In-Hand Manipulation Planning with Underactuated Robotic Hands

A Thesis presented

by

Eric Tola

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Master of Science

in

Mechanical Engineering

Stony Brook University

December 2018

Stony Brook University

The Graduate School

Eric Tola

We, the thesis committee for the above candidate for the

Master of Science degree, hereby recommend

acceptance of this thesis

Dr. Nilanjan Chakraborty
Assistant Professor of the Department of Mechanical Engineering

Dr. Imin Kao
Professor of the Department of Mechanical Engineering

Dr. Qiaode Ge
Chair and Professor of the Department of Mechanical Engineering

Dr. Anurag Purwar
Research Professor of the Department of Mechanical Engineering

This thesis is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Thesis

In-Hand Manipulation Planning with Underactuated Robotic Hands

by

Eric Tola

Master of Science

in

Mechanical Engineering

Stony Brook University

2018

This thesis presents a method for performing in-hand manipulation with underactuated robotic hands by using combinations of one degree of freedom motion primitives derived from specific in-hand manipulation skills. Occupational therapy literature has defined five in-hand manipulation skills in order to develop a standardized test for the in-hand manipulation capabilities of their patients. The five skills are: palm to finger tip translation, finger tip to palm translation, shift, simple and complex rotations. This work focuses on composing the first three in-hand manipulation skills as combinations of one degree of freedom motion primitives. A 3D two link underactuated finger and object interaction can be modeled, instantaneously, as a 2D planar closed loop mechanism with one degree of freedom by assuming there is no slip between the contact of the finger and object. The no slip condition guarantees that the contact point does not change so we can model the contact as a joint in a mechanism. By solving the inverse dynamics problem as a linear optimization program at each instant, the torques at the joints that can perform the desired motion primitive are found and are used for a torque computed control plan. Since the finger is underactuated and only has one controllable DoF, modeling the motion primitive in this manner has two advantages. First, a 2D planar closed loop mechanism only has 1 DoF which means if the assumptions are maintained, the motion primitive can be performed by a 1 DoF finger. Second, each actuator will provide torques to each of the two joints of each finger and the relationship between each of the joint torques. The transmission coefficient of the finger describes this relationship. A physical robotic hand with the transmission capabilities to provide a ratio of torques between the two joints can utilize these motion primitives to perform in-hand manipulation. When considering the multi-fingered implementation of the in-hand manipulation motion primitives, one of the fingers must be used to perform the motion primitive while the other fingers are used to support the object. The supporting fingers' contact forces are computed using a second order cone program that ensures the contact forces satisfy their friction cone constraints. These contact forces provide the necessary object wrenches to balance object wrenches generated by the finger manipulating the object and any external object wrenches that may break the constraints during each instance of the motion primitive. A combination of motion primitives with a combination of manipulating and supporting fingers could perform in hand-manipulation with underactuated robotic hands.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A robot's ability to traverse and manipulate objects in its environment determines the number of tasks that it can perform reliably. Focusing on object manipulation, generally, objects are either grasped or reconfigured depending on the desired task. An object is grasped in order to ensure it will not fall when being held by the robot. The object can then be transported to a different location or secured for another operation. In some occasions, the object being grasped is not in the optimal configuration for its use in a desired task. The object needs to be reconfigured in order to ensure it can be used efficiently. In-Hand Manipulation (IHM) is the ability to move a grasped object with respect to the palm using the fingers. This allows for a robot to grab an object, change its configuration depending on the task, and then perform the task. If done properly, IHM is an extremely powerful tool that can extend the dexterous manipulation tasks that Underactuated Robotic Hands (URH) can perform.

This thesis presents a method of utilizing simplified IHM skills as a way of planning IHM with underactuated robotic hands. The advantages of underactuation is that they are generally cheaper, light weight and allow for the hand to adapt to object geometries during grasping. Underactuation comes at the price of a lack of control, which has limited underactated hands mainly to power grasping applications where object geometries might be irregular. The planning method presented will help allow URHs plan for IHM tasks. The IHM skills are broken down into combinations of motion primitives. Each motion primitive is a 1 DoF motion which can be approximated, instantaneously, as a four bar closed loop mechanism. We consider IHM with a 2 DoF finger with only 1 actuator meaning each finger is underactuated by a degree of one. Since there is only 1 controllable DoF and the motion primitive is a 1 DoF motion, theoretically, an underactuated 2R finger can perform that motion primitive.

The finger is an open-chain manipulator meaning that torques are required at both joints in order to ensure the finger satisfies the closed loop mechanism model. Then how can one use an underactuated hand in order to perform the motion primitive in a real world scenario? When solving for the torques at the joints required to perform each motion primitive, a ratio between the joint torques is found. This ratio is called the Transmission Coefficient of finger. For each time step of each motion primitive, the required transmission coefficient can be calculated. The robot hand being considered contains a Transmission Mechanism which can vary the transmission coefficient or the torques at the 1st joint to the 2nd joint of the finger. The design of the robotic hand must be considered with along with its control strategies. Building a hand that you cannot control or designing control schemes for a hand that is infeasible both present very real challenges to current progress. However, this does not mean one should dismiss ideas that are not feasible at the current time since they may, or may not, become useful in the future with continued research. The design of the transmission mechanism for the URH is very important to the method. Depending on the physical design, the URH will have a range of feasible transmission coefficient values which will determine which motion primitives the URH can or cannot perform under particular parameters. This can be used during the planning to ensure that the URH can perform all of the desired motion primitives required for the IHM task.

Occupational literature has been interested in IHM with regards to developing a test to help detect patients at younger ages with fine motor function issues or muscular problems. Charlotte E. Exner developed a test for IHM by defining a system to rank fine motor skills [30, 29]. Dr. Exner identified 5 types of skills which where shared between different IHM tasks. In her papers Dr. Exner defined 5 types of IHM skills as

1. finger-to-palm translation - moving an object from the finger pads to the palm of the hand.

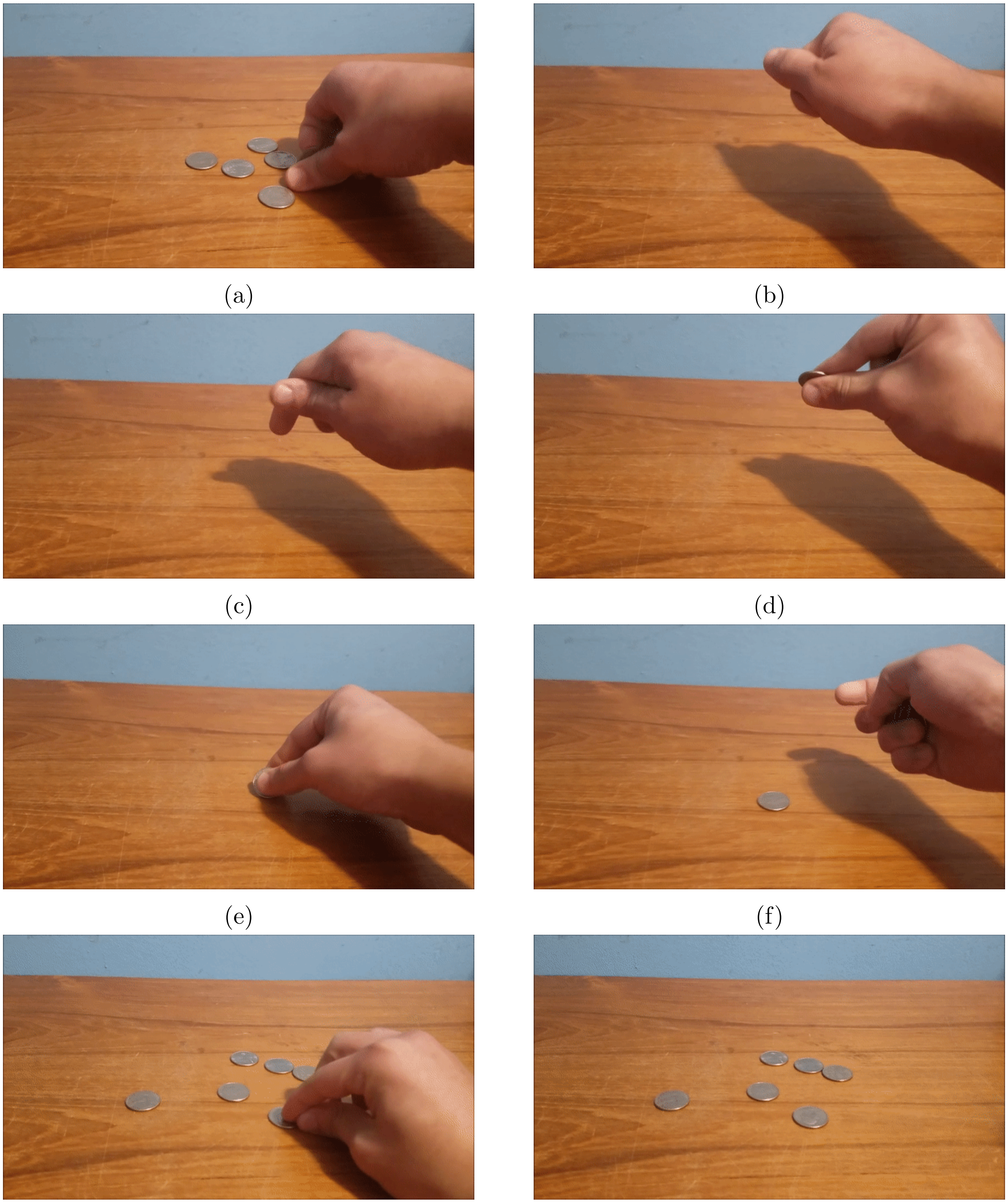
2. palm-to-finger translation - moving an object from the palm to the finger pads.

3. shift - using the finer pads to produce a slight linear adjustment of the object.

4. simple rotation - rotating an object by using the thumb in opposition to the fingers. The fingers usually act as a unit and the object is usually rotated less than 180°.

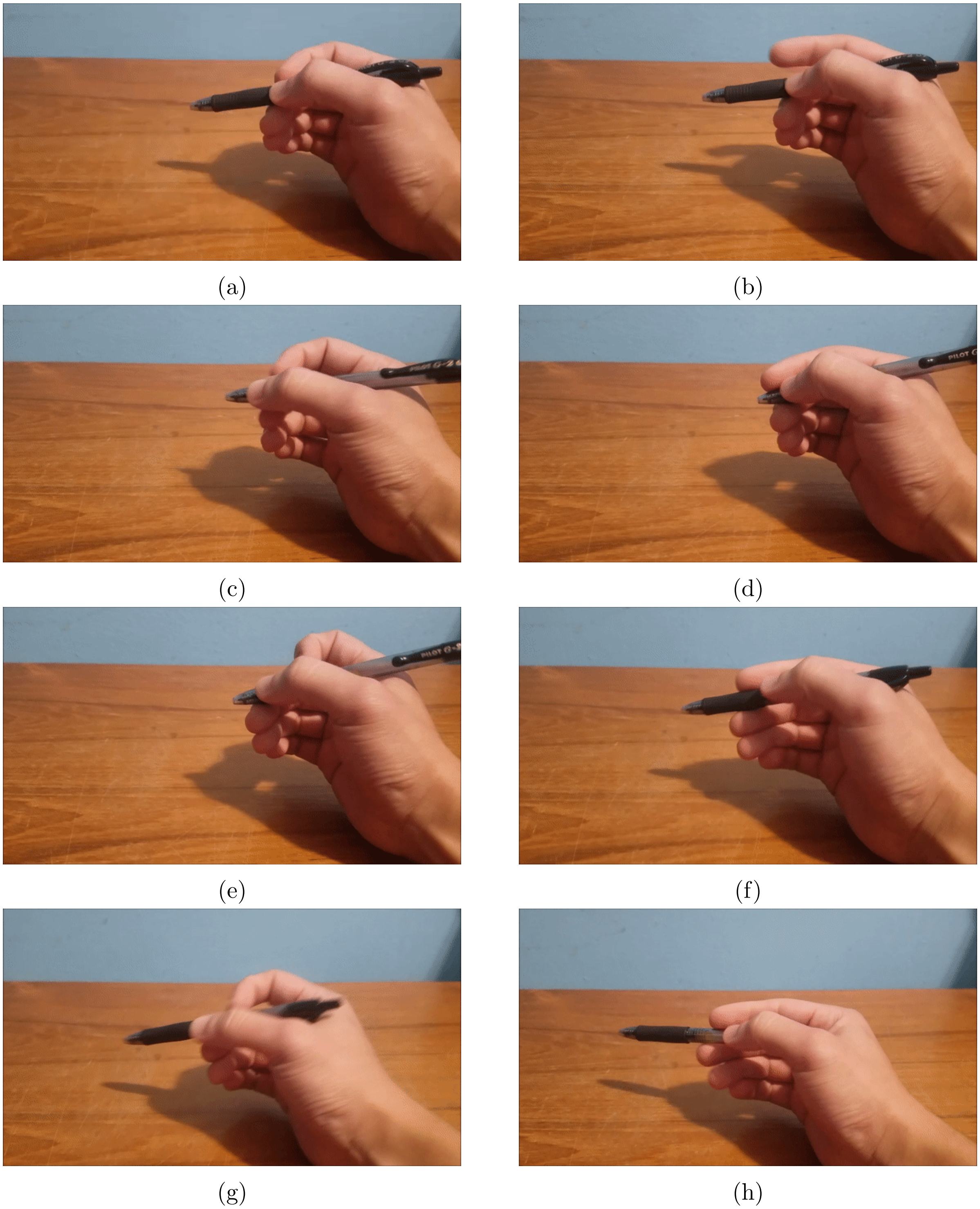
5. complex rotation - rotating an object at least 180° by using differentiation of finger movements and active thumb movements.

