AN ABSTRACT OF THE THESIS OF

Lucas P. Hill for the degree of Master of Science in Robotics presented on June 4, 2018.

Title: Ballistic Rebounds for Wrapping a Target with a Casting Manipulator

Abstract approved: _____

Ross L. Hatton

Casting manipulation in robotics is the act of launching a tethered projectile and using it to interact with the environment. One such interaction is wrapping a target. In this work, we identify throwing and tether control parameters for successfully wrapping a target with an end-weighted cable, evolving through a set of methods to do this. Our particular focus is on parameters that are robust to variation in parameters. We then demonstrate successful wrapping of a target by a physical casting robot using this new approach.

Ballistic Rebounds for Wrapping a Target with a Casting Manipulator

by

Lucas P. Hill

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 4, 2018
Commencement June 2018

Master of Science thesis of Lucas P. Hill presented on June 4, 2018.

APPROVED:

Major Professor, representing Robotics

Director of the School of Mechanical, Industrial, and Manufacturing Engineering

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Lucas P. Hill, Author

ACKNOWLEDGEMENTS

I would like to acknowledge my adviser Dr. Ross Hatton and my committee, Dr. Ravi Balasubramanian, Dr. Matthew Johnston, and Dr. Jonathan Hurst. Members of the Laboratory for Robotics and Applied Mechanics, both past and present, with a special thank you to Hossein Faraji for being with me since the beginning. Members of the RDML and many individuals of the graduate student body in CoRIS. My family for being there to talk things out. And most importantly, Wendy Xu, my wife who's incredible support and patience with my trudge through gradschool has kept this enjoyable.

-Lucas Hill

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

Casting manipulation in robotics is the act of launching a tethered projectile and using the tether to retrieve the projectile or manipulate it mid-air. Robots that perform activities like extraplanar exploration would benefit from having a casting manipulator as it allows them to launch sensors or sample-gathering tools to locations nearby that would normally be too costly or even impossible to reach by driving or using a standard rigid arm.

Previous work in casting manipulation has examined techniques for precisely placing the tethered projectile, often with some type of grasping end-effector. We propose that the set of actions of a casting manipulator can be expanded by incorporating interactions between the tether and a target object. In particular, we examine wrapping a distal target with the tether so as to make a firm connection between the casting manipulator and the object.

In the simplest form, the trajectory of a wrap is a combination of a parabolic trajectory that transitions into a spiraling-in trajectory originating at the target. There is a possibility that in between the parabolic and the spiraling-in trajectory projectile enters onto a circular trajectory, centered on the anchor tether point. Wrapping with a casting manipulator is difficult because like most casting manipulator tasks, the initial conditions that set up the ballistic trajectory of the projectile are sensitive to small variations. These variations make a difference, as

once the projectile is air-born, the amount of control that can be exerted on the projectile is limited to a single pulling force applied through the tether. Further complicating the process, some ballistic trajectories require the pay out and retraction of the tether for the trajectory to be fully realized. We demonstrated this in our previous work where we found trajectories that smoothly transitioned (zero-jerk) from a parabolic ballistic trajectory to the spiraling-in trajectory of a wrap.

While no solutions have been developed for the specific problem of wrapping a target with a tethered projectile without feedback, there are ensemble control techniques from particle physics and computational optimization techniques that have solved similar problems. These techniques have been applied to solving high variance systems with kinematic constraints and solving dynamic systems with no variation. By combining these two techniques together we have developed a method that allows for the formulation of this problem as one that can be solved with standard optimization tools.

Using our tool and leveraging the geometry of the trajectories and the dynamics that govern their transitions, we propose that a wrap of a tethered projectile can be achieved using an imprecise throwing machine by selecting the correct combination of initial conditions: release angle, release velocity, and tether length. The particular dynamic we leverage is the moment when the tether goes taut, causing the projectile to change direction in a bounded manner. We select this moment by looking for points in the trajectory that are fairly robust to variation in the aforementioned initial conditions. For the rest of this thesis, we will refer to

this moment when the tether goes taut as the rebound.

Tethers, compared to similarly sized rigid manipulators, are lighter and more compact while maintaining a useful level of functionality. Arisumi et al. have leveraged these advantages to perform tasks ranging from precise placements of end effectors[5] to controlling projectile orientation during launch and flight[2, 4], and mitigating impact on the payload upon landing through multiple-projectile casting [3]. Similarly, Hatakeyama and Mochiyama have investigated the phenomena surrounding high-speed projectiles and altering their trajectories with passive dynamic components [11, 10].

The sections of this work are as follows. In chapter 2 we give the context of our paper through related works. Following in chapter 3 we present our materials and methods for wrapping with a tethered projectile. Chapter 4 presents the results of validation of our methods and chapter 5 analyzes the results. We end our paper with our conclusions in chapter 6.

Chapter 2: Literature Review

We seek to understand the fundamentals of whole-tether casting manipulation. Like whole-body manipulation[22], whole-tether casting manipulation seeks to use the tether, in addition to the projectile, to generate the desired interaction. Prior research in the area of casting manipulation has been focused on positioning the cast object to land at a particular location around the robot.[2, 5, 7].

The study of casting manipulators pioneered by Arisumi et al, has been focused on casting manipulator end placement. [2, 5, 4] To this end, they have developed techniques for orienting and positioning objects mid-flight through impulses applied to the tether. In particular, their work with projectile position and attitude control through the method of multiple braking [5, 4] has relevance to our present work.

To be able to adjust the attitude and position of a launched projectile, Arisumi developed the method of multiple braking control. Multiple braking is the act of applying a momentary braking force to the tether multiple times as the projectile is on its ballistic trajectory. The impulses applied through the tether are transmitted to the projectile. Appropriately timed impulses allow for the system to perform manoeuvres such as dropping the projectile in a desired location, or impaling a hollowed out ball on a rod as Arisumi et al. performed in their kendama experiments [4, 3].

In the spirit of whole-body manipulation[6, 22, 1], we seek to understand the dynamics of using the whole tether to interact with the environment, rather than just the end-effector as in [5, 2, 7]. Suzuki et al. have done some preliminary exploration into these tether dynamics, wrapping targets near the robot [23, 24, 25].

The goal of our effort is to the tethered projectile to wrap a distant object, with the goal of creating a secure connection to that object. In particular we examine the use of an end-weighted line to wrap a cylindrical target above and forward of the projectile launch location.

Positioning the end-mass portion of a tethered projectile only utilizes the projectile for its interactions and thus explores only a portion of the possible applications of a casting manipulator. Wrapping involves using the projectile's tether to wrap a distant object with the objective of creating a secure connection to that object. Wrapping the tether around a object, unlike end positioning, is more heavily influenced by the trajectory and velocity the projectile has through it's flight.

Throwing the projectile starts by accelerating the projectile to the desired release velocity in one of two ways. If the projectile can be pulled up tight against the end of the arm, then the arm simply pivots backward to a ready position, pauses and then accelerates forward to get the projectile to speed. If the projectile cannot be pulled up tight against the arm, then the arm goes through a swing up motion, as proposed by Izumi et al. [14]. This indirectly accelerates the projectile to the desired velocity in a controlled fashion. During this stage the tether is held at constant length so as to maximize the controllability of the action. Optimizing the rate of energy transferred to the projectile requires that the length of the tether

be dynamically changed during the swing-up process.[19]

2.1 Rebound Wrapping

In our previous work, we examined the conditions for generating an ideal wrap, in which the tethered projectile transitions with zero jerk from it's ballisitic trajectory to the inward spiraling wrapping trajectory. In this paper, we extend the tools we have already developed [12] and identify a second strategy for wrapping a target less sensitive to inputs. We call it rebound wrapping; as it uses the dynamics of the tether going taut to redirect the projectile onto a more desirable path, as illustrated in figure 3.4.

Rebound of a tethered projectile has much in common with the impact dynamics of a tethered, gravity-free, frictionless disc described by Faraji et al [9] and the hybrid dynamics of walking as described by Ramezani et al. [20]. In both works a semi-rigid collision is treated as an instantaneous event that changes the momentum of the system based on the spatial configuration and velocity of the system at the time of the event. However, neither work looks at tethered projectiles in a gravity field.

Impact dynamics have been used in casting manipulators before by Hatakeyama and Mochiyama to increase the time a launched tethered projectile spends at the extreme extent of its range [10]. Their inspiration came from the biological mechanism behind the chameleon's tongue. Their extremely simple mechanism, essentially a magnet tethered to a spring, was able to snag small ferrous objects

out of the air by allowing the magnet to fire, dwell near the area of the object, and then use the energy stored in the spring to return the magnet and object to the robot. Slightly less related to the field of impact dynamics, but still in the same area of leveraging simple dynamics for high-speed projectile tasks, Hatakeyama and Mochiyama showed that a tethered projectile could be precisely placed by winding the tether around a rotating drum and using the inertia of the drum to control the momentum of the projectile so that it dropped at the desired point [11]. We look to their work as inspiration for taking advantage complex dynamics using simple mechanisms.

