", in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 38–44.

[4] T. Tanev and B. Stoyanov, "On the performance indexes for robot manipulators", *Problems of Engineering Cybernetics and Robotics*, vol. No.49, pp. 64–71, 01 2000.

[5] M. Caballer Castells, "Control of a ur5e robotic arm through a simulate 2d camera", 2020.

[6] H. Adachi, N. Koyachi, T. Arai, and K.-I. Nishimura, "Control of a manipulator mounted on a quadruped", in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*. IEEE, 1996, vol. 2, pp. 883–888.

[7] H. Chai, Y. Li, R. Song, G. Zhang, Q. Zhang, S. Liu, J. Hou, Y. Xin, M. Yuan, G. Zhang, and Z. Yang, "A survey of the development of quadruped robots: Joint configuration, dynamic locomotion control method and mobile manipulation approach", *Biomimetic Intelligence and Robotics*, vol. 2, no. 1, pp. 100029, 2022.

[8] K. Nagasaka, Y. Kawanami, S. Shimizu, T. Kito, T. Tsuboi, A. Miyamoto, T. Fukushima, and H. Shimomura, "Whole-body cooperative force control for a two-armed and two-wheeled mobile robot using generalized inverse dynamics and idealized joint units", in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 3377–3383.

[9] L. P. Ramalepa and R. S. Jamisola, "A review on cooperative robotic arms with mobile or drones bases", *International Journal of Automation and Computing*, vol. 18, no. 4, pp. 536–555, 2021.

[10] D. B. Poudel, "Coordinating hundreds of cooperative, autonomous robots in a warehouse", *Jan*, vol. 27, no. 1-13, pp. 26, 2013.

[11] M. Bjelonic, R. Grandia, O. Harley, C. Galliard, S. Zimmermann, and M. Hutter, “Whole-body mpc and online gait sequence generation for wheeled-legged robots”, in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8388–8395.

[12] J. Hooks, M. S. Ahn, J. Yu, X. Zhang, T. Zhu, H. Chae, and D. Hong, “Alphred: A multi-modal operations quadruped robot for package delivery applications”, *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5409–5416, 2020.

[13] M. Bjelonic, C. D. Bellicoso, Y. de Viragh, D. Sako, F. D. Tresoldi, F. Jenelten, and M. Hutter, “Keep rollin’—whole-body motion control and planning for wheeled quadrupedal robots”, *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2116–2123, 2019.

[14] M. E. Stieber, D. G. Hunter, and A. Abramovici, “Overview of the mobile servicing system for the international space station”, *EUROPEAN SPACE AGENCY-PUBLICATIONS-ESA SP*, vol. 440, pp. 37–42, 1999.

[15] H. Cruijssen, M. Ellenbroek, M. Henderson, H. Petersen, P. Verzijden, and M. Visser, “The european robotic arm: A high-performance mechanism finally on its way to space”, in *42nd Aerospace Mechanisms Symposium*, 2014.

[16] B. Ma, Z. Jiang, Y. Liu, and Z. Xie, “Advances in space robots for on-orbit servicing: A comprehensive review”, *Advanced Intelligent Systems*, p. 2200397, 2023.

[17] C. Roman, D. Igor, T. Victor, and T. Alexander, “The active test facility and experimental study of the multi-link space robot in ground conditions”, in *2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation (ROMA)*. IEEE, 2017, pp. 1–6.

[18] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, “Modular self-reconfigurable robot systems [grand challenges of robotics]”, *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43–52, 2007.

[19] B. Jenett and K. Cheung, “Bill-e: Robotic platform for locomotion and manipulation of lightweight space structures”, in *25th AIAA/AHS Adaptive Structures Conference*, 2017, p. 1876.

[20] M. H. Hasbulah, F. A. Jafar, and M. H. Nordin, “Comprehensive review on modular self-reconfigurable robot architecture”, *Int. Res. J. Eng. Technol*, vol. 6, no. 4, pp. 1317–1331, 2019.

[21] T. Fukuda and S. Nakagawa, “Dynamically reconfigurable robotic system”, in *Proceedings. 1988 IEEE international conference on robotics and automation*. IEEE, 1988, pp. 1581–1586.

[22] G. I. Fernandez, S. Gessow, J. Quan, and D. Hong, “Self-aligning rotational latching mechanisms”, in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2022.

[23] G. I. Fernandez, S. Gessow, J. Quan, and D. W. Hong, “Self-aligning rotational latching mechanisms: Optimal geometry for mechanical robustness”, *Journal of Mechanisms and Robotics*, vol. 16, no. 1, pp. 011007, 2024.

[24] X. Lin, G. I. Fernandez, Y. Liu, T. Zhu, Y. Shirai, and D. Hong, “Multimodal multi-agent optimization for limms, a modular robotics approach to delivery automation”, in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12674–12681.