Some examples of daily activities that require IHM skills include picking up an pencil and rotating it to a position for writing, holding coins or other small objects in the hand and bringing one out to the finger tips, separating thin pages in a book, and flipping a bottle cap before placing it on the bottle. All of these tasks require different combinations of manipulating and supporting fingers. Some of the tasks can even be done with more than one object in the hand. Below are some figures to demonstrate the five IHM skills.



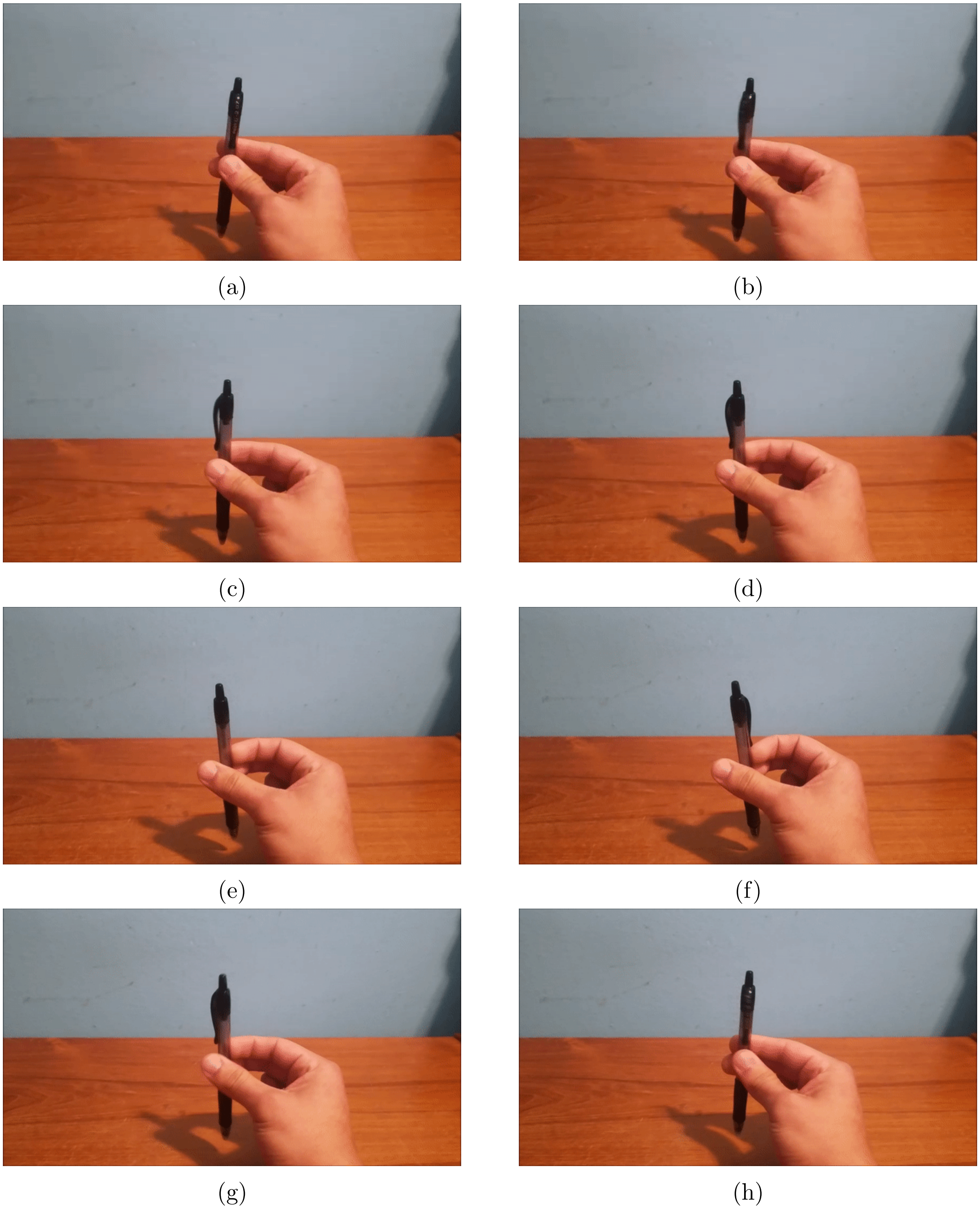
(a) (b) (c) (d) (e) (f) (g) (h)

Figure 1.1: Finger-to-Palm and Palm-to-Finger Translation



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 1.2: Shift



(a)

(b)

(c)

(d)

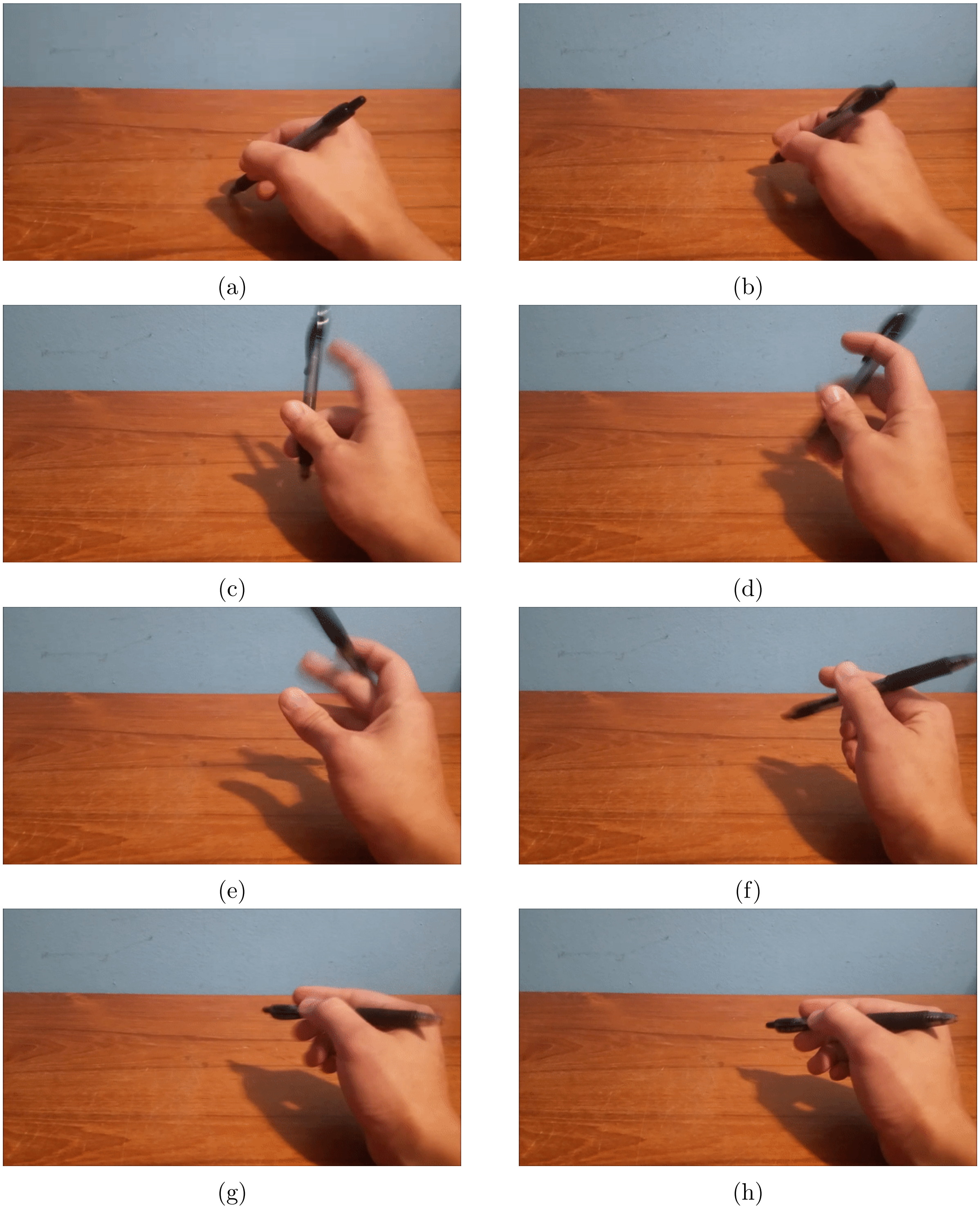
(e)

(f)

(g)

(h)

Figure 1.3: Simple Rotation



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 1.4: Complex Rotation

One of the issues in robotic hand literature is the lack of a metric in order to compare IHM of different robotic hands. I propose that using the 5 IHM skills as a type of metric for IHM of robotic hands. Similar to how an Occupational Therapist might test a patient, a robot hand could be tasked with performing some of the IHM tasks stated above. In order to perform those tasks, the robot hand must be able to perform the 5 IHM skills. Robots can be scored similar to how a patient might be scored on how successful they were at the IHM tasks.

When considering IHM with multiple fingers, there is not a clear choice on how to decide which finger does what when given a desired object configuration. How does one robustly plan for IHM tasks? If the hand performing IHM has 3 fingers, 1 of the fingers will be performing the desired motion primitive while the other 2 fingers provide the supporting object wrenches to keep the assumptions of the motion primitive valid. Each of the fingers roles may then change in order to perform the desired IHM task. When using the motion primitives to design the IHM task, the desired object motion must be mapped from the 2D problem to the 3D problem. The object forces due to manpulation by the fingers are the transformed into object wrenches in 3D. The object wrenches are then used to perform a second-order cone program convex optimization in order to solve for the contact wrenches required given contact points and contact frames [12].

Figure 1.5 displays the method at the high level. First, a desired IHM skill is identified and then it is defined as a task. The task is then broken up into a set of motion primitives. The motion primitives are then converted into wrenches in 3D in order to solve for the support finger wrenches given contact points. Once the support finger forces and grasp locations are determined the results can be turned into control signals for actuators in order to test on hardware. In a real world scenario, sending control signals to your hardware in an open-loop manner will not work properly due to the errors present in the system. The controller must have some feedback in order to correct for errors and ensure that the object being manipulated by the finger is following the desired plan. The algorithm first plans the desired motion primitive. The torques from the first time step of the plan are then fed as control signals to the hardware. The hardware then performs that time step of the motion and the feedback from the sensors is given to the planner. The plan is then recomputed in order to correct for any present errors. This loop continues to run until the object is brought to the desired configuration within some error criteria.