Simply accounting for the rebound is not enough to help us make wraps more consistently. Being able to select points in the parameter space would enable the use of a less precision machine to perform the same task. Pekarovskiy et al examined the robust and consistent throwing of a simple projectile in a 2-dimensional low gravity environment using a shotput like motion with a simple arm [15, 17, 16]. By heavily parameterizing the trajectory of an arm using splines, they were able to relate the variations in a single parameter, the deceleration point of their arm, to variations in the final x-position of the projectile. We extend their work in this paper by creating method that allows for the analysis of sensitivity of a system by relating an arbitrarily sized set of input parameters to a single function that encompasses all desired aspects of the final state, not just a single position.

2.2 Visual Servoing & Other Feedback Methods

Very recent work by Ito et al [13] has looked at wrapping a distal target with a bullwhip mounted to a robotic arm. While similar to our work, their approach involved extreme instrumentation, placing dozens of visual markers along the length of the whip and using motion capture to estimate the configuration of the whip as the arm moves it through space. The use of motion capture has been a trend in casting manipulation from early on. Fagiolini and Arisumi's research groups have often used visual servoing (tracking the projectile) throughout much of their casting manipulation work to enable complex end-effector placement tasks [3, 8]. In particular, Arisumi's work in mid-air redirection via impulse application used a vision system to track the position of the projectile. Both in Arisumi's and Ito's setupsthe visual feedback came from outside the robot and even outside the workspace of the task. Pekarovskiy did work on ideal trajectories that could be deformed online using feedback to meet the requirements of a throw [18]. Our setup is not capable of such sensing needs and so requires a different approach that takes advantage of the dynamics of the robot and the geometry of the target we wish to interact with. We demonstrate the validity and robustness of our approach on an experimental robot.

Chapter 3: Materials and Methods

3.1 A Model of Wrapping

Our approach to produce a wrap with a tethered projectile is based on the model shown in figure 3.1, in which a casting manipulator with a mass at the tip, and negligible tether mass wraps around a circular target with radius r, while gravity, g, acts on the end-mass. A tension force F is applied via the anchored end of the tether, we refer to this as the proximal end as it is the portion of the tether that is closer to the robot throwing the projectile. This wrapping model admits slip between the tether and the target and makes the simplifying assumption that the tension T in the portion of the tether distal to the anchor remains positive for all time. Therefore this distal portion remains taut, forming a straight line between the perimeter of the target and the mass.

We take the two generalized coordinates for the system as θ, the counterclock-

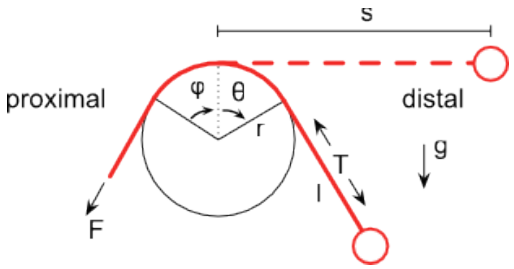


Figure 3.1: Model for the casting manipulator wrapping around its target.

wise angle from the top of the target to the departure-point of the distal portion of the tether, and s, the length of the distal portion of the tether when $\theta = 0$. Alternate choices of coordinates include the length of the distal portion of the tether, $d = s - r\theta$ and the θ angle, but several factors favor our chosen coordinates. Most importantly, s is differentially independent from θ, with the effect that $\dot{s}$ directly corresponds to the slip velocity of the tether around the target and that solutions with $\dot{s} = 0$ correspond to wrapping without slipping; in contrast, changes in d are produced by both wrapping and slipping. Similarly, measuring the wrapping angle from a fixed point on the target rather than the proximal contact point decouples the configuration of the system from this contact point; a given (s, θ) pair always describes the same physical configuration, independent of the (counterclockwise) proximal contact angle ϕ.

Applying standard Lagrangian techniques to the system in figure 3.1 produces its dynamic equations of motion,

$$\ddot{s} = (s - T\theta)\dot{\theta}^2 - T/m - g\sin\theta \tag{3.1}$$

$$\ddot{\theta} = (T\dot{\theta}^2 - 2\dot{s}\dot{\theta} - g\cos\theta)/(s - T\theta). \tag{3.2}$$

The components of $\ddot{s}$ correspond respectively to the mass's centripetal acceleration (with respect to the instantaneous distal contact point, rather than the target's center), the tension restraining the mass against this acceleration, and the action of gravity parallel to the tether. The angular acceleration, $\ddot{\theta}$, terms

reflect the increases in rotational speed that accompany the shortening of the distal length l during wrapping or negative slipping, together with the gravitational moment around the distal contact point.

In constructing 3.1 and 3.2, we use the internal distal tension T as the generalized force on the system, rather than the applied proximal tension F. This choice is driven by the potentially complex friction effects between the tether and the target (i.e. capstan effects); by defining our dynamic system's boundary at the distal contact point, we make it easy to insert friction models of various complexity.

3.1.1 Wrapping Without Slipping

For our purposes, we consider that once a wrap starts, we assume no slippage between the tether and the target. This is a reasonable assumption for 2 main reasons: 1) the tether is taut when the wrap starts due to the fix length of tether and fixed anchor point, s, cannot change. 2) Once one encirclement of the target happens, the magnification of frictional forces due to the capstan effect is large and no slippage occurs. This later reasoning lends itself to the idea that if you can produce one wrap, the projectile will continue to wrap until it has no tether left. This is due the decreasing radius of the spiral that causes the projectile to increase speed, while lowering the energy requirement of the projectile to reach the apex of subsequent spirals as the apex gets closer and closer to the target.

The simplest behavior corresponding to 3.1 and 3.2 is wrapping without slip-

ping, in which $\dot{s} = \ddot{s} = 0$ and 3.2 simplifies to

$$\ddot{\theta} = (T\dot{\theta}^2 - g\cos\theta)/(s - T\theta). \tag{3.3}$$

This behavior occurs either when the proximal end of the tether is fixed or for certain combinations of applied F and frictional effects between the tether and the target, mentioned earlier. In both cases, the net effect is that F becomes a holonomic constraint force.

In wrapping without slipping there are no controllable inputs after the tether has made contact with the target and the tether goes taut, so the success of a wrap is determined entirely by the state of the manipulator when it strikes the target. To identify the initial states corresponding to a wrap, we solve 3.2 for the distal tension T with $\ddot{s} = 0$ to find its value as a constraint force,

$$T_0 = m(v^2/d - g\sin\theta), \tag{3.4}$$

where $v = d\dot{\theta}$ is the linear velocity of the mass, directed orthogonally to the tether. If the initial conditions are such that T_0 remains positive for all time, then the manipulator wraps the target.

Stated another way, maintaining tension on the distal portion of the tether throughout the wrap causes the end of the manipulator to spiral into the target. If tension on the tether is lost at any point during the wrap, the tether becomes slack and the projectile follows a ballistic trajectory. This ballistic trajectory can result in a wrapping failure. To guarantee that the tether stays taut through the

entire spiraling trajectory, the projectile must possess an initial velocity greater than an energy-derived quantity driven by s,

$$\bar{v}^2(\theta = 0) > (3\bar{s} + (2 - 3\pi/2)), \tag{3.5}$$

where the initial velocity is nondimensionalized by gravity and the radius of the target, $\bar{v}^2 = v^2/(gr)$, and the initial tether length is nondimensionalized by just the radius of the target $\bar{s} = s/r$.

3.2 A Model of Throwing

We use a model of throwing, not launching, because we chose to use a simple arm for accelerating the projectile up to the appropriate speed at the appropriate release point. Our design intent was to have a simple, repeatable, single mechanism for both projectile acceleration and anchor positioning. The arm we built does this by feeding tether out the end of the arm that holds the projectile when it is thrown, thus we only need a single motor for control. This is opposed to a mechanism that launches the projectile through some linear acceleration and then uses another actuator to control the anchor point. Other issues we encountered with launchers is the need for a rapid deceleration for a projectile to launch, something that requires powerful actuators that are up to the task and can take the abuse of repeated hard accelerations. We prototyped a hand-actuated launcher capable of producing a wrap as seen in figure 3.2, that deteriorated as the accelerating

mechanism destroyed its hard-stop.