[25] D. Bamburry, “Drones: Designed for product delivery”, *Design Management Review*, vol. 26, no. 1, pp. 40–48, 2015.

[26] “Types of curb and gutter”, Standard S-410-2, Bureau of Engineering, City of Los Angeles, 1989.

[27] “California building code”, California code of regulations, California Building Standards Commission, 2010.

[28] S. R. Buss and J.-S. Kim, “Selectively damped least squares for inverse kinematics”, *Journal of Graphics tools*, vol. 10, no. 3, pp. 37–49, 2005.

[29] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator”, in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2004, vol. 3, pp. 2149–2154 vol.3.

[30] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “Mit cheetah 3: Design and control of a robust, dynamic quadruped robot”, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2245–2252.

[31] M. U. Draz, M. S. Ali, M. Majeed, U. Ejaz, and U. Izhar, “Segway electric vehicle”, in *2012 International Conference of Robotics and Artificial Intelligence*, 2012, pp. 34–39.

[32] B. Gao, X. Zhang, W. Yao, Y. Guo, Y. Bao, and J. Xie, “Dynamics and motion control of an underactuated two-wheeled mobile robot”, in *2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2012, pp. 179–184.

[33] S. Seok, A. Wang, D. Otten, and S. Kim, "Actuator design for high force proprioceptive control in fast legged locomotion", in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1970–1975.

[34] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots", *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, 2016.

[35] M. I. Gökler and M. B. Koc, "Design of an automatic tool changer with disc magazine for a cnc horizontal machining center", *International Journal of Machine Tools and Manufacture*, vol. 37, no. 3, pp. 277–286, 1997.

[36] A. B. Hays, P. Tchoryk Jr, J. C. Pavlich, G. A. Ritter, and G. J. Wassick, "Advancements in design of an autonomous satellite docking system", in *Spacecraft Platforms and Infrastructure*. SPIE, 2004, vol. 5419, pp. 107–118.

[37] F. Gampe, K. Priesett, and R. Bentall, "A modular docking mechanism for in-orbit assembly and spacecraft servicing", in *NASA. Ames Research Center 19th Aerospace Mech. Symp.*, 1985.

[38] W. Mccown and N. Bennett, "Structural latches for modular assembly of spacecraft and space mechanisms", in *NASA. Langley Research Center, The 22nd Aerospace Mechanisms Symposium*, 1988.

[39] A. Sproewitz, A. Billard, P. Dillenbourg, and A. J. Ijspeert, "Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture", in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 4259–4264.

[40] C. Parrott, T. J. Dodd, and R. Groß, "Higen: A high-speed genderless mechanical connection mechanism with single-sided disconnect for self-reconfigurable modular robots", in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3926–3932.

[41] C. Liu, Q. Lin, H. Kim, and M. Yim, "Smores-ep, a modular robot with parallel self-assembly", *arXiv preprint arXiv:2104.00800*, 2021.

[42] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-tran: Self-reconfigurable modular robotic system", *IEEE/ASME transactions on mechatronics*, vol. 7, no. 4, pp. 431–441, 2002.

[43] M. Yim, D. G. Duff, and K. D. Roufas, "Polybot: a modular reconfigurable robot", in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. IEEE, 2000, vol. 1, pp. 514–520.

[44] T. Fukuda, T. Ueyama, Y. Kawauchi, and F. Arai, "Concept of cellular robotic system (cebot) and basic strategies for its realization", *Computers & electrical engineering*, vol. 18, no. 1, pp. 11–39, 1992.

[45] M. Nilsson, "Heavy-duty connectors for self-reconfiguring robots", in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. IEEE, 2002, vol. 4, pp. 4071–4076.

[46] H. M. Yip, Z. Wang, D. Navarro-Alarcon, P. Li, Y.-h. Liu, and T. H. Cheung, "A new robotic uterine positioner for laparoscopic hysterectomy with passive safety mechanisms: Design and experiments", in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3188–3194.

[47] A. Sproewitz, M. Asadpour, Y. Bourquin, and A. J. Ijspeert, "An active connection mechanism for modular self-reconfigurable robotic systems based on physical latching", in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3508–3513.

[48] N. Eckenstein and M. Yim, "The x-face: An improved planar passive mechanical connector for modular self-reconfigurable robots", in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3073–3078.

[49] Z. Aboura, N. Talbi, S. Allaoui, and M. Benzeggagh, "Elastic behavior of corrugated cardboard: experiments and modeling", *Composite structures*, vol. 63, no. 1, pp. 53–62, 2004.

[50] D. Noh, Y. Liu, F. Rafeedi, H. Nam, K. Gillespie, J.-s. Yi, T. Zhu, Q. Xu, and D. Hong, "Minimal degree of freedom dual-arm manipulation platform with coupling body joint for diverse cooking tasks", in *2020 17th International Conference on Ubiquitous Robots (UR)*, 2020, pp. 225–232.