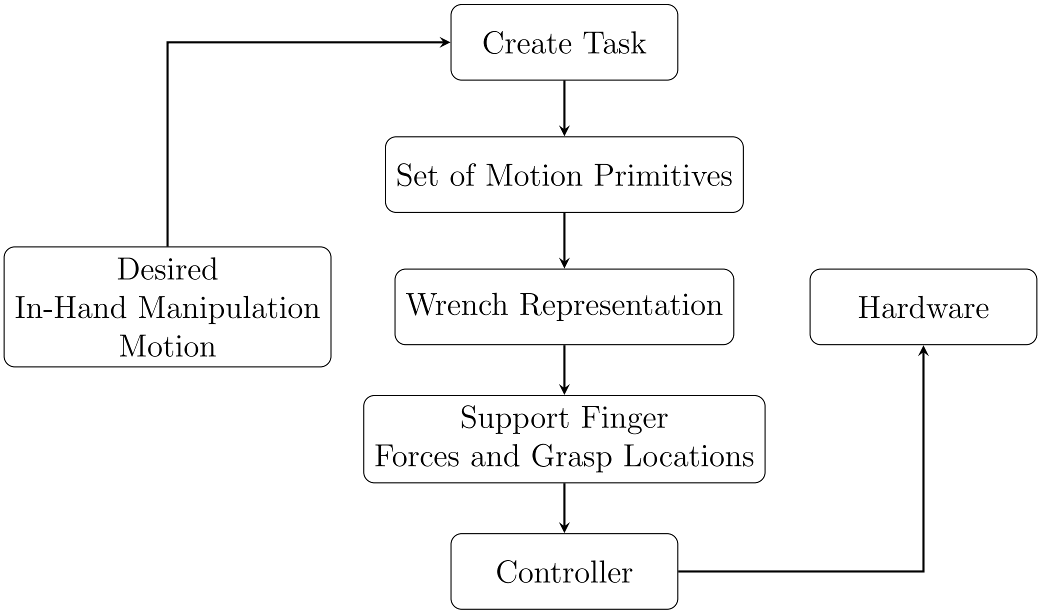


Figure 1.5: In-Hand Manipulation Planning Method.

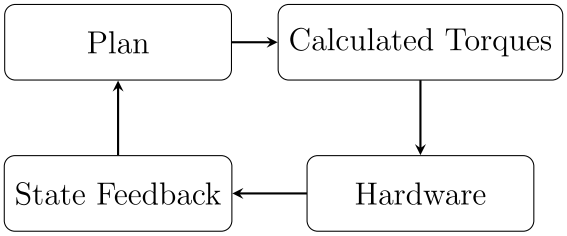


Figure 1.6: Closed-Loop In-Hand Manipulation Algorithm.

1.1 Related Work

There are few papers, to the best of my knowledge, that handle both underactuated robotic hands and IHM simultaneously. Furthermore, as far as I know, current literature has no principle way to perform IHM planning with URHs. The work presented in this paper aims to fill that gap and provide a method for performing IHM planning with URHs.

In a paper by L. Odhner and A. Dollar, the authors present a method to perform dexterous manipulation tasks with an underactuated robotic hand [68]. The paper shows that holonomomically constrained grasps can be analyzed to determine the manifold of stable object configurations that can be performed from an initial configuration. Planar hand-object systems are holonomically constrained if the contacts can be modeled with idealized normal, shear and no-slip rolling constraints. This formulation allows for a function mapping actuator coordinates to object coordinates for non-infinitesimal motions. The results of this method presents that the planar hand could translate and rotate the object some distance while maintaing the grasp. An important difference between the dexterous manipulation described in paper and IHM is that the contact points did not change during the manipulation. Due to this, the resulting object motion was small thus making the number of potential dexterous manipulation tasks small. In a following paper, the authors expanded their work to perform a flip-and-pinch process with an URH [70]. Their results show that thin objects can be grasped from a table top with their URH design and grasping strategy. When modeling the flip-and-pinch process, the authors modeled the finger-object interactions in a similar manner presented in this thesis. When the fingers make contact with the object they model the system as a four bar linkage. The contact point is modeled as a pin joint with the no-slip constraint. The authors state that this underactuated approach to the task is easier in some areas to a fully actuated finger. This is because no computation is needed to calculate the trajectory of the object or the fingers. Due to the design of the URH and the passive constraints, the flip-and-pinch process can be performed on a varying number of thin objects. There are a myriad of different types of URHs developed in previous research, all with the similar goal of utilizing the advantages of URHs while minimizing the lack of control for dexterous tasks [28, 4, 5, 36, 60, 15, 18, 88, 69, 91, 81, 27, 65]. I have listed the vast number of hands I have found in the research in case the reader is curious to explore the literature.

The manipulability ellipsoids of URH was studied by a group of researchers from Italy in [74]. The paper mainly focuses on grasps instead of performing IHM tasks with URHs but provides a great analysis of the kinematic and force manipulability of URHs which was not previously provided. This work follows a lineage of works revolving around the idea of hand synergies in grasping. This idea of postural synergies is the trajectories while grasping for a human hand are in a configuration space of a smaller dimension than the DoFs of the hand. The central nervous system might use these postural synergies to simplify the control of the human hand during grasps. They called this configuration space the space of postural synergies or the eigengrasp space [34]. Interestingly enough, this research began as a neuroscience paper where a study was performed on the static hand posture when grasping to use different tools [77]. They found that the subjects adopted distinct hand shapes for the various objects but the joint angles of the fingers did not vary independently. This meant that control of the hand involves the regulation of a general shape of the hand while simultaneously providing small fine motor adjustments during grasps. Many papers after this looked into theory and further studies were performed [63, 20, 22, 23, 24].

In other areas, these papers focus on deriving models for URHs [11, 83, 48, 52]. The focus in this area is typically driven to analize and design better URHs. Each paper assumes a type of underactuated finger or URH with a particular tendon configuration and design. These papers provide a large number of different analysis, synthesis and modeling techniques [71, 54, 78, 79, 84, 16, 43, 44, 45, 57, 90]. Other papers develop different underactuated manipulation model and control techniques [58, 61, 3, 82, 76, 41, 33]. Large studies on grasping and force closure have also been peformed [55, 10, 12]. Empirical methods have been researched as well where different machine learning methods have been applied to robotic hands research[32, 31, 73, 75, 50, 51, 37, 87, 86, 2]. This paper utilizes a combination of analytical numerical and machine learning methods to perform some in-hand manipulation with their adaptive robot hands [56]. The models learned are just the dynamic models for slipping and other difficult to model phenomena. While impressive the dexterous IHM is limited to small in-hand motions like the work from [68].

Object reconfigurations can be done by placing the object down and regrapsing the object multiple times in order to achieve the desired object configuration. Two or more robots can work together, passing the object between their end effectors, in order to perform object reconfigurations as well. Extrinsic manipulation, using external objects or forces to aid in object manipulation, is also another valid type of object reconfiguration technique. This utilizes external forces from the environment like gravity or external fixed objects [26]. This paper performs IHM with push-primitives described in their work [17]. It pushes the grasped object against its environment in order to perform in-hand dexterous manipulation with a simple manipulator. A model is developed to predict the force required by the external pusher to break the grasp and move the object. The results of the paper are promising in that with a simple manipulator and setup, more advanced manipulation tasks can be performed efficiently. Many other papers have explored extrinisc manipulation as well [42, 40, 1, 6, 8, 7, 80]. This paper provides an example of a robot hand design inspired by extrinsic manipulation with interesting results [59].

Prior research has studied IHM with URHs by developing hands with specific configurations in order to perform particular dexterous IHM tasks. A robust planning method could allow for different tasks to studied aprori and guide designs of URHs. Figure 1.7 is a combination of work from a few papers which is another form of guided robotic hand designs [43, 44, 45]. By determining the required object wrenches in order to perform a particular dexterous manipulation task each of the equations will provide information on the physical capabilities a robotic hand will need to have. Hardware to exhibit these particular capabilities can be studied in tandem with this mapping. To the best of my knowledge, there is no prior research that explicitly provides a planning method for IHM with URHs.

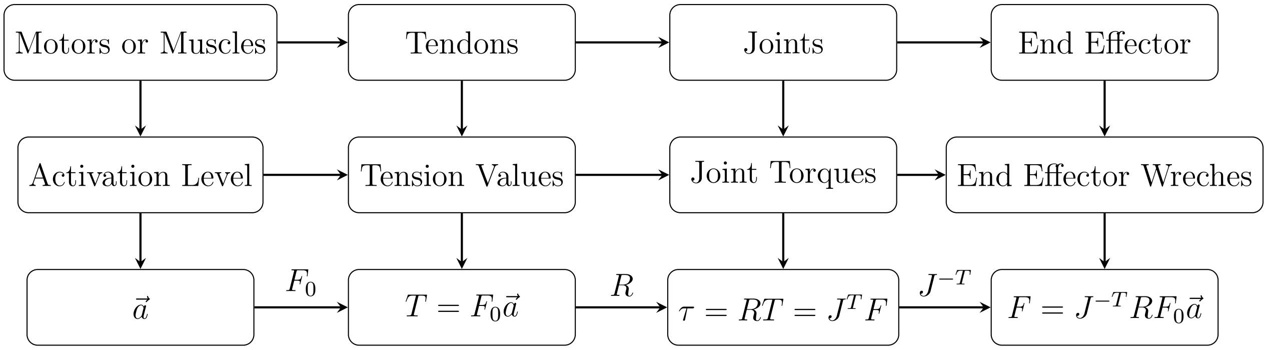


Figure 1.7: Mapping of End Effector Wrenches to Motor Inputs.

In the work presented in the PhD dissertation [38], robotic IHM is shown with a ”push and support” method which is similar to the method used in this thesis. The idea is simply that one or multiple fingers provide the active contact forces to perform the desired motion while the other fingers provide adequate supporting contact forces. The support contact forces must not only prevent the object from being released from the grasp but must not restrict the object so the active finger cannot perform its function. In the dissertation, the robotic hand being used is a fully actuated robotic hand and is currently one of the better hands available to researchers. The results show that indeed it is possible to perform desired IHM of the grasped object with a fully-actuated hand. The author utilized a reinforcement learning scheme in order to design the controller for the robotic hand. Although the motions shown were small, I find the results promising. This means that with further investigation IHM can be performed with robotic hands. Whether or not they can be performed with URHs is still to be proven.

Chapter 2

In-Hand Manipulation Motion Primitives

This chapter will explain the method of developing instantaneous mechanism motion primitives. A defined task T is described as a set of desired object configurations at each time step. The path that task T takes can be approximated by the best combination of motion primitives to perform that motion. Each motion primitive is a matrix of joint torques which corresponds to a part of the overall task. Each of the motion primitives are calculated using a linear optimization program to minimize torque at the joints to perform the desired motion. The linear system of equations and constraints between joint forces and torques are derived from the sum of forces of the four bar mechanism dynamic analysis. The desired path is generated using SVAJ curves which define position (S), velocity (V), acceleration (A), and jerk (J) of a motion at the boundary conditions. Each curve is defined using a polynomial fit given each of the boundary conditions for the path. This path is then used to solve for the fingers' respective positions, velocity, and accelerations given a known contact point. At each time step of the motion primitive, the linear optimization program minimizes the torques at the joints to perform the motion designed from the SVAJ curves. The linear optimization program is subject to equality constraints from the dynamic analysis and inequality constraints from the 2D friction cone constraints. The calculated torques ensure contact forces that both satisfy the friction cone and perform the desired object motion designed using the SVAJ curve.

2.1 Task Definition

It is important to first explicitly define a task in the context of this problem. A desired task can be defined as desired object configurations g_{obj} at each time instance for a frame fixed at the objects center of mass. Each object configuration is with respect to the hand base frame. Let T represent a desired task. The desired task T may be written as follows

$$T = \begin{bmatrix} g_{obj}^{(1)} & \dots & g_{obj}^{(k)} \end{bmatrix} \tag{2.1}$$

where k is the total number of configurations required to perform the given task. The configuration $g \in SE(3)$ is the 4 x 4 homogeneous representation of the transformation matrix. It is defined as

$$g_{ab} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix} \tag{2.2}$$

which specifies the configuration of frame b with respect to frame a. The rotation matrix $R_{ab} \in SO(3)$ is a 3 x 3 rotation matrix which defines the orientation of frame b with respect to frame a. The elements of R depend on the axis about which the frame is rotating. Basic rotations about each respective axis is given as

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{2.3}$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{2.4}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

where each axis is defined using the right-hand rule. The angles $\alpha\beta\gamma$ defines the how much rotation occurs about each axis. General rotations are obtained by using matrix multiplication of each of the basic rotations. The total rotation matrix R is then defined as a product of rotations about each of the axis.

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha) \tag{2.6}$$

The position $p_{ab} \in \mathrm{R}^3$ is a 3 x 1 vector that defines the position of frame b with respect to frame a. This position vector is just the x, y, and z components of the frame and is written as follows.

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.7}$$

The homogeneous form appends 1 to point coordinates and 0 to vectors. This representation gives intuition about the rules on manipulating vectors and points.