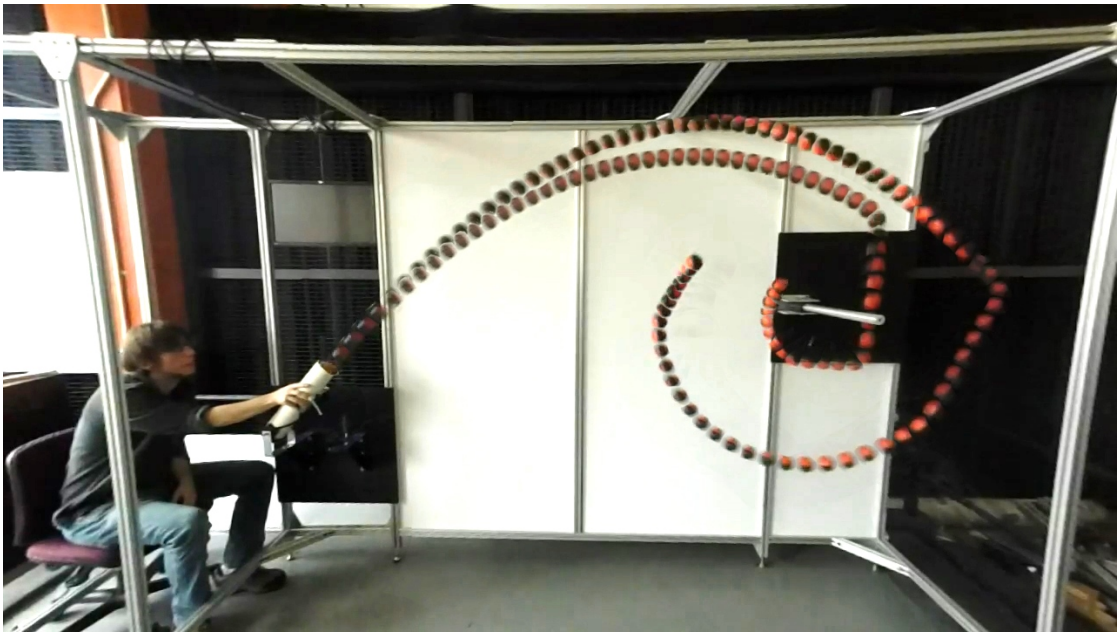


Figure 3.2: An early prototype launching device producing a wrap.

Similar to our model of wrapping, our throwing model assumes negligible air resistance and tether mass. Also carried over from the wrapping model is the coordinate system. For throwing, all 2D Cartesian coordinates are defined from the center of the target, and angles obey the right-hand rule, increasing in the counter-clockwise direction. The parameters that we use to define the ballistic trajectory are the xy coordinates of the center of the arm ($\mathbf{x}_{arm}$), the release angle (ϕ), and release velocity of the projectile ($\dot{\phi}$). We use these parameters because they are directly related to what variables we have control over. The parameters

are then used to created a set of initial conditions,

$$\mathbf{q}_0 = \begin{bmatrix} d\cos\phi \\ d\sin\phi \\ -d\dot{\phi}\sin\phi \\ d\dot{\phi}\cos\phi \end{bmatrix} + \begin{bmatrix} \mathbf{x}_{arm} \\ \mathbf{0} \end{bmatrix} \tag{3.6}$$

where d is the distance from the center of rotation of the arm to the center of mass of the projectile when held to the end of the arm. The trajectory for the given parameter set is then given by,

$$\dot{\mathbf{q}} = \begin{bmatrix} \mathrm{q}_3 \\ \mathrm{q}_4 \\ 0 \\ -g \end{bmatrix} \tag{3.7}$$

where g is the acceleration due to gravity and q_3 and q_4 are the respective x and y velocities of the projectile.

We define the limits of the ballistic trajectory of the projectile through the tether length, L, as the tether places a boundary condition on the projectile. One way to conceptualize this is that the fully extended tether describes a circle. When the projectile reaches this circle and the tether goes taut, the projectile acts as if it has hit a wall. As will be shown in Zero-Jerk wrapping §3.3, we look for throwing trajectories that avoid this event, and in Rebound wrapping, §3.5 we look for specific positions on the circle that produce the most controlled redirection of energy.

3.3 Zero-Jerk Wrapping: Smooth Transitions

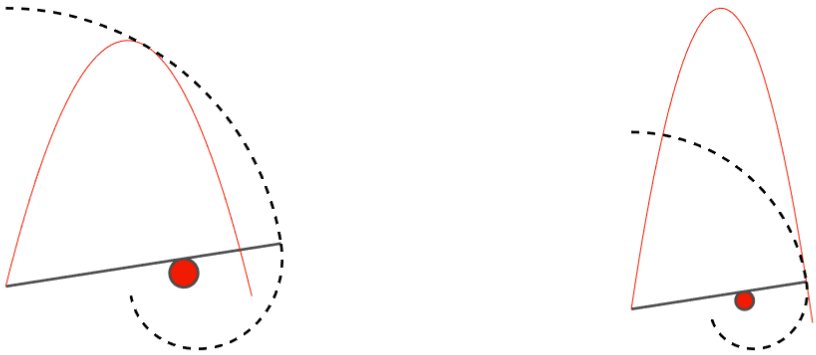
As mentioned in section §2.2, we see a need for the wrapping action to be performable without heavy sensing and rapid, online calculations for feedback control. To this end, of easily performing a wrap, our first idea that we developed we called Zero-Jerk wrapping. Zero-Jerk wrapping occurs when the intersection point between the ballistic trajectory of the thrown projectile and the tether boundary limit, (the circle and spiral that describe the full extension of the tether), meet at a point where the spatial 3rd derivative is zero and the 2nd derivative vectors are equivalent. Meeting these two conditions is equivalent to showing that the derivative of the curvature of the parabolic trajectory and the boundary limit are the same. At this point of transition the projectile experiences a smooth change in momentum, and all the energy it had in its ballistic trajectory is wholly transferred into the spiraling trajectory that wraps around the target. In other words,

$$\frac{d}{d\alpha}\left\|\frac{d\mathbf{T}_p}{d\alpha}\right\| = \frac{d}{d\alpha}\left\|\frac{d\mathbf{T}_b}{d\alpha}\right\| \tag{3.8}$$

where $d\alpha^2 = dx^2 + dy^2$ and $\mathbf{T}_p$ and $\mathbf{T}_b$ are the unit tangents to the curves that describe the ballistic parabolic trajectory and the boundary limit, respectively. Once a zero-jerk transition point is established, it is a simple task to determine if a wrap will succeed or not. If the projectile has enough energy when it transitions, satisfying 3.5, then it will wrap. We walked through this process backwards to get the necessary throwing parameters to achieve a zero-jerk wrap. More on this can be found in our previous work [12].

3.4 Shortcomings of Zero-Jerk Wrapping

What we did not realize when we developed Zero-Jerk wrapping however is that the set of parabolas that describe all possible ballistic trajectories that have 3rd-order smooth transitions to circles fall into one of two categories. The first category is the trajectories that obey the condition and stay inside the boundary just barely touch the circle, and then because no additional force acts on them, continue on the parabolic trajectory they had before, as seen in figure 3.3a. The second category that the rest of the parabolic trajectories fall into is the category of exiting the boundary limit which has larger practical implications. It indicates that for this throw to work, the tether of the projectile must be payed out so that the projectile can achieve it's ballistic trajectory, and then it must be retracted to the exact length at the exact moment when the transition from ballistic parabola to wrapping spiral is to occur. Figure 3.3b demonstrates this.



a Single Intersection b Double Intersection

Figure 3.3: Types of tangencies between parabolic trajectories and circular or spiral boundary limits.

3.5 Rebound Wrapping

Faced with the mechanical complexities of Zero-Jerk wrapping, we asked ourselves whether or not we should have avoided sudden changes in momentum. Our conclusion was that we should embrace the jerk and Rebound wrapping was born. The underpinnings of Rebound wrapping were that while the exact change in momentum the projectile undergoes when the tether goes taut is incredibly hard to predict, the range of changes in momentum is bounded. The two extremes are defined by perfectly inelastic and perfectly elastic collisions. The former results in a projectile with less velocity that it started with but all of its remaining velocity is tangential to the tether. In the latter, case the perfectly elastic collision allows the projectile to keep all of it's momentum, but a portion of it is reflected back in the direction of the tether as seen in figure 3.4. All other possibilities fall in this range.