[51] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues", *The International journal of Robotics research*, vol. 1, no. 1, pp. 4–17, 1982.

[52] S. Bruno, S. Lorenzo, V. Luigi, and O. Giuseppe, "Robotics: modelling, planning and control, 2010", *Cited on*, vol. 1, 1994.

[53] L. Xuan, "Generalized benders decomposition with continual learning for hybrid model predictive control in dynamic environment", *arXiv preprint arXiv:2310.03344*, 2023.

[54] C. Liu and M. Yim, "Reconfiguration motion planning for variable topology truss", in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1941–1948.

[55] M. Zhao, T. Anzai, F. Shi, X. Chen, K. Okada, and M. Inaba, "Design, modeling, and control of an aerial robot dragon: A dual-rotor-embedded multilink robot with the ability of multi-degree-of-freedom aerial transformation", *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1176–1183, 2018.

[56] T. Zhu, G. I. Fernandez, C. Togashi, Y. Liu, and D. Hong, "Feasibility study of limms, a multi-agent modular robotic delivery system with various locomotion and manipulation modes", in *2022 19th International Conference on Ubiquitous Robots (UR)*. IEEE, 2022, pp. 30–37.

[57] C. Liu, M. Whitzer, and M. Yim, "A distributed reconfiguration planning algorithm for modular robots", *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4231–4238, 2019.

[58] K. Gilpin, K. Kotay, D. Rus, and I. Vasilescu, "Miche: Modular shape formation by self-disassembly", *The International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 345–372, 2008.

[59] Y. Zhang, W. Wang, P. Zhang, and P. Huang, "Reinforcement learning-based task planning for self-reconfiguration of cellular satellites", *IEEE Aerospace and Electronic Systems Magazine*, 2021.

[60] N. Gandhi, D. Saldana, V. Kumar, and L. T. X. Phan, "Self-reconfiguration in response to faults in modular aerial systems", *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2522–2529, 2020.

[61] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning", in *2001 European control conference (ECC)*. IEEE, 2001, pp. 2603–2608.

[62] T. Kataoka, "A multi-period mixed integer programming model on reconfigurable manufacturing cells", *Procedia Manufacturing*, vol. 43, pp. 231–238, 2020.

[63] X. Lin, M. S. Ahn, and D. Hong, "Designing multi-stage coupled convex programming with data-driven mccormick envelope relaxations for motion planning", in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9957–9963.

[64] X. Lin, G. I. Fernandez, and D. W. Hong, "Reduce: Reformulation of mixed integer programs using data from unsupervised clusters for learning efficient strategies", in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4459–4465.

[65] A. Aydinoglu and M. Posa, "Real-time multi-contact model predictive control via admm", *arXiv preprint arXiv:2109.07076*, 2021.

[66] Y. Shirai, X. Lin, A. Schperberg, Y. Tanaka, H. Kato, V. Vichathorn, and D. Hong, "Simultaneous contact-rich grasping and locomotion via distributed optimization enabling free-climbing for multi-limbed robots", *IEEE Proc. 2022 IEEE/RSJ Int. Conf. Intell. Rob. Syst*, 2022.

[67] H. Dai, G. Izatt, and R. Tedrake, "Global inverse kinematics via mixed-integer convex optimization", *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1420–1441, 2019.

[68] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[69] A. U. Raghunathan, D. K. Jha, and D. Romeres, "Pyrobocop: Python-based robotic control & optimization package for manipulation and collision avoidance", *arXiv preprint arXiv:2106.03220*, 2021.

[70] O. Stein, J. Oldenburg, and W. Marquardt, "Continuous reformulations of discrete–continuous optimization problems", *Computers & chemical engineering*, vol. 28, no. 10, pp. 1951–1966, 2004.

[71] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers", *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[72] G. Bledt and S. Kim, "Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization", in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2019, pp. 6316–6323.

[73] X. Lin, G. Fernandez, S. Ghassemi, and D. W. Hong, "Feasibility study of an aerial lifting device using aerodynamic drag for ascent", in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* American Society of Mechanical Engineers, 2019, vol. 59247, p. V05BT07A016.

[74] X. Lin, J. Zhang, J. Shen, G. Fernandez, and D. W. Hong, "Optimization based motion planning for multi-limbed vertical climbing robots", in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2019, pp. 1918–1925.

[75] G. I. Fernandez, C. Togashi, D. W. Hong, and L. F. Yang, "Deep reinforcement learning with linear quadratic regulator regions", *arXiv preprint arXiv:2002.09820*, 2020.

[76] X. Lin, G. I. Fernandez, and D. W. Hong, "Benchmark results for bookshelf organization problem as mixed integer nonlinear program with mode switch and collision avoidance", *arXiv preprint arXiv:2208.13158*, 2022.