1. The sum or difference of a vector is a vector.

2. The sum of a vector and a point is a point.

3. The difference between two points is a vector.

4. The sum of two points has no physical meaning.

This changes the dimension from 3 to 4 at the expense of convenience of the homogeneous form. This allows for rigid body transformations to be represented as a group which implies

1. If $g_1, g_2 \in SE(3)$, then $g_1 g_2 \in SE(3)$.

2. The identity element, I, exists as a 4 x 4 matrix in SE(3).

3. If $g \in SE(3)$, then its inverse is defined as

$$g^{-1} = \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix} \in SE(3) \tag{2.8}$$

4. The composition rule for rigid body transformations is associative meaning that transformations may rearranged as long as the sequence of rigid body transformations is not changed.

A desired task with two object configurations is shown in figure 2.1. A desired task may be defined with more desired object configurations but for simplicity just two desired object configurations are considered. Given the initial and the final pose, motion primitives are used to plan for a motion between the desired object configurations.

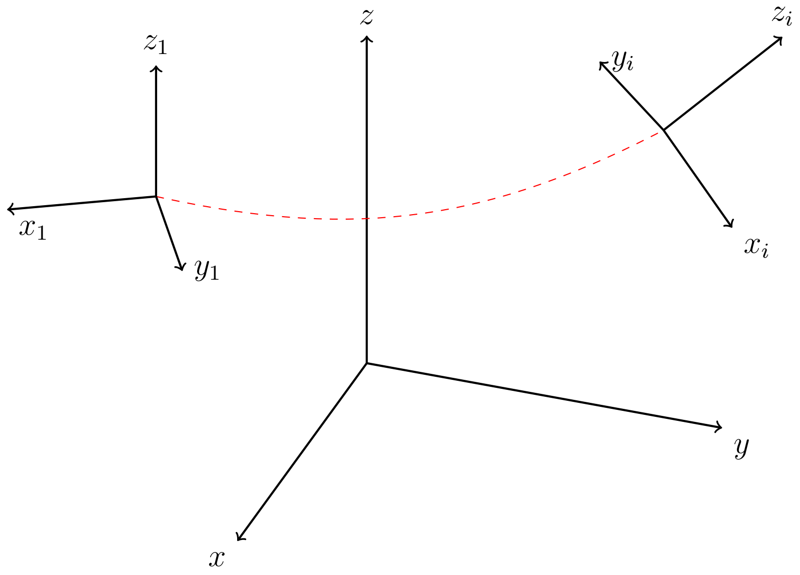


Figure 2.1: Desired task with two configurations.

2.2 Instantaneous Mechanism Motion Primitives

Performing manipulation with a 1 DoF underactuated finger requires a clever planning strategy in order to not be limited by the lack of full control of the finger. This is handled by modeling the underactuated finger and object, instantaneously, as a four-bar linkage. A four-bar linkage has been shown to be the simplest possible pin-jointed mechanism for a 1 DoF controlled motion [67]. Four-bar linkages are the most commonly used mechanism and are extremely versatile in the motions it has been able to generate. These advantages are highly desirable for the feasibility of this method. A wide variety of motions can be generated meaning many different motion primitives may be developed and each motion primitive developed is a 1 DoF motion. By generating 1 DoF motion for a 1 DoF underactuated finger, this ensures that the motion can be performed if the assumptions and constraints of the mechanism remain true.

Modeling the finger and object interaction as a mechanism assumes that the contact between the finger and object is modeled as a pin-joint thus there is no slip at the contact point between the finger and object. This must also be carefully handled to ensure that the forces exerted by finger can only push the object or pull the object through the use of contact friction.

2.2.1 Motion Primitives

A desired IHM task T, composed of desired object configurations g_{obj}, can be composed of combinations of motion primitives S_T. The matrix, S_T, is the matrix containing all of the motion primitives, S_m, for task T

$$S_T = \begin{bmatrix} S_1 & \cdots & S_m \end{bmatrix} \tag{2.9}$$

where m is the number of motion primitives used to approximate task T. Each motion primitive S_m is a j x n matrix composed of a vector of torques τ at the joints of the finger at each instance. The vector τ is composed as

$$\tau = \begin{bmatrix} \tau_1 & \cdots & \tau_j \end{bmatrix}^T \tag{2.10}$$

where j is the number of joints of the finger and n is the number of discretizations used.

$$S_m = \begin{bmatrix} \tau_1 & \cdots & \tau_n \end{bmatrix} \tag{2.11}$$

In this work, S_m is a 2 x n matrix since each finger has only two joints.

$$S_m = \begin{bmatrix} \tau_1 & \cdots & \tau_{1n} \\ \tau_2 & \cdots & \tau_{2n} \end{bmatrix} \tag{2.12}$$

2.2.2 Trajectory Generation

When performing any desired motion, the kinematics of the trajectories must first be designed. Each trajectory in this work includes the position (S), velocity (V), acceleration (A) and jerk (J) of the object. The trajectories are designed as segments of boundary conditions of the motion. The motion between the boundary condition points are solved for using n power polynomial functions. The value of n depends on the number of boundary conditions. The boundary condition $b_c \in R^{4\times 2}$ is defined as

$$b_c = \begin{bmatrix} s_i & s_f \\ v_i & v_f \\ a_i & a_f \\ j_i & j_f \end{bmatrix} \tag{2.13}$$

where s_i and s_f are the initial and final positions of the segment. The other boundary conditions are defined similarly. Depending on which of the four variables you wanted to constrain your boundary conditions will be appropriately sized. In this case, the first three variables, S, V, and A, are constrained which makes $b_c \in R^{3\times 2}$.

$$b_c = \begin{bmatrix} s_i & s_f \\ v_i & v_f \\ a_i & a_f \end{bmatrix} \tag{2.14}$$

Thus the polynomial function becomes what is known as a 3-4-5 polynomial and is defined as

$$s = C_0 + C_1x + C_2x^2 + C_3x^3 + C_4x^4 + C_5x^5 \tag{2.15}$$

$$v = C_1 + 2C_2x + 3C_3x^2 + 4C_4x^3 + 5C_5x^4 \tag{2.16}$$

$$a = 2C_2 + 6C_3x + 12C_4x^2 + 20C_5x^3 \tag{2.17}$$

$$j = 6C_3 + 24C_4x + 60C_5x^2 \tag{2.18}$$

and can be solved as a linear system of equation given the boundary conditions from 2.14. Converting to matrix form, the equations 2.15, becomes

$$A = \begin{bmatrix} 1 & x_i & x_i^2 & x_i^3 & x_i^4 & x_i^5 \\ 0 & 1 & 2x_i & 3x_i^2 & 4x_i^3 & 5x_i^4 \\ 0 & 0 & 2 & 6x_i & 12x_i^2 & 20x_i^3 \\ 1 & x_f & x_f^2 & x_f^3 & x_f^4 & x_f^5 \\ 0 & 1 & 2x_f & 3x_f^2 & 4x_f^3 & 5x_f^4 \\ 0 & 0 & 2 & 6x_f & 12x_f^2 & 20x_f^3 \end{bmatrix} \tag{2.19}$$

the matrix $A \in \mathrm{R}^{6x6}$ where the values of x, in this case, is the time period between initial and final boundary conditions. In typical uses, the value of x would be defined by the cam angle. The coefficients become a 6 x 1 vector.

$$c = \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix} \tag{2.20}$$

The boundary conditions are formed into a 6 x 1 vector.

$$b = \begin{bmatrix} s_i \\ v_i \\ a_i \\ s_f \\ v_f \\ a_f \end{bmatrix} \tag{2.21}$$

Given the initial and final boundary conditions along with the initial and final time periods for the motion, the coefficients for the polynomial can be solved for as follows.

$$c = A^{-1}b \tag{2.22}$$

This provides the vector c which contains all of the coefficients for the polynomials for the desired trajectory.

2.2.3 Calculating Joint Torques with a Linear Optimization Program

If a motion is able to be efficiently approximated as an instantaneous mechanism, we can formulate the force and torque equilibrium equations of the finger object system as a set of linear equations. We can then setup the problem as a linear program with linear constraints to ensure feasibility of the motion primitive. We perform a dynamic force analysis using Newton's equations

$$\sum \boldsymbol{F} = m\boldsymbol{a} \qquad \qquad \sum \tau = \mathrm{I}\boldsymbol{\alpha}$$

Where $\boldsymbol{F} \in \mathrm{R}^{2x1}$ is a 2 x 1 vector of forces in the x and y directions and $\tau \in \mathrm{R}$ is the torque in the z direction (out of the plane)

$$\boldsymbol{F} = \begin{bmatrix} F_x \\ F_y \end{bmatrix} \qquad \qquad \tau = \tau_z$$

The mass and the moment of inertia are variables m and I respectively. We model the links of the fingers to be a rod of length L rotating about its center so $I = \frac{1}{12}mL^2$. The object's moment of inertia changes with its shape.

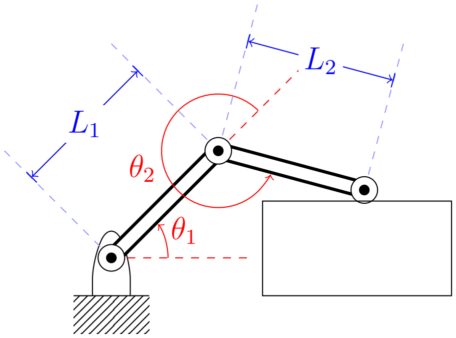


Figure 2.2: 2R Finger manipulating a rectangular object.

Consider the 2R finger in Figure 2.2 manipulating the object. By modeling the system as a closed-loop mechanism at an instantaneous time step, the equations of motion can be written for the desired motion. Assuming there is no slip at the contact point, we may model the interaction as a joint and the entire system as a mechanism. The finger-object configuration in Figure 2.2 mostly resembles a slider-crank mechanism. As in Figure 2.3, first draw free body diagrams for each of the links of the finger and the object.

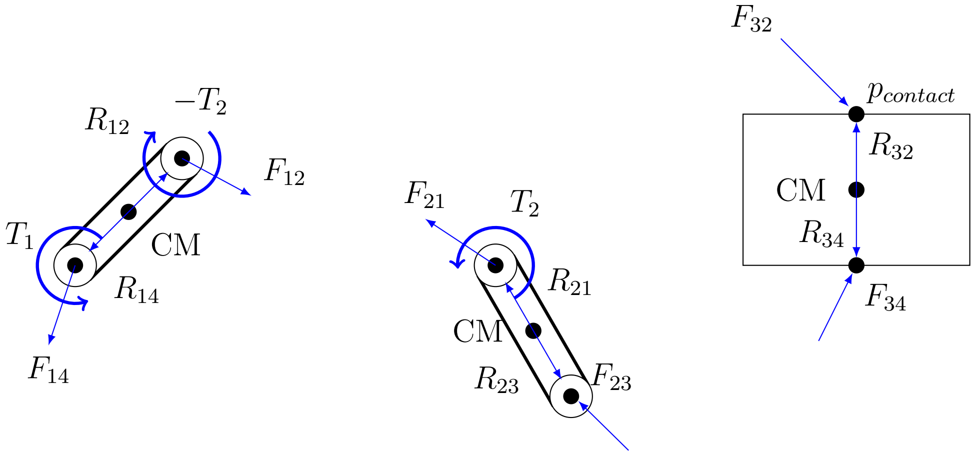


Figure 2.3: Free body diagram of a 2R finger manipulating a rectangular object

The vectors F are the forces acting at the joints of the mechanism, R is the radius vector giving the distance and direction of the force action from each link's respective center of mass. The torques are denoted by T. The subscripts determine from where each vector goes to and from. For example F_{14} describe a force acting on link 4, the ground link, from link 1. Similarly, the vector R_{14} is the radius from the joint between links 1 and 4 to the center of mass of link 1.

Summing the forces and torques acting on the mechanism sets up a set of linear equations to represent the equations of motion of the system. We write the set of linear equations for each of the links as follows

Link 1

$$F_{14x} + F_{12x} = m_1 a_{1x} - F_{g1x} \tag{2.23}$$

$$F_{14y} + F_{12y} = m_1 a_{1y} - F_{g1y} \tag{2.24}$$

$$T_1 + (R_{14x}F_{14y} - R_{14y}F_{14x}) + (R_{12x}F_{12y} - R_{12y}F_{12x}) - T_2 = I_1\alpha_1 \tag{2.25}$$

Link 2

$$F_{23x} - F_{12x} = m_2 a_{2x} - F_{g2x} \tag{2.26}$$

$$F_{23y} - F_{12y} = m_2 a_{2y} - F_{g2y} \tag{2.27}$$

$$T_2 + (R_{23x}F_{23y} - R_{23y}F_{23x}) - (R_{21x}F_{12y} - R_{21y}F_{12x}) = I_2\alpha_2 \tag{2.28}$$

Link 3

$$F_{34x} - F_{23x} = m_3 a_{3x} - F_{g3x} \tag{2.29}$$

$$F_{34y} - F_{23y} = m_3 a_{3y} - F_{g3y} \tag{2.30}$$

$$(R_{34x}F_{34y} - R_{34y}F_{34x}) - (R_{32x}F_{23y} - R_{32y}F_{23x}) = I_3\alpha_3 \tag{2.31}$$

where each of the links has undergone a force and torque balance. Note that the reaction torque of the torque at the second joint is considered in the first link. We end up with a set of a total of 9 linear equations and 10 variables. However, our finger is underactuated so the torques T_1 and T_2 are not independent. The relationship is described as

$$T_1 = cT_2 \tag{2.32}$$

where c is the transmission coefficient of the finger. This value is based on the design of the torque transmission mechanism and how joint torques are physically achieved.