Utilizing the rebound of the projectile is also in line with our goal of "easily" producing a wrap by avoiding the usage of high fidelity sensors with high bandwidth actuators. By taking advantage of the predictable nature of the rebound, we can be less precise with the exact trajectory we throw on. Enabling us to continue using our simple throwing robot. To improve the change of a wrap with Rebound wrapping we also perform a sensitivity analysis to try to find combinations of parameters that have a lower sensitivity to variation section §3.6. The inclusion of the sensitivity analysis we decided upon after finding that our machine's mechanism for releasing the projectile was even less precise than anticipated when working on

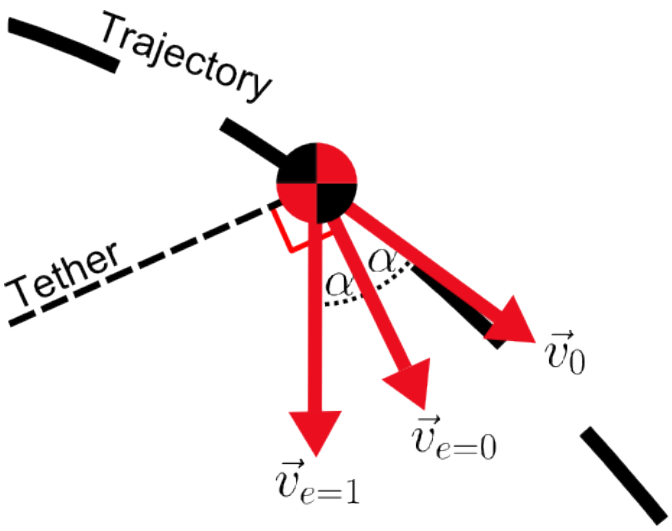


Figure 3.4: Cartoon of the range of changes in momentum a projectile can achieve at the moment of a rebound, $e = 0$ is a perfectly inelastic rebound and $e = 1$ is perfectly elastic.

Zero-Jerk wrapping.

3.5.1 Split Model: Rebounds in Throwing and Spiral Limits

Modeling rebound wrapping required an extension of the wrapping and throwing models described in sections §3.1 and §3.2. We use a dual parameterization of the position of the projectile relative to the target using both Cartesian and polar coordinates. Cartesian coordinates we used for ease of describing the change in trajectory of the projectile during a rebound, as well as ease of forward simulation of the ballistic trajectory of the projectile, both during the throw and wrapping phase. Polar coordinates we use to define the boundaries on the projectile created by the tether going taut.

In Zero-Jerk wrapping all calculations about the projectile could be done by considering the geometry of the parabolic throw and the boundary limits and so their models could be easily combined into one. With the modeling of the dynamics of the rebound and the incorporation of polar coordinates came the necessity of keeping the models separate. Fundamentally, this is because of the maps from the Cartesian space of the projectile to the polar coordinates is different for wrapping and throwing. In wrapping we consider that the angular portion of the polar coordinates of the projectile to have a winding number, indicating the number of times the projectile has gone around the target and thus what layer of the spiral it is in. In the throwing model, the polar position of a projectile needs to be able to specify whether the projectile is in the area where it will encounter the circular boundary limit of the tether or the spiral boundary limit of the tether. The polar coordinates in the throw are mapped about a line that extends from the anchor point to the upper edge of the target and has it's angle defined by,

$$\theta_0 = \arctan 2(-y_{anchor}, -x_{anchor}) + \arcsin(\frac{r}{\sqrt{x_{anchor}^2 + y_{anchor}^2}}) \tag{3.9}$$

Above the line we assume we are in the circular boundary limit, below the line, we assume inside the spiral, as seen in figure 3.5. What makes this a natural division line is that the projectile starts on the line and if it ever dips below it on its trajectory before its x coordinate becomes greater than zero, the wrap fails because the projectile never makes it over the target.

In either model, rebounds are treated the same way. We assume no external

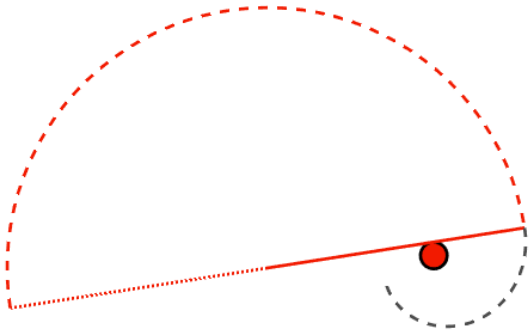


Figure 3.5: Mapping zones from Cartesian to poloar. Points in the red (top) zone map to the circular boundary limit. Points in the gray (bottom) zone map to the spiral boundary limit.

forces on the system, so the projectile's velocity immediately after rebound is the same velocity it will possess on its subsequent trajectory. Using the notation in figure 3.4, where $\vec{v}_0$ is the projectile velocity vector prior to rebound, and $\vec{v}_{e=1}$ is the rebound velocity vector immediately after a perfectly elastic rebound, we can calculate $\vec{v}_{e=1}$ using $\vec{v}_0$ by reflecting it over the tangent of the tether, $\mathbf{n}_T$,

$$\vec{v}_{e=1} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = -\vec{v}_0 + 2(\mathbf{n}_T \cdot \vec{v}_0)\mathbf{n}_T. \tag{3.10}$$

A perfectly inelastic rebound has the more simple calculation of,

$$\vec{v}_{e=0} = (\mathbf{n}_T \cdot \vec{v}_0)\mathbf{n}_T. \tag{3.11}$$

Calculating a rebound that is partially elastic is slightly more involved, as we must account for the dynamic parameters of the system, and cannot rely solely on geometry. We start by finding the projectile's velocity vector immediately after rebound, first finding the x, y coordinates of the point of rebound. We plug them in to the rebound velocity vector equation

$$\vec{v}_e = -e\mathbf{n}_N(\vec{v}_0 \cdot \mathbf{n}_N) + \mathbf{n}_T(\vec{v}_0 \cdot \mathbf{n}_T) \tag{3.12}$$

where e is the coefficient of restitution and $\vec{v}_0$ is the velocity of the projectile at the moment of rebound. Both $\mathbf{n}_N$ and $\mathbf{n}_T$ are normalized vectors that are respectively normal

$$\mathbf{n}_N = \frac{1}{\sqrt{x^2 + y^2}} \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.13}$$

and tangential

$$\mathbf{n}_T = \frac{1}{\sqrt{x^2 + y^2}} \begin{bmatrix} y \\ -x \end{bmatrix} \tag{3.14}$$

to the tether at the point it goes taut.

Transition from the throwing model to the wrapping model relies on the detection of when the projectile has rebounded inside the set of polar coordinates associated with the spiral. Upon detection, the throwing simulation is halted, the condition of the projectile after rebound extracted and then used as the initial conditions of the simulation of the wrapping model. The end condition for the wrapping model and detection of whether it was a success or failure look at the

energy the projectile possess after is rebounds with an angle shallow enough to be considered in orbit around the target. If the projectile never achieves this angle or doesn't have enough energy, the wrap was not successful. The angle we've chosen in our work is roughly 2 degrees.

Our specific implementation of throwing model further differs in that it also has maximum number of rebounds it is allowed to execute before exiting with an error if it doesn't have a rebound inside the spiral portion first. The spiral model simulation is allowed to rebound indefinitely until it encounters a failure, described below.

3.5.2 Failure Modes for Wrapping

Inside our simulations we have two main condition checks to see if either the throw or wrap simulation will fail. The first condition is direction of the projectiles velocity in polar coordinates. If the projectile's angle coordinate velocity is ever positive $\nu > 0$ the projectile cannot produce a wrap as the projectile is either unwinding in the wrapping case or in the throwing case it is on a trajectory that will fail to pass over the target to be wrapped. This can also look like the projectile bouncing back and forth inside the spiral. This is usually more of a function of energy however, as has been alluded to many times and defined by 3.5.

3.6 Sensitivity & Goodness: The Most Likely Parameters for a Wrap

Our primary innovation that allows us to consistently wrap using an inconsistent robot is the sensitivity analysis we do on a goodness function that has been mapped to the parameters that set up a throw. Sensitivity analysis to find robust parameters for our system is a difficult task on the surface as our measure of success is binary. Either the wrap succeeds or not. To make the analysis more tractable we need a measure of the goodness of a trajectory that has continuous variation, and ideally has diminishing returns the further a parameter set gets from the minimum goodness. For our purposes we also want a function that is positive at all points, has a bounded minimum and maximum, and when the bare minimum conditions for a wrap are met, returns 1.

Due to the simulated nature of our analysis, our goodness function does not directly operate on the parameters of the throw, $\Omega = [\mathbf{x}_{arm}, \phi, \dot{\phi}, e, L]$ (respectively, arm center in x and y, arm release angle, arm angular velocity, coefficient of restitution, tether length), but on features of the trajectory that results from them. Specifically the velocity of the projectile on its final rebound, which we refer to from now on as $\gamma(\Omega)$. Thus our function of goodness is $f(\gamma(\Omega))$.