We then define the matrix A_{eq} as the coefficient matrix for the set of linear equations as

$$A_{eq} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -R_{14y} & R_{14x} & -R_{12x} & R_{12y} & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_{21y} & -R_{21x} & -R_{23x} & R_{23y} & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_{32y} & -R_{32x} & -R_{34x} & R_{34y} & 0 & 0 \end{bmatrix} \tag{2.33}$$

We let x be the force-torque vector and define it as

$$x_{eq} = \begin{bmatrix} F_{14x} \\ F_{14y} \\ F_{12x} \\ F_{12y} \\ F_{23x} \\ F_{23y} \\ F_{34x} \\ F_{34y} \\ T_1 \\ T_2 \end{bmatrix} \tag{2.34}$$

Note that all of the forces are given with respect to the base frame of the finger. We then let the vector b_{eq} be equal to the right-hand side of equations 2.24 - 2.31.

$$b_{eq} = \begin{bmatrix} m_1 a_{1x} - F_{g1x} \\ m_1 a_{1y} - F_{g1y} \\ I_1 \alpha_1 \\ m_2 a_{2x} - F_{g2x} \\ m_2 a_{2y} - F_{g2y} \\ I_2 \alpha_2 \\ m_3 a_{3x} - F_{g3x} \\ m_3 a_{3y} - F_{g3y} \\ I_3 \alpha_3 \end{bmatrix} \tag{2.35}$$

Certain motion primitives have simplifications that can be made to reduce the total number of variables and equations. Undesired linear and rotational accelerations can be set to zero to solve for desired motion primitives that do not allow the object to have accelerations in those directions.

The linear program can now be formulated in order to minimize the torques at the joints for a given motion. Due to a no-slip condition at the contact point, a two-dimensional friction cone constraint is added to the linear program to ensure that all the contact forces lie within the friction cone thus preventing slip. This ensures that none of the contact forces can cause the contact point to change during manipulation. The friction cone constraints must also be satisfied by the contact between the object and the ground. The friction cone constraint can be written as two inequalities in the following way

$$|F_{23x}| \leq \mu_2 F_{23y} \tag{2.36}$$

$$|F_{23x}| - \mu_2 F_{23y} \leq 0 \tag{2.37}$$

$$F_{23x} - \mu_2 F_{23y} \leq 0 \tag{2.38}$$

$$-F_{23x} - \mu_2 F_{23y} \leq 0 \tag{2.39}$$

Upper and lower bounds can be placed on the forces and torques based on physical limitations in the hardware. This includes ensuring that the y-component of contact force between the finger and the object is strictly positive. This physically represents the inability of the finger to pull on the object without the use of friction. The linear program can now be written as follows

$$
\begin{aligned}
\text{minimize} \quad & \tau \\
\text{subject to} \quad & A_{eq}x = b_{eq} \\
& lb \geq x \leq ub \\
& F_{23x} - \mu_2 F_{23y} \leq 0 \\
& -F_{23x} - \mu_2 F_{23y} \leq 0 \\
& F_{34x} - \mu_1 F_{34y} \leq 0 \\
& -F_{34x} - \mu_1 F_{34y} \leq 0
\end{aligned}
\tag{2.40}
$$

where ub and lb are the upper and lower bounds for the variables in x. The linear program is solved for each segment of the motion primitive. The torques for each of the motion primitives S_m which approximate task T are calculated allowing for a torque computed control to perform the desired task T.

2.3 In-Hand Manipulation Motion Primitive Examples

In this section an example of each of the three motion primitives, sliding, rolling and tipping will be worked out. Utilizing the instantaneous mechanism method to approximate the motions, solutions are shown performing these motion primitives. The inputs parameters are given along with figures of the resulting data.

2.3.1 Sliding Motion Primitive

The desired motion for the *Sliding Motion Primitive*, slides the object along the floor. The object does not leave the ground so it has no y-component accelerations or angular accelerations. Table 2.1 lists the values for the parameters utilized to produce the results below.

Variables	**Values**	**Units**
Link 1 Length	0.08	m
Link 2 Length	0.05	m
Link 1 Mass	0.5	kg
Link 2 Mass	0.3	kg
Link 1 Moment of Inertia	$2.667 * 10^{-4}$	kg m^2
Link 2 Moment of Inertia	$6.25 * 10^{-5}$	kg m^2
Object Mass	0.05	kg
Object Height	0.03	m
Object Length	0.05	m
Object Moment of Inertia	0	kg m^2
Finger Object Coefficient of Friction	.9	%
Object Ground Coefficient of Friction	.1	%
Acceleration of Gravity	9.80665	$\frac{m}{s^2}$
Gravity Direction	$\frac{3\pi}{2}$	radians
Force of Gravity x - direction	$-1.8015 * 10^{-15}$	$\frac{m}{s^2}$
Force of Gravity y - direction	-9.8066	$\frac{m}{s^2}$
Global Time Step	0.01	s
Finger Contact Position wrt Object Frame	[0 0.015]	m
Motion Primitive Time Steps	[0.5 0.5 0.5]	s
Motion Primitive Desired x - Coordinates	[0.05 0.08 0.05 0.08]	m

Table 2.1: Sliding Motion Primitive Parameters

The moments of inertias were calculated assuming that the links were rectangles and their centers of gravity lie at the center of the link. We now formulate the Newton-Euler equations of motion. Figure 2.4 shows the free body diagram of the system. Each of the links and the object have force balance equations that define system. The equations can now be solved simultaneously for given kinematic input which in this case is the generated trajectory which is the desired motion of the object.

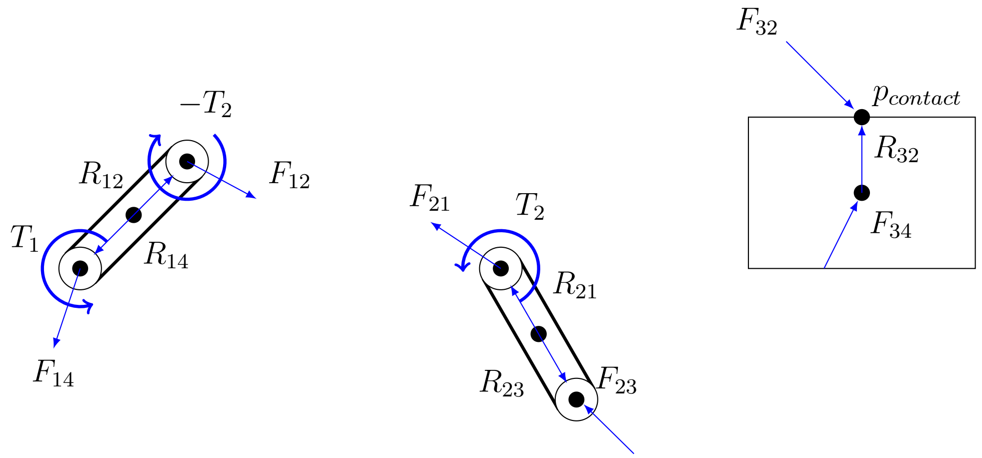


Figure 2.4: Sliding Free Body Diagrams

The object is in constant contact with the ground thus the reaction force F_{34} is assumed to act at the center of gravity of the object. The reaction torque due to the actuation torque at the second joint on the first link is accounted for as well.

The dynamics may now be written for each of the links and the object.

Link 1

$$F_{14x} + F_{12x} = m_1 a_{1x} - F_{g1x} \tag{2.41}$$

$$F_{14y} + F_{12y} = m_1 a_{1y} - F_{g1y} \tag{2.42}$$

$$T_1 - T_2 + (R_{14x}F_{14y} - R_{14y}F_{14x}) + (R_{12x}F_{12y} - R_{12y}F_{12x}) = I_1\alpha_1 \tag{2.43}$$

Link 2

$$F_{23x} - F_{12x} = m_2 a_{2x} - F_{g2x} \tag{2.44}$$

$$F_{23y} - F_{12y} = m_2 a_{2y} - F_{g2y} \tag{2.45}$$

$$T_2 + (R_{23x}F_{23y} - R_{23y}F_{23x}) - (R_{21x}F_{12y} - R_{21y}F_{12x}) = I_2\alpha_2 \tag{2.46}$$

Link 3

$$sign(v_{3x})\mu_1 F_{34y} - F_{23x} = m_3 a_{3x} - F_{g3x} \tag{2.47}$$

$$F_{34y} - F_{23y} = -F_{g3y} \tag{2.48}$$

$$(R_{34x} - R_{34y}(\pm\mu_1))F_{34y} - (R_{32x}F_{23y} - R_{32y}F_{23x}) = 0 \tag{2.49}$$

The forces due to gravity is considered on the system which is calculated as the product of mass of the body, the acceleration and the direction of gravity. The assumptions of zero y-component velocities, accelerations, and angular velocities and accelerations simplify the dynamics of the equations from 2.41 to 2.49. The angular acceleration for the object is written as $\alpha_3 = 0$ and the y-component of the linear acceleration can be written as $a_{3y} = 0$. The x-component of the ground reaction force F_{34} is assumed to be $F_{34x} = \pm\mu_1 F_{34y}$ since the object is sliding along the floor and thus the lateral force is due to friction. By writing the joint forces as follows, $F_{12} = -F_{21}$ and $F_{23} = -F_{32}$, this allows for the equations to be written as a linear system of equations. The coefficient matrix, A_{eq}, and vectors x, b_{eq}, can be written as follows.

$$A_{eq} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -R_{14y} & R_{14x} & -R_{12x} & R_{12y} & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & R_{21y} & -R_{21x} & -R_{23x} & R_{23y} & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & \pm\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix} \tag{2.50}$$

$$x = \begin{bmatrix} F_{14x} \\ F_{14y} \\ F_{12x} \\ F_{12y} \\ F_{23x} \\ F_{23y} \\ F_{34y} \\ T_1 \\ T_2 \end{bmatrix} \tag{2.51}$$

$$b_{eq} = \begin{bmatrix} m_1 a_{1x} - F_{g1x} \\ m_1 a_{1y} - F_{g1y} \\ I_1 \alpha_1 \\ m_2 a_{2x} - F_{g2x} \\ m_2 a_{2y} - F_{g2y} \\ I_2 \alpha_2 \\ m_3 a_{3x} - F_{g3x} \\ -F_{g3y} \end{bmatrix} \tag{2.52}$$

Now the motion primitive can be formulated into a linear program which minimizes the torque applied to the joints. The dynamics are used as equality constraints while the friction cone constraints and the bounds are the inequality constraints.

$$\begin{aligned} \text{minimize} \quad & \tau \\ \text{subject to} \quad & A_{eq}x = b_{eq} \\ & F_{23x} - \mu_2 F_{23y} && \leq 0 \\ & -F_{23x} - \mu_2 F_{23y} && \leq 0 \\ & F_{23y} && \geq 0 \\ & F_{34y} && \geq 0 \end{aligned} \tag{2.53}$$