The velocity features is actually a minimum energy condition that we have alluded to multiple times throughout this paper and is defined by 3.5. In the following we more rigourously define it and put it in terms of our specific model. Given the length of the tether extending from the contact point on the target to

the projectile,

$$s(\theta) = s_0 + (\theta - \theta_0)r \tag{3.15}$$

where θ is the angle of the tether and r is the radius of the target, we can calculate the height of the projectile,

$$y = r\cos(\theta) + s(\theta)\sin(\theta) \tag{3.16}$$

Differentiating by θ and we find that the max height is at $-3\pi/2$, note that $s(-3\pi/2) = h_{max}$. To then find the energy necessary for wrapping, we look at the height and velocity it has apex. First, the velocity necessary to negate gravity:

$$\begin{aligned} T = F_{centrifugal} &= F_{gravity} \\ \tfrac{mv_{min}^2}{h_{max}} &= mg \\ v_{min}^2 &= s(\tfrac{-3\pi}{2})g \end{aligned} \tag{3.17}$$

where v_{min} is the minimum velocity of the projectile, assuming all velocity is tangential to the spiral. If 3.17 holds true, then the energy contained at this point, relative to the 0,0 coordinate is:

$$E_{min} = \frac{1}{2}mv_{min}^2 + mgh_{max} \tag{3.18}$$

The current energy in the projectile then has to be equal to or greater than this amount:

$$E_{min} \leq E_{current} \tag{3.19}$$

$$\frac{1}{2}v_{min}^2 + gh_{max} \leq 1/2v_{curr}^2 + gy \tag{3.20}$$

Which becomes

$$s(-3\pi/2)g + 2g(h_{max} - y) \leq v_{curr}^2 \tag{3.21}$$

or

$$g(3h_{max} - 2y) \leq v_{curr}^2. \tag{3.22}$$

To meet our criteria for the goodness function, we then bound to the values to $[0, 2]$ to get our goodness function:

$$f = \frac{2}{\pi}\arctan\Big(\dot{\mathrm{q}} \cdot \dot{\mathrm{q}} - g(3h_{max} - 2y)\Big) + 1. \tag{3.23}$$

3.7 The Flatness of Goodness

To find the least sensitive, and thus optimal parameter point Ω^* on the manifold of $f(\gamma(\Omega))$ that is as far away from $f = 1$ as possible, we take a point in the goodness manifold, parameterized by initial conditions and assume it has second order characteristics (hyper-parabolas). We then look for what parameter sets this linearization crosses goodness = 1. The smallest distance between the point we start at and these set of crossing points

Our second-order linearization is done with the Jacobian and Hessian of the goodness function (taken with respect to the parameters). We then find its roots and then find the root closest to the linearization point. Our first step to finding this point is creating a second-order linearization around of f, via the Taylor series

expansion,

$$\hat{f}(\Omega) = f(\gamma(\Omega_0)) + \mathrm{J}_{\Omega_0}(\Omega - \Omega_0) + (\Omega - \Omega_0)^T \mathrm{H}_{\Omega_0}(\Omega - \Omega_0), \tag{3.24}$$

where the Jacobian is $\partial f/\partial\Omega$ which in turn has a Hessian given by $\partial^2 f/\partial\Omega_i\partial\Omega_j$, all of these are defined a on point-by-point basis.

If all of the eigenvalues of the Hessian of the linearization given by 3.24 are positive $\partial^2\hat{f}/\partial\Omega_i\partial\Omega_j = 2H_{\Omega_0}$, the goodness is greater than 1, and the Jacobian, $\partial\hat{f}/\partial\Omega = J_{\Omega_0} + 2(\Omega - \Omega_0)H_{\Omega_0}$, is a zero-vector, then this point is awesome because locally the goodness can only go up from here! Parameter points that do not fit these criteria require further analysis and computation.

To calculate how "far" away we are from nearest $f(\gamma(\Omega)) = 1$ crossing we apply a constrained optimization technique on the linearization $\hat{f}$. With the optimization we want Ω^* to be as close to the point of linearization Ω_0 as possible, while still respects $\hat{f} = 1$. We seed this optimization with the non-optimal crossings given by 1-dimensional projections of the linearization. The axes of projection are given by the eigenvectors of the Hessian $2\mathrm{H}_{\Omega_0}$.

Generation of the projection involves taking an eigenvector of $2\mathrm{H}_{\Omega_0}$, $\mathbf{v}$ and using it as a projection vector to project Ω and $\partial\hat{f}/\partial\Omega$ into a one-dimensional space we call z. We use z to define a new, 1-dimensional goodness function $\bar{f}(z)$ that we make a couple big assumptions about, the first is that it's curvature is given by $\partial^2 f/\partial z^2 = \lambda$, the eigenvalue corresponding to our eigenvector. From that we derive the following equation for behavior of the manifold around a parameter

point $\Omega_0 \cdot \mathbf{v} = z_0$ along an axis given by eigenvector:

$$\bar{f}(z) = \frac{\lambda}{2} z^2 + bz + c \tag{3.25}$$

The value of b is given by the evolution,

$$\left.\frac{\partial \bar{f}}{\partial z}\right|_{z_0} = \lambda z_0 + b. \tag{3.26}$$

Substituting in the projection of the Jacobian along the eigen vector for the partial derivative of $\bar{f}$ we get,

$$b = \left.\frac{\partial \hat{f}}{\partial \Omega}\right|_{\Omega_0} \cdot \mathbf{v} - \lambda z_0 \tag{3.27}$$

Calculating c is as follows:

$$\bar{f}(z_0) = \frac{\lambda}{2} z_0^2 + bz_0 + c \tag{3.28}$$

Which is then reformulated as,

$$c = \bar{f}(z_0) - \frac{\lambda}{2} z_0^2 - \Big(\left.\frac{\partial \hat{f}}{\partial \Omega}\right|_{\Omega_0} \cdot \mathbf{v} - \lambda z_0 \Big) z_0. \tag{3.29}$$

Running this through a quadratic equation to find the z values that correspond with a crossing of the line $\hat{f} = 1$,

$$z_{cross} = \frac{-b \pm \sqrt{b^2 - 2\lambda(c-1)}}{\lambda} \tag{3.30}$$

Once we have z_{cross} in hand, we project it back into the parameter space to get $\Omega_{cross} = z_{cross}\mathbf{v}$ which we use for the following optimization problem:

$$\Omega^* = \min_{\Omega}(\Omega - \Omega_0) \cdot (\Omega - \Omega_0) \tag{3.31}$$

subject to,

$$1 = \hat{f}(\Omega). \tag{3.32}$$

From this we generate what we refer to as the score,

$$k = \left(\Omega^* - \Omega_0\right) \cdot \left(\Omega^* - \Omega_0\right) \tag{3.33}$$

This means that a score can be used to indicate the maximum guaranteed deviation a set of parameters can have and still be guaranteed to wrap. It is possible for deviations that result fall outside of the "ball" with a radius $\sqrt{k}$ to still produce successful wraps, but it is not guaranteed.

3.8 Practical Implementation

Finding the optimal parameter set with the largest possible score, k, we use the optimization technique of simulated annealing [21]. Simulated annealing was chosen over traditional techniques like gradient descent or the built-in optimizer found in MATLAB because of the nature of the gradient of f and how we calculated it. Because we can only calculate $\gamma(\Omega)$ via simulation calculation, our gradient

calculation of $\partial f / \partial \Omega$ is also simulation dependent. In addition f is not a smooth function with a single global maximum, much less the score k. Simulated annealing works by perturbing the parameter set randomly to see if can move to a better set, occasionally moving to a worse set to ensure that it doesn't get stuck in a local optima. Thus it does not need a function to be smooth around a set so long as it can product a gradient estimate.

3.8.1 The Casting Manipulator Testbed

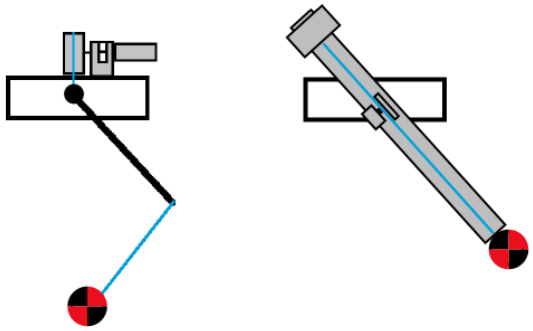


Figure 3.6: Pictured here are the front-view caricatures of the BAM Mk.I and the BAM Mk.II system.

Our casting manipulator robot — the Ballistic Arm Manipulator (BAM) — is composed of a throwing arm and tethered projectile. BAM Mk.I controlled the tether length and release through an active spool, designed to be able to reel in the tether during the projectile's flight, and to achieve the zero-jerk wrapping approach discussed earlier and presented in [12].