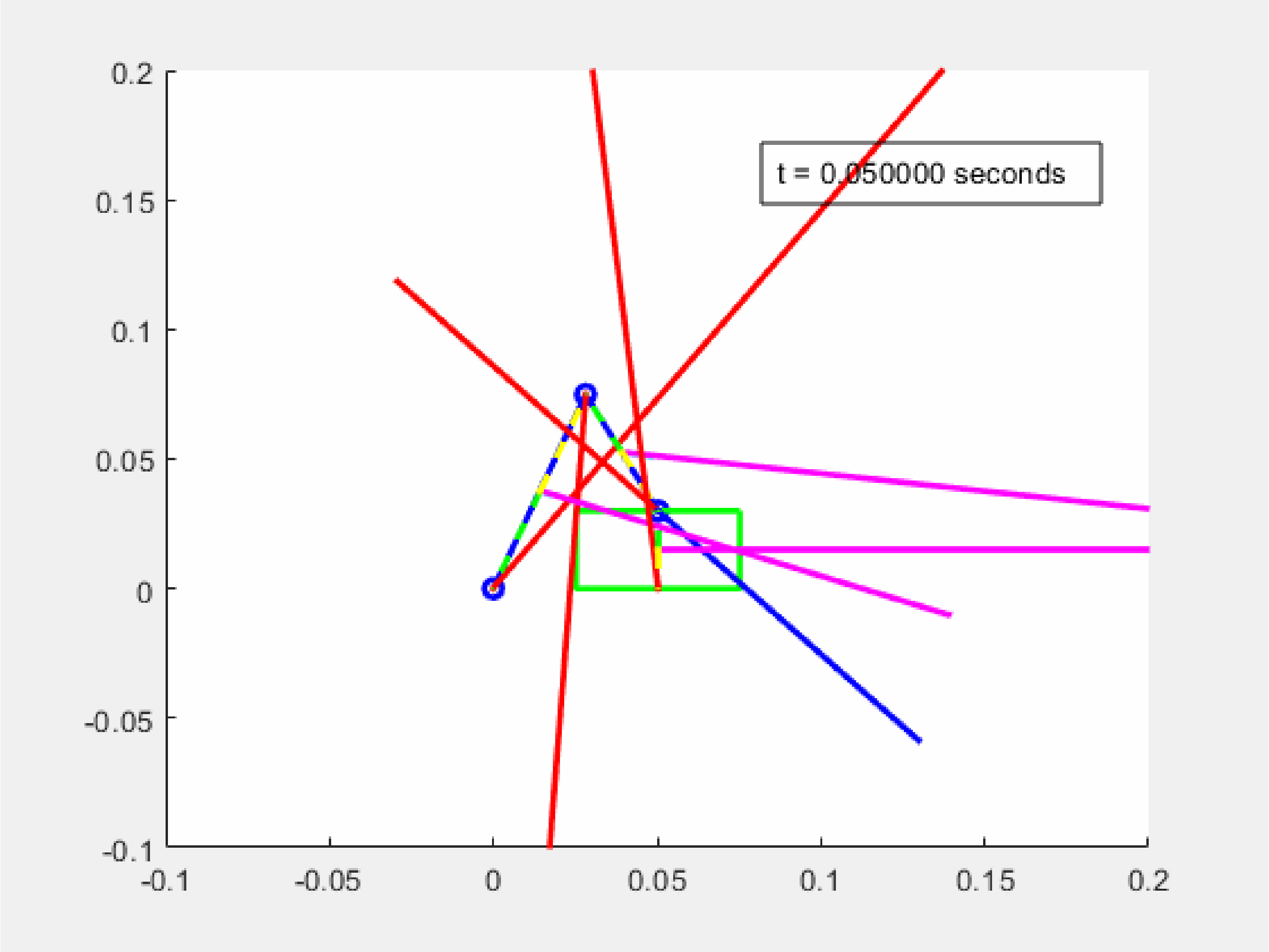


Figure 2.5: Sliding Motion Primitive

In order to solve the dynamics, first, the kinematics must be solved. The SVAJ curve is calculated using equations 2.15 and 2.22. Figure 2.1 gives the desired positions in meters in the world frame which is located at the base of the 2R linkage. Using the desired positions for the boundary conditions the desired trajectory can be generated.

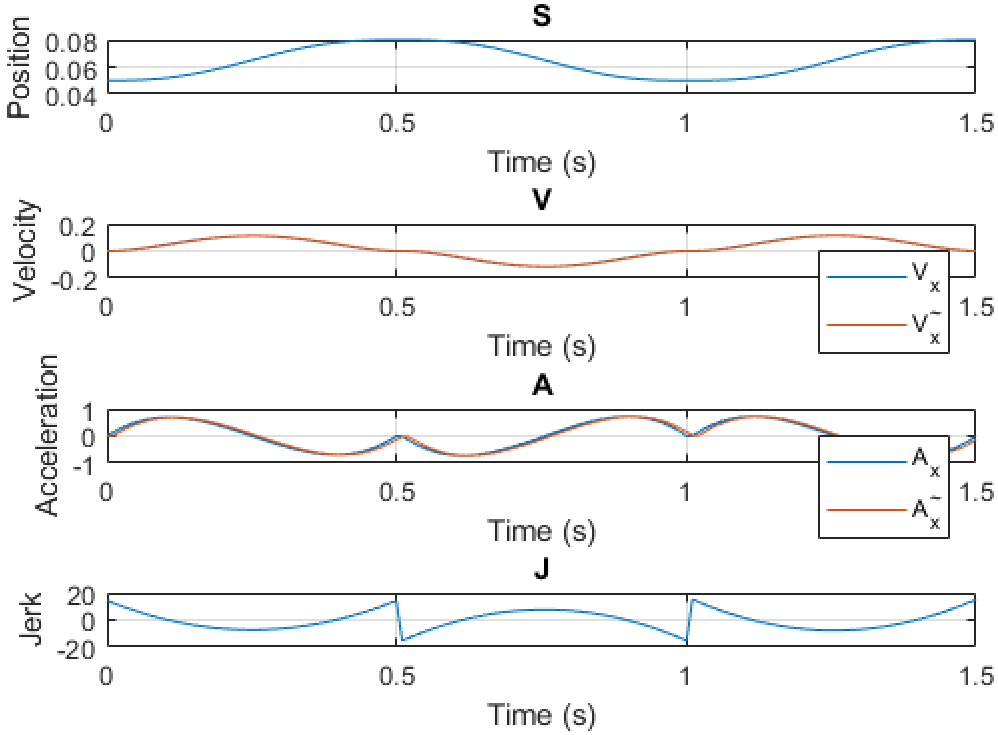


Figure 2.6: Sliding Motion Primitive Desired Trajectory

Figure 2.6 is the desired trajectory curve for the sliding motion primitive. The object slides from 0.05 m to 0.08 m then back again to 0.05 m complete its motion. The velocity and acceleration graphs have both a blue and an orange line. The blue line is the desired or planned trajectory. The orange line is the approximation given from the position S. The finite difference of the positions were taken to calculate the velocities and then the same was done to those velocities to calculate the corresponding accelerations. From this point the position values are then used when solving for the finger parameters. The position of the finger is calculated based on the contact position and the position of the object. The inverse kinematics is solved and the joint angles are calculated for each instance. Then the finite difference can be taken of the positions of the centers of masses and the joint angles to solve for the linear and angular velocities and accelerations for the finger.

The linear program 2.53 can be solved for each of the steps of the trajectory. The following figures are the results of the motion primitive.

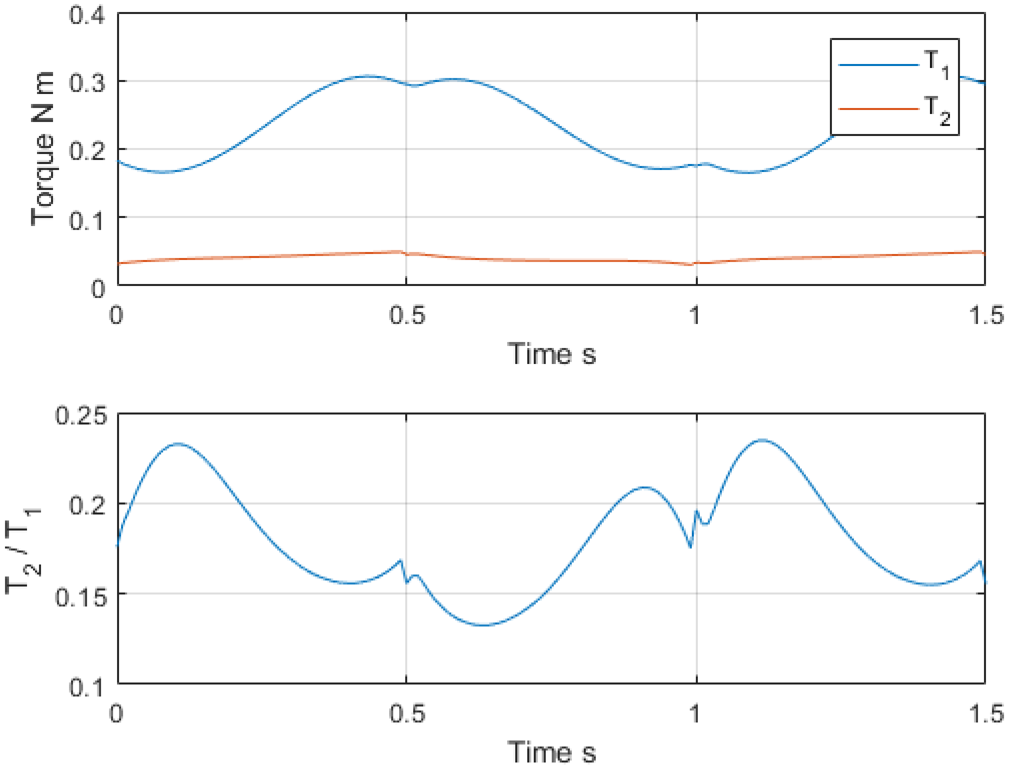


Figure 2.7: Sliding Motion Primitive Torque and Transmission Coefficient

Figure 2.7 shows the resulting torque and the calculated transmission coefficient c. The value of c being a low value is promising meaning that a large difference between the two torques is not required for this motion.

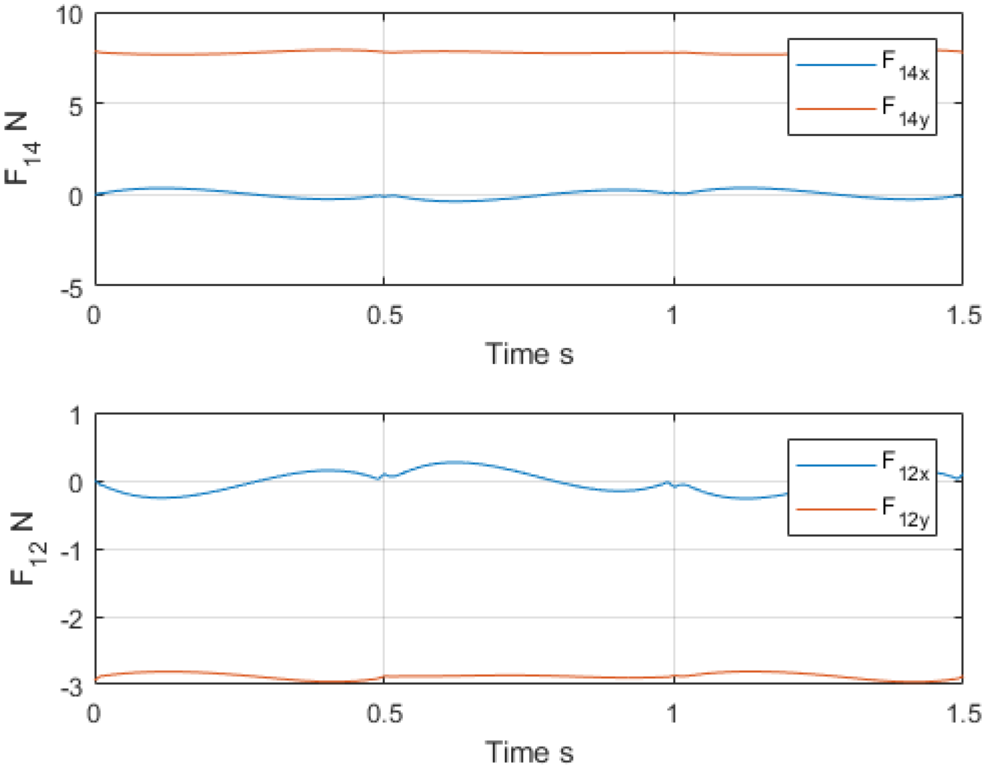


Figure 2.8: Sliding Motion Primitive Link 1 Forces

Figure 2.8 shows the forces acting on the first link at the first and second joints. The following figure 2.9 is two plots of the forces at the contact between the finger and the object (F_{23}) and the object and the ground (F_{34}). The directions of the y-components of forces F_{23} and F_{34} are both strictly positive. Ensuring that F_{23} is positive ensures the finger cannot pull on the block without the use of friction requiring contact to be maintained. Their directions can be verified in 2.4. The friction forces appose the direction of motion so their directions are determined by the opposite sign of the velocity.

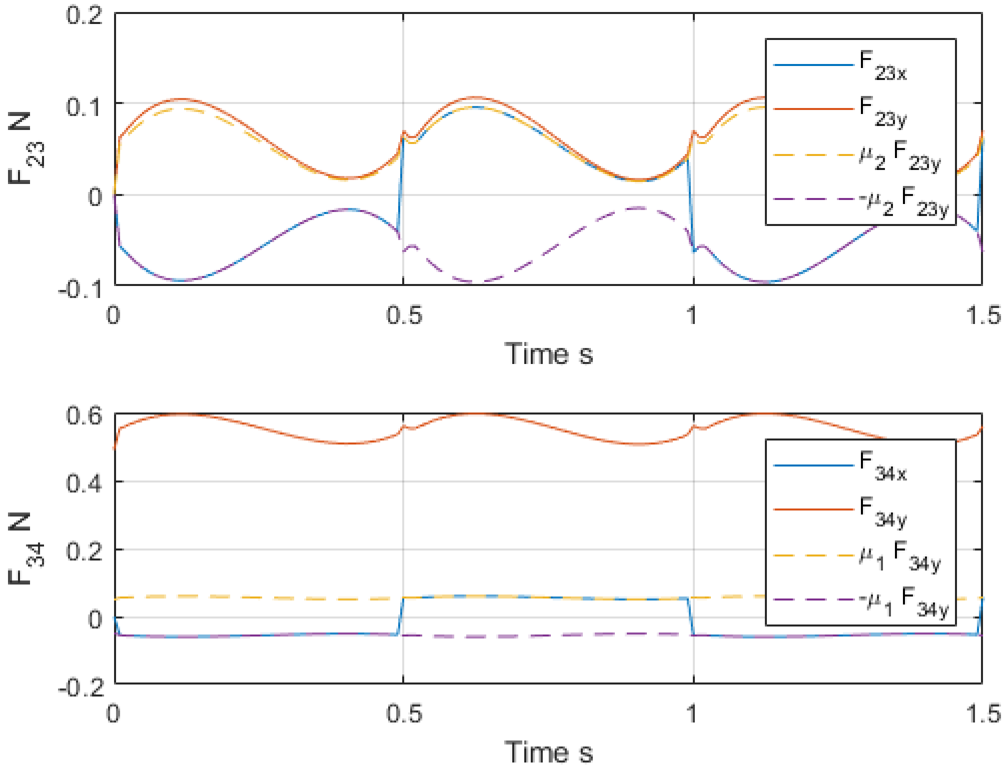


Figure 2.9: Sliding Motion Primitive Links 2 and 3 Forces

The following figure 2.10 verifies that the linear program has maintained the friction cone constraints. Forces that lie within the friction cone will not cause sliding at the contact point between the finger and the object. The contact between the object and the block in this method is a joint thus forces within the friction cone will maintain the no-slip condition.

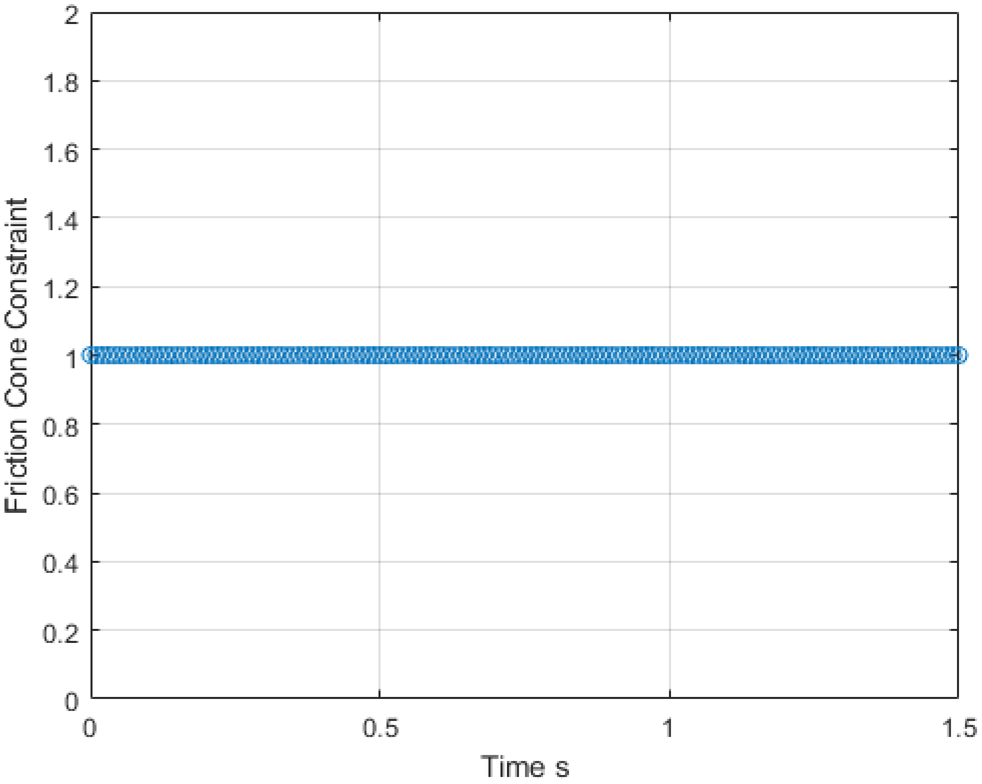


Figure 2.10: Sliding Motion Primitive Friction Cone Constraint

Figures 2.11 , 2.12 and 2.13 are plots of the linear and angular accelerations of the links and object during the motion primitive. It can be visually verified from figure 2.13 that the object comes to a stop during the desired time intervals due to the velocity and acceleration being zero. The y-components of the velocities are plotted and can be verified to be zero as well.

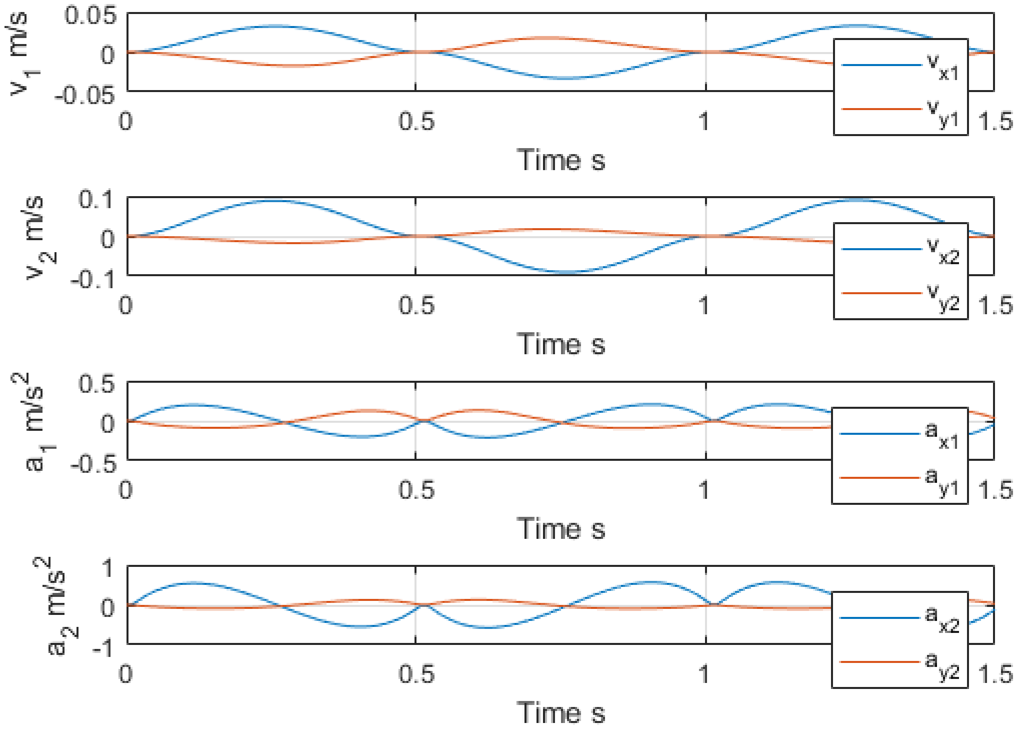


Figure 2.11: Sliding Motion Primitive Links 1 and 2 Linear Velocity and Accelerations

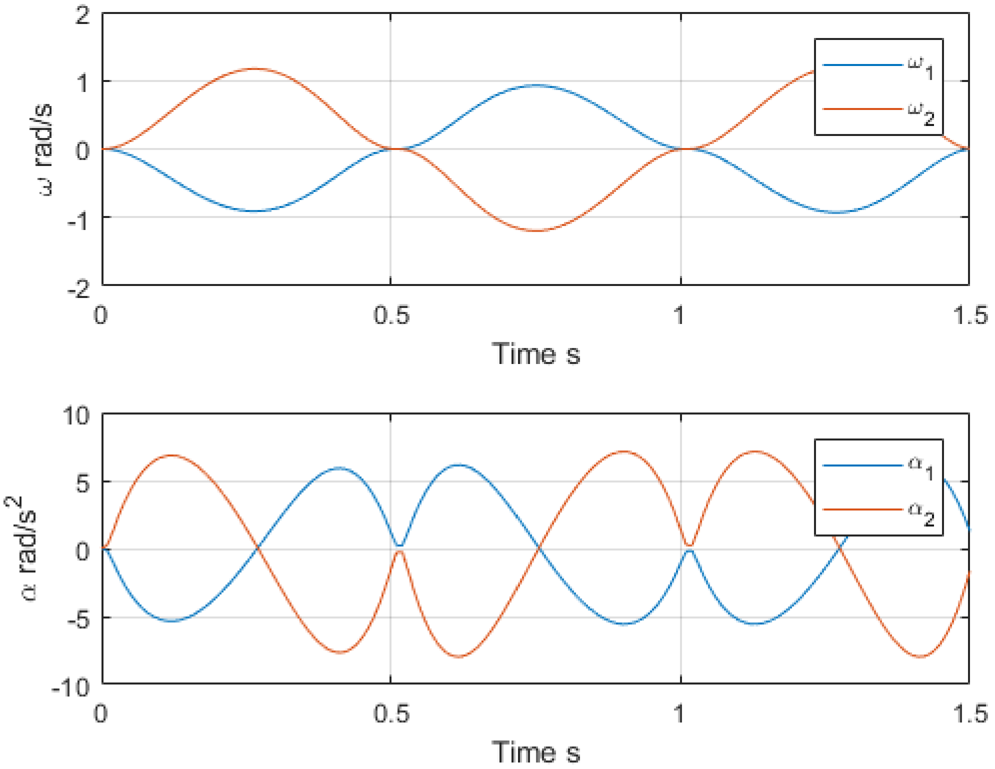


Figure 2.12: Sliding Motion Primitive Links 1 and 2 Angular Velocity and Accelerations

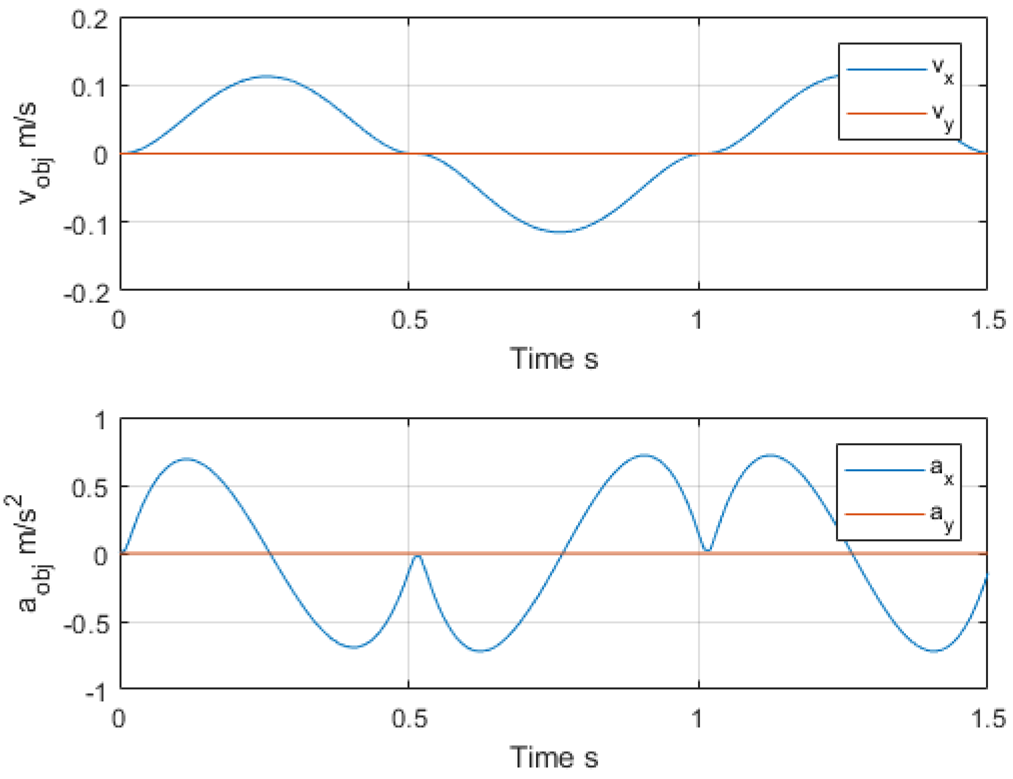


Figure 2.13: Sliding Motion Primitive Object Linear Velocity and Accelerations

2.3.2 Rolling Motion Primitive

An example of the *Rolling Motion Primitive* will be developed in this section. The rolling motion is similar to sliding because in both cases the object translates across a floor thus having no y-component of velocities or accelerations. However, there are some differences due to assumptions of the type of contact. The object is now rolling instead of sliding. For rolling, there is a relationship between the rolling angle and the distance traveled. The assumption used is a no-slip condition between the object and the ground. This means that there is no-slip between the object and the ground at the point of contact or that the velocity at the point of contact (on the object) is zero. The linear motion of a wheel rolling without slip is given by this equation.