Our second generation, BAM Mk.II, abandoned the active spool in favor of a

more responsive passive spool and brake system. The Mk.I spool had an average "release" time of 50ms, while the Mk.II braking system is approximately 10ms. In addition the new BAM has a centralized design in that all components, but the motor rotating the arm, are integrated into the arm. We did this to eliminate routing the tether through multiple bends, reducing the friction in the system. The other major design change was a "fork" at the end of arm. enabling the projectile to be held rigidly to the end of the arm when accelerated. This modification eliminated the need to perform a swing-up action to get the projectile to the desired pre-throw state. Figure 3.6 illustrates the difference in design between the two generations of BAM.

Dynamic measurements from both Mk.I and Mk.II indicate that the tether, a braided fishing line, and bean-bag projectile are both dynamically dead, with a coefficient of restitution close to zero. However, the anchoring components and the arm itself add non-negligible amounts of damping and elasticity, which dominate the tethered projectile's impact response. This semi-elastic impact response when the tether goes taut plays an important role in our new, rebound-based method of wrapping a target.

Chapter 4: Results

Results are broken down into two sections. Our first section, Simulation Results §4.1, reports on the findings of our optimization to find a parameter set that maximizes robustness. Experimental Results §4.2 applies our technique to data from prior experiments to validate our methods.

4.1 Simulation Results

Performing our optimization on a set of parameters seeded close to our actual system's throwing parameters, and letting it run for 1000 iterations resulted in the parameters given in table 4.1, with a score of 0.447. A massive improvement over the initial parameter's score of 1.36×10^{-3}. Figure 4.1 shows what this optimal throw looks like. It should be noted that while there are 6 parameters over which the optimization scores a particular point, only the release angle, release velocity and tether length were permitted to be changed. This restriction is because of the nature of the robot. The center position of the robot's arm is difficult to move and the coefficient of restitution the system is not changeable.

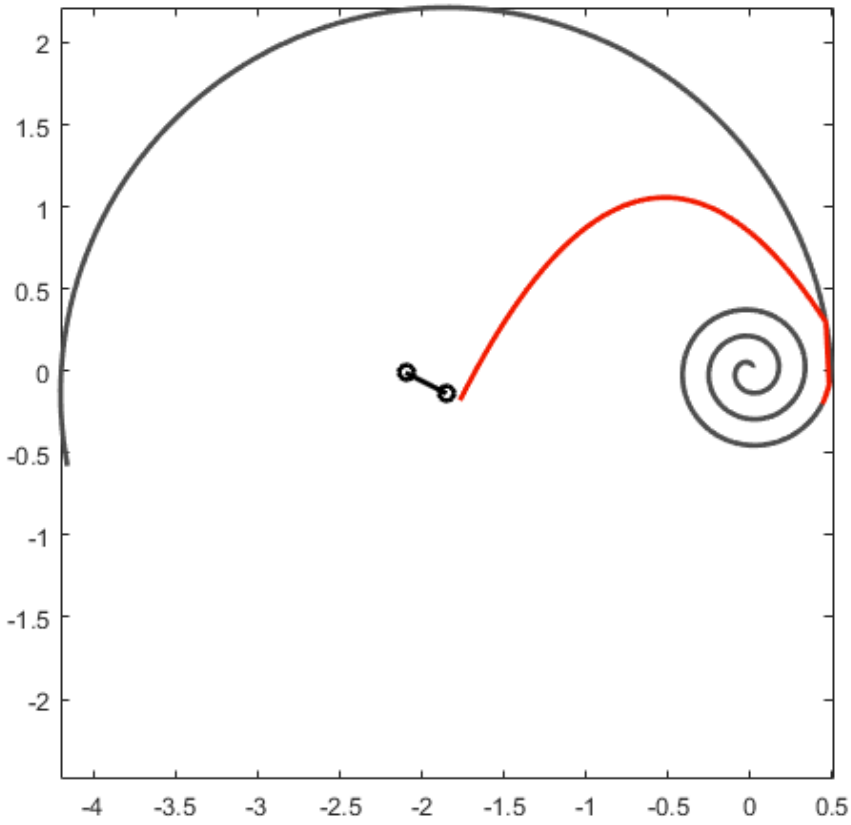


Figure 4.1: The throw with the greatest degree of robustness to variations in parameters for our given physical configuration of the casting manipulator.

Table 4.1: The optimal parameters for a throw constrained by our physical robot

Arm center x	-2.1 m
Arm center y	-0.01 m
Release angle ϕ	-0.468 rad
Release velocity $\dot{\phi}$	14.85 rad/sec
Coefficient of Restitution e	0.32
Tether Length L	2.35 m

4.2 Experimental Results

In prior work, through trial and error, we found a set of throwing parameters that allowed us to demonstrate a wrap was still possible with variations in the length of tether, if we could leverage the rebound appropriately. Taking this data, we backwards processed it to derive the actual trajectory we threw the projectile on. Experimental determination of the coefficient of restitution for a rebound placed the value around $e = 0.32$. These parameters and other are given in table 4.2 and the success rates for the varying tether lengths given in table 4.3. An ANOVA Single Factor test performed on the percentage of successful wraps is inconclusive about the difference in success between the trials. To get more conclusive results ($p - value < 0.05$) would require roughly 50 throwing attempts per trial.

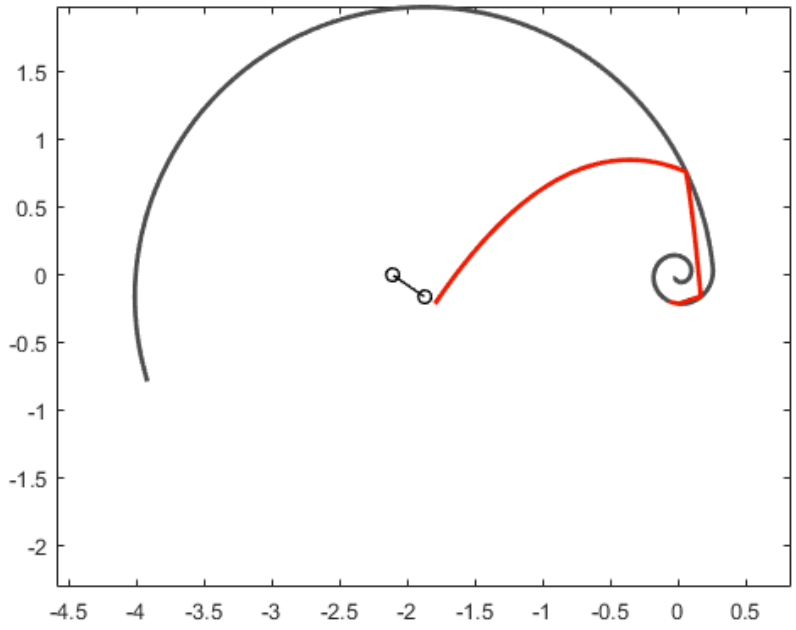
Figures 4.2, 4.3, and 4.4 shows the comparison of the simulated and the actual trajectory. Of the 30 throws in total performed, there was significant variance in the release angle and release velocity. The standard deviation on the release angle was $0.0644 rad$ and the velocity, $0.279 rad/sec$.

Table 4.2: Table of universal parameters for our throw.

Arm center x	-2.1m
Arm center y	-0.01m
Release angle ϕ	-0.6134 rad
Release velocity $\dot{\phi}$	14.70 rad/sec
Coefficient of Restitution e	0.32

Table 4.3: Results from the throws, n=10 throws per tether length

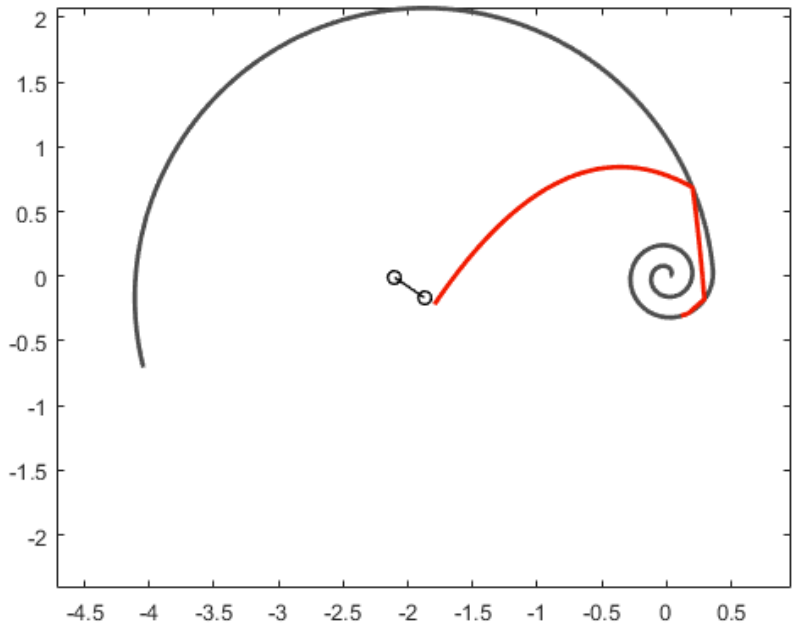
Trial	Tether Length	Goodness	Score	Successful Wraps
1	2.15m	1.10	1.81×10^{-6}	70%
2	2.24m	1.83	1.36×10^{-3}	50%
3	2.30m	1.86	2.87×10^{-3}	70%



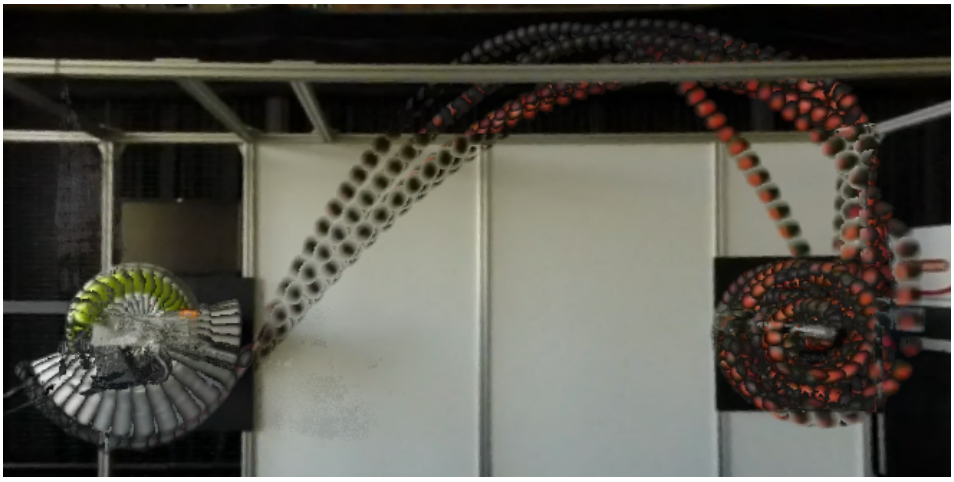
a Simulation

b Experiment

Figure 4.2: Trial 1: Tether length $L = 2.15$ meters

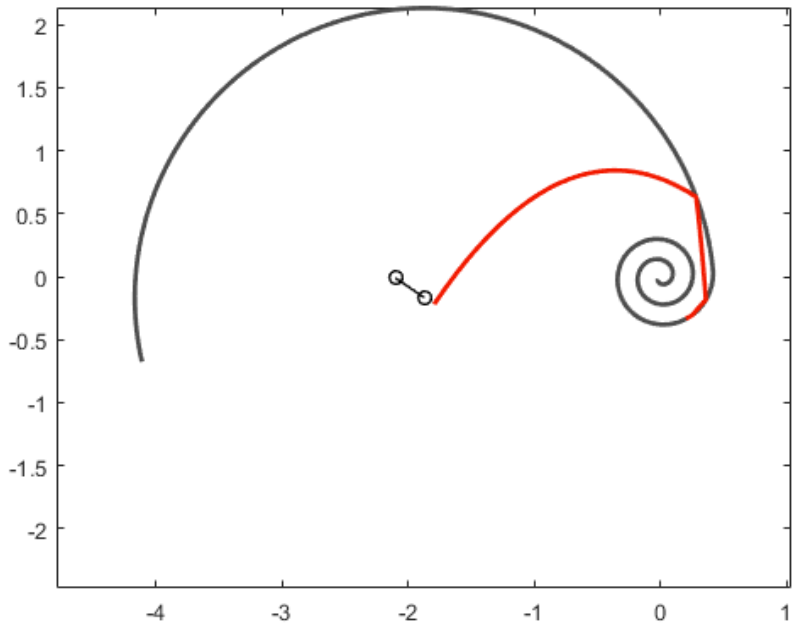


a Simulation

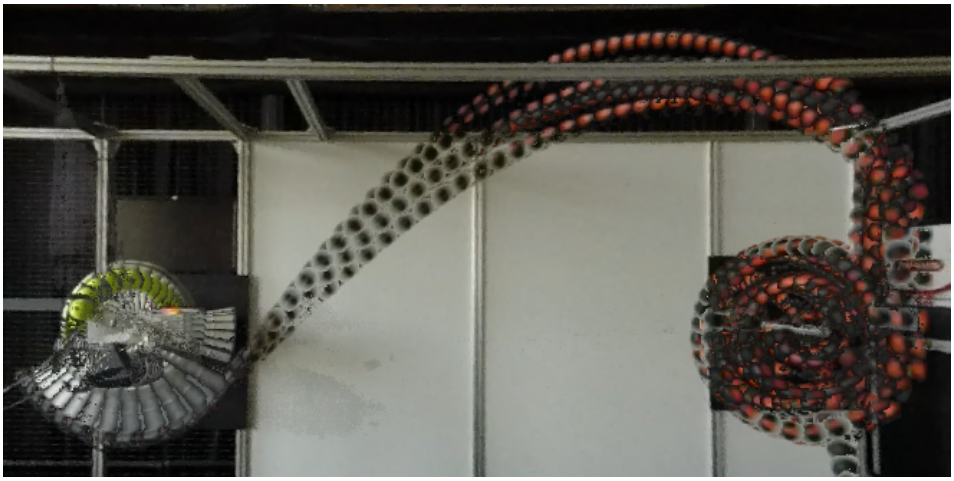


b Experiment

Figure 4.3: Trial 2: Tether length $L = 2.24$ meters



a Simulation



b Experiment

Figure 4.4: Trial 3: Tether length $L = 2.30$ meters

Chapter 5: Discussion

We realized that the task we were seeking to perform was at the limits of our robot's capabilities, in power, accuracy, and sensor feedback, so we evolved our methods from Zero-Jerk wrapping to Rebound wrapping.

5.1 Analysis of the Simulation

We observed peculiar behavior in parameter selection. When running the trajectory optimization with no constraints on the parameters, (except for bounding the coefficient of restitution to the domain $[0, 1]$), the optimization found highly scored parameter sets tended to position the center of the arm above the target, throw in horizontally flat trajectories, and select coefficients of restitution as close to zero as possible. Casual analysis agrees seems to support this phenomena logically. By placing the arm higher, there is more potential energy to convert into velocity for a wrap and trajectories that start horizontally have less variance in final position. Also, a coefficient of restitution close to zero means that the projectile only has to rebound inside the spiral with a minimal amount of energy and it is guaranteed to wrap.

Constrained optimization tells an alternate, albeit, interesting story. Letting the optimization occur across single parameters at a time and using the trajectory

parameters specified in 4.2 and tether length $L = 2.24m$ as a basis of comparison we found the following, (results given in table 5.1). By themselves, release angle, release velocity, and the coefficient of restitution had the most dramatic improvements in score as they increased it by several orders of magnitude over the base score of score of 1.36×10^{-3}. Intuitively, these findings make sense as the release angle and velocity specify the major features of the throwing trajectory, and as discussed before, the closer the coefficient of restitution is to zero, the more robust a given trajectory is to variation. At the opposite end of the spectrum, changes in the the x-coordinate of the arm center had no visible improvement on the score for the base set of parameters, implying that the x-coordinate of the arm, under small perturbations, has little effect. In the middle of the spectrum sit the y-coordinate of the arm center and the tether length. As mentioned previously, the higher the arm is above the target, the more energy the projectile will have, thus more likely to wrap and a higher score. Tether length has a harder to discern effect on the score as it primarily changes the contact point of the projectile, and longer tether lengths result in more energy to wrap as the projectile is on a wider orbit around the target.

5.2 Analysis of Experimental Data

Data from the experiment did show an agreement between the goodness function and a successful wrap being possible. However, the same data was inconclusive about the relationship between the scores of those locations and the likelihood of

Table 5.1: Simulation across individual parameters

Parameter	Value	Goodness	Score
Arm center x	-2.103 m	1.82	1.36×10^{-3}
Arm center y	0.240 m	1.92	5.00×10^{-3}
Release angle ϕ	-0.468 rad	1.96	0.0392
Release velocity $\dot{\phi}$	13.44 rad/sec	1.91	0.0587
Coefficient of Restitution e	0.0773	1.95	0.470
Tether Length L	2.29 m	1.85	2.02×10^{-3}

a successful wrap. Further analysis used the scores of the three trials to create bounding "balls" of radius $\sqrt{k}$ for the parameters. A parameter set inside the ball is guaranteed to wrap. Combining this with the mean and standard deviation of the throwing trajectories reveals that only a small percentage of the throws where in the guaranteed "ball". Table 5.2 shows these results. The extremely low probabilities found in table 5.2 suggest that the guarantees provided by the scores are extremely conservative when it comes to estimating the acceptable area around a parameter point.

Table 5.2: Probability that for the distribution of release angle $\mathbb{E}(\phi) = \mathbb{N}(-0.6134, 0.0644)$ and release velocity $\mathbb{E}(\dot{\phi}) = \mathbb{N}(14.70, 0.2791)$, a wrap can be guaranteed.

Trial	Score	Probability
1	1.81×10^{-6}	$< 0.006\%$
2	1.36×10^{-3}	3.6%
3	2.87×10^{-3}	7.3%

5.3 The Difficulties of Wrapping

To put in perspective on these results, consider that there are many more ways for a tether to fail to wrap an object than there are ways to successfully wrap. Most common of the reasons for failure is lack projectile energy, as has been alluded multiple times through out this paper and defined in equations 3.5 and 3.22. It could be thrown with insufficient energy to even reach the target, it could lose too much energy after a series of rebounds repeatedly remove energy for the system, or it could have one particular harsh rebound when the velocity of the projectile is orthogonal to the boundary limit and on rebound loses most of its energy.

Other causes of failure could be that a rebound causes the projectile to move counter-clockwise around the target. Either unwinding a wrap inside the spiral or causing the projectile to rebound left of the target missing it all together. The projectile could impact the target and cause a failure. Something that was not obvious in our models or initial analysis of the problem was that the projectile could also encounter its own tether while wrapping and that could cause a wrap to fail.

All of these failures are can be planned for in using Zero-Jerk or Rebound wrapping. What drove us to incorporate a sensitivity analysis were the failures caused by the inconsistencies in the robot itself. Inconsistencies like inability to release at same release angle, possibly the largest issue due to the high speed nature of wrapping. A small variation in angle at the speed we throw can result in a large change in the parabolic trajectory and thus a change in success or fail.

Another robot inconsistency that did not have as much of an immediate effect but should be accounted for in a higher fidelity model the angle tether makes with the arm during a rebound. Some angles apply more torque to the arm and that changes how the arm factors into the coefficient of restitution. Sensitivity analysis allows us to intelligently make choices on how to best mitigate hard to control inconsistencies and perform a successful wrap.

Chapter 6: Conclusion

Through this paper we have presented a method for wrapping a target with tether projectile without feedback control over the projectile, but by careful selection of throwing parameters. Our method leveraged the tethered projectile's ability to rebound. Through the use of a sensitivity analysis we demonstrated how to select throwing parameters robust to variation. Our innovation with the sensitivity analysis was to create a goodness function as a proxy for condition on a binary system: successful or unsuccessful. Using an annealing method, we find optimally robust trajectories. We validated our method through simulation and data from a physical throwing robot, the BAM. The data supported our method of rebound wrapping and the formulation of our goodness function, but was inconclusive on the effectiveness of our method for scoring the robustness of a throw.

Our next step in this process of further developing general methods of wrapping with casting manipulators calls for the exploration of more diverse tether physics, more tightly defining the physical phenomena around projectile rebound, examining the effects of compliant throwing arms and tether anchors. Of most interest to us however is the effect of having increased tether mass to projectile mass ratio, bringing the system dynamics closer to those seen in whips and similar tools, an area almost completely unexplored at this time.

Bibliography

[1] H. Arisumi, J.-R. Chardonnet, and K. Yokoi. Whole-body motion of a humanoid robot for passing through a door - opening a door by impulsive force -. In *IEEE/RSJ IROS*, 2009.

[2] Hitoshi Arisumi, Masatsugu Otsuki, and Shinichiro Nishida. Launching penetrator by casting manipulator system. *IEEE International Conference on Intelligent Robots and Systems*, pages 5052–5058, 2012.

[3] Hitoshi Arisumi, Masatsugu Otsuki, and Shinichiro Nishida. Soft Landing of Capsule by Casting Manipulator System. Chicago, IL, sep 2014.

[4] Hitoshi Arisumi, Kazuhito Yokoi, and Kiyoshi Komoriya. Kendama game by casting manipulator. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 138–145, 2005.

[5] Hitoshi Arisumi, Kazuhito Yokoi, and Kiyoshi Komoriya. Casting Manipulation Midair Control of a Gripper by Impulsive Force. *IEEE Transactions on Robotics*, 24(2):402–415, 2008.

[6] A Dietrich, T. Wimbock, A Albu-Schaffer, and G. Hirzinger. Reactive Whole-Body Control: Dynamic Mobile Manipulation Using a Large Number of Actuated Degrees of Freedom. *IEEE Robotics Automation Magazine*, 19(2):20–33, jun 2012.

[7] A. Fagiolini, F.A.W. Belo, M.G. Catalano, F. Bonomo, S. Alicino, and A. Bicchi. Design and control of a novel 3D casting manipulator. pages 4169–4174, may 2010.

[8] Adriano Fagiolini, Hitoshi Arisumi, and Antonio Bicchi. Visual-based feedback control of Casting Manipulation. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2005, pages 2191–2196, apr 2005.

[9] Hossein Faraji, Stephanie Veile, Samantha Hemleben, Pavel Zaytsev, Joel Wright, Hans Luchsinger, and Ross L Hatton. DSCC2015-9708. In *ASME 2015 Dynamic Systems and Control Conference*, 2015.

[10] T. Hatakeyama and H. Mochiyama. Shooting manipulation system with high reaching accuracy. pages 4652–4657, sep 2011.

[11] T. Hatakeyama and H. Mochiyama. Shooting Manipulation Inspired by Chameleon. *IEEE/ASME Transactions on Mechatronics*, 18(2):527–535, apr 2013.

[12] L. Hill, T. Woodward, H. Arisumi, and R. L. Hatton. Wrapping a target with a tethered projectile. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1442–1447, May 2015.

[13] Koichiro Ito, Yuji Yamakawa, and Masatoshi Ishikawa. Winding Manipulator Based on High-speed Visual Feedback Control. 2017.

[14] Shinsuke Izumi, Yukio Takeda, Masaru Higuchi, Hiroaki Funabashi, and Hitoshi Arisumi. Motion Generation of a Casting Manipulator With a Large Working Space. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 489–495, jan 2002.

[15] Masafumi Okada, Alexander Pekarovskiy, and Martin Buss. Robust Trajectory Design for Object Throwing based on Sensitivity for Model Uncertainties. *International Conference on Robotics and Automation*, (c):3089–3094, 2015.

[16] Alexander Pekarovskiy and Martin Buss. Optimal control goal manifolds for planar nonprehensile throwing. In *IROS*, 2013.

[17] Alexander Pekarovskiy, Thomas Nierhoff, Sandra Hirche, and Martin Buss. Dynamically Consistent Online Adaptation of Fast Motions for Robotic Manipulators. *IEEE Transactions on Robotics*, 34(1):166–182, 2018.

[18] Alexander Pekarovskiy, Thomas Nierhoff, Jochen Schenek, Yoshihiko Nakamura, Sandra Hirche, and Martin Buss. Online deformation of optimal trajectories for constrained nonprehensile manipulation. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2015-June, pages 2481–2487, 2015.

[19] Auke A. Post, Gert de Groot, Andreas Daffertshofer, and Peter J. Beek. Pumping a playground swing. *Human Kinetics Journals, Motor Control*, 11(2):136–150, apr 2007.

[20] Alireza Ramezani, Jonathan W. Hurst, Kaveh Akbari Hamed, and J. W. Grizzle. Performance Analysis and Feedback Control of ATRIAS, A Three-Dimensional Bipedal Robot. *Journal of Dynamic Systems, Measurement, and Control*, 136(2):021012, 2014.

[21] J.S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. 2003.

[22] M. Stilman, K. Nishiwaki, and S. Kagami. Learning object models for whole body manipulation. pages 174–179, nov 2007.

[23] T. Suzuki and Y. Ebihara. Casting Control for Hyper-Flexible Manipulation. pages 1369–1374, apr 2007.

[24] T. Suzuki, K. Shintani, and H. Mochiyama. Control methods of hyper-flexible manipulators using their dynamical features. volume 3, pages 1511–1516 vol.3, aug 2002.

[25] Takahiro Suzuki, Yuji Ebihara, Yoshinobu Ando, and Makoto Mizukawa. Casting and Winding Manipulation with Hyper-Flexible Manipulator. In *IEEE/RSJ IROS*, pages 1674–1679, oct 2006.