$$d = R\theta \tag{2.54}$$

d is the distance traveled given the radius of the wheel R and θ is difference between the starting rotation angle and the ending rotation angle. This formula is exactly the same one as the arc length formula. Due to rolling without slip, each point on the wheel makes contact with the ground. The distance traveled is essentially the arc length of the given wheel.

Table 2.2 shows the parameters used in this example.

Variables	**Values**	**Units**
Link 1 Length	0.08	m
Link 2 Length	0.05	m
Link 1 Mass	0.5	kg
Link 2 Mass	0.3	kg
Link 1 Moment of Inertia	$2.667 * 10^{-4}$	kg m^2
Link 2 Moment of Inertia	$6.25 * 10^{-5}$	kg m^2
Object Mass	0.05	kg
Object Radius	0.02	m
Object Moment of Inertia	$1 * 10^{-5}$	kg m^2
Finger Object Coefficient of Friction	.9	
Object Ground Coefficient of Friction	.1	
Acceleration of Gravity	9.80665	$\frac{m}{s^2}$
Gravity Direction	$\frac{3\pi}{2}$	radians
Force of Gravity x - direction	$-1.8015 * 10^{-15}$	$\frac{m}{s^2}$
Force of Gravity y - direction	-9.8066	$\frac{m}{s^2}$
Global Time Step	0.01	s
Finger Contact Position wrt Object Frame	[0 0.02]	m
Motion Primitive Time Steps	[0.5 0.5]	s
Motion Primitive Desired x - Coordinates	[0.08 0.075 0.08]	m

Table 2.2: Rolling Motion Primitive Parameters

The moments of inertias of the links were calculated as rectangles in the same manner as in the previous section. The object is assumed to be circular (or cylindrical) so the inertia of a disk was used and can be calculated as follows.

$$I_{disk} = \frac{1}{2}mr^2 \tag{2.55}$$

Figure 2.14 shows the free body diagram for the system. A key difference between the sliding motion primitive is the ground friction is now acting at the contact point thus exerting a torque about the object's center of gravity.

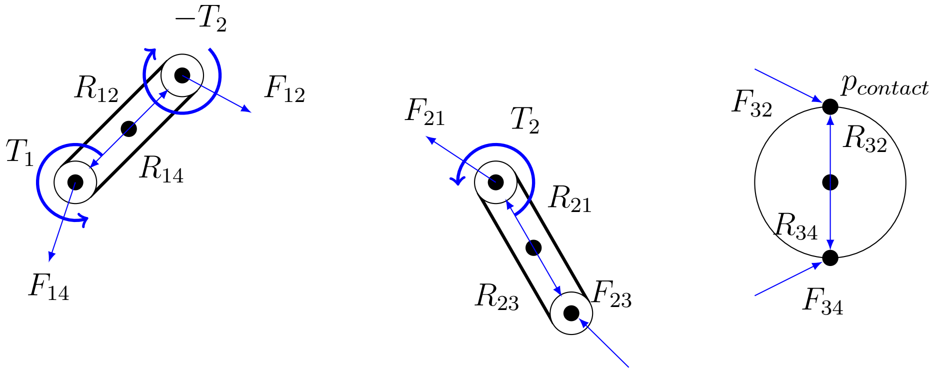


Figure 2.14: Rolling Free Body Diagrams

The forces can now be formulated into the a linear system of equations. The ground reaction torque due on the forces applied by the finger must now be considered which means there are a total of 9 equations.

Link 1

$$F_{14x} + F_{12x} = m_1 a_{1x} - F_{g1x} \tag{2.56}$$

$$F_{14y} + F_{12y} = m_1 a_{1y} - F_{g1y} \tag{2.57}$$

$$T_1 + (R_{14x}F_{14y} - R_{14y}F_{14x}) + (R_{12y}F_{12y} - R_{12x}F_{12x}) - T_2 = I_1\alpha_1 \tag{2.58}$$

Link 2

$$F_{23x} - F_{12x} = m_2 a_{2x} - F_{g2x} \tag{2.59}$$

$$F_{23y} - F_{12y} = m_2 a_{2y} - F_{g2y} \tag{2.60}$$

$$T_2 + (R_{23y}F_{23y} - R_{23x}F_{23x}) - (R_{21x}F_{12y} - R_{21y}F_{12x}) = I_2\alpha_2 \tag{2.61}$$

Link 3

$$F_{34x} - F_{23x} = m_3 a_{3x} - F_{g3x} \tag{2.62}$$

$$F_{34y} - F_{23y} = -F_{g3y} \tag{2.63}$$

$$(R_{34x} - R_{34y} sign(v_{3x})\mu_1)F_{34y} - (R_{32x}F_{23y} - R_{32y}F_{23x}) = I_3\alpha_3 \tag{2.64}$$

The equations can be formulated into their corresponding matrices. The coefficients become matrix A_{eq}, the variables are formed into vector x and the constants are lumped into vector b_{eq}. The equations can then be formed into a linear program and the minimal torques for each joints can be found.

$$A_{eq} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -R_{14y} & R_{14x} & -R_{12y} & R_{12x} & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_{21y} & -R_{21x} & -R_{23y} & R_{23x} & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_{32y} & -R_{32x} & -R_{34y} & R_{34x} & 0 & 0 \end{bmatrix} \tag{2.65}$$

$$x = \begin{bmatrix} F_{14x} \\ F_{14y} \\ F_{12x} \\ F_{12y} \\ F_{23x} \\ F_{23y} \\ F_{34x} \\ F_{34y} \\ T_1 \\ T_2 \end{bmatrix} \tag{2.66}$$

$$b_{eq} = \begin{bmatrix} m_1 a_{1x} - F_{g1x} \\ m_1 a_{1y} - F_{g1y} \\ I_1 \alpha_1 \\ m_2 a_{2x} - F_{g2x} \\ m_2 a_{2y} - F_{g2y} \\ I_2 \alpha_2 \\ m_3 a_{3x} - F_{g3x} \\ -F_{g3y} \\ I_3 \alpha_3 \end{bmatrix} \tag{2.67}$$

The y-component of the acceleration has been set to 0 in order to restrict the rolling motion solutions to a motion which keeps the object in contact with the ground. The linear program is written as follows

$$
\begin{aligned}
\text{minimize} \quad & \tau \\
\text{subject to} \quad & A_{eq}x = b_{eq} \\
& F_{23x} - \mu_2 F_{23y} && \leq 0 \\
& -F_{23x} - \mu_2 F_{23y} && \leq 0 \\
& F_{34x} - \mu_1 F_{34y} && \leq 0 \\
& -F_{34x} - \mu_1 F_{34y} && \leq 0 \\
& F_{23y} && \geq 0 \\
& F_{34y} && \geq 0
\end{aligned}
\tag{2.68}
$$

where the dynamics are entered as an equality constraint, $A_{eq}x = b_{eq}$, and the 2D friction cone constraint, along with the upper and lower bounds, are written as inequality constraints.

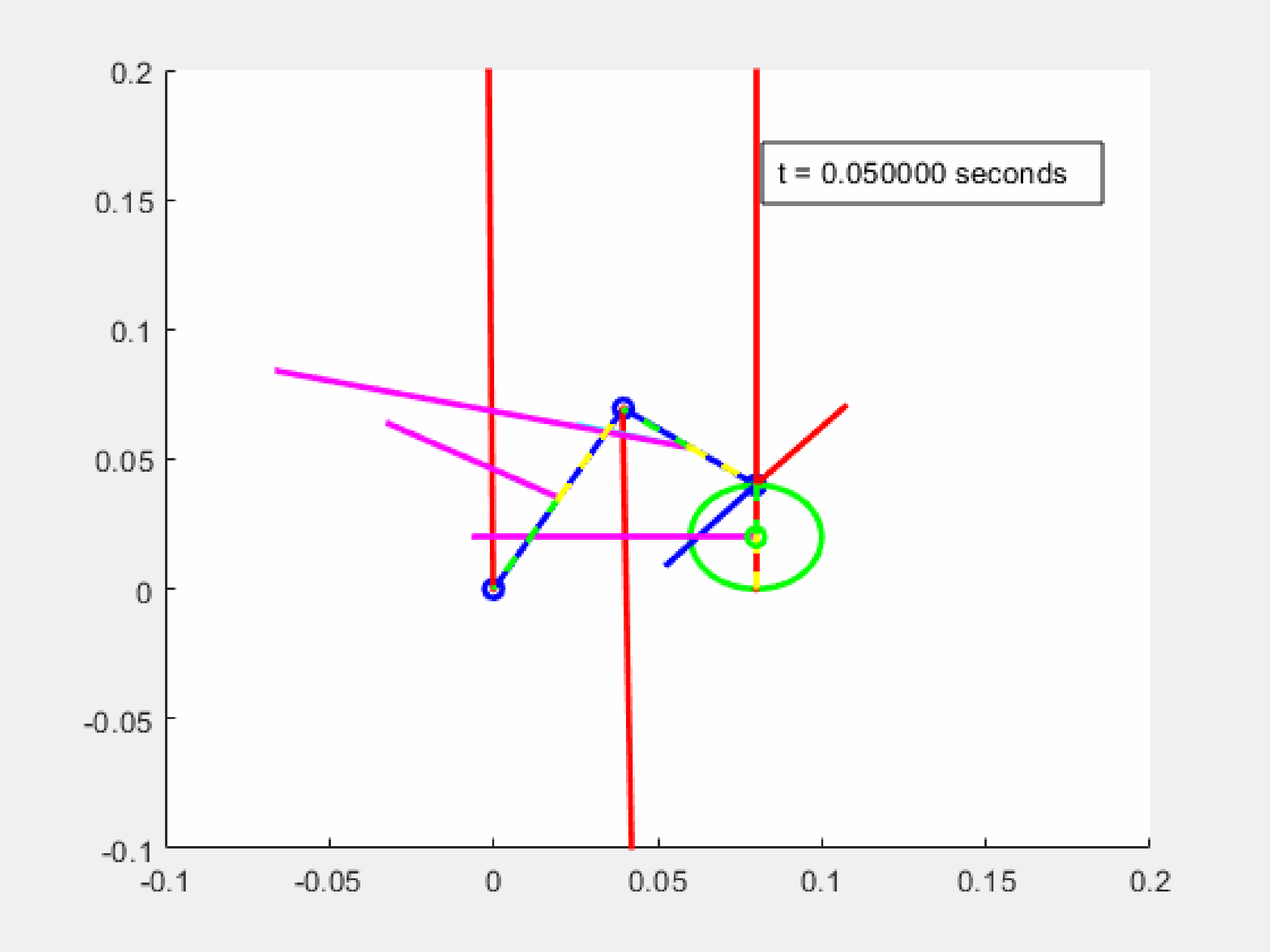


Figure 2.15: Rolling Motion Primitive

Figure 2.16 shows the designed trajectory for the rolling motion primitive. The finite difference was used to approximate the derivatives which are used in the linear program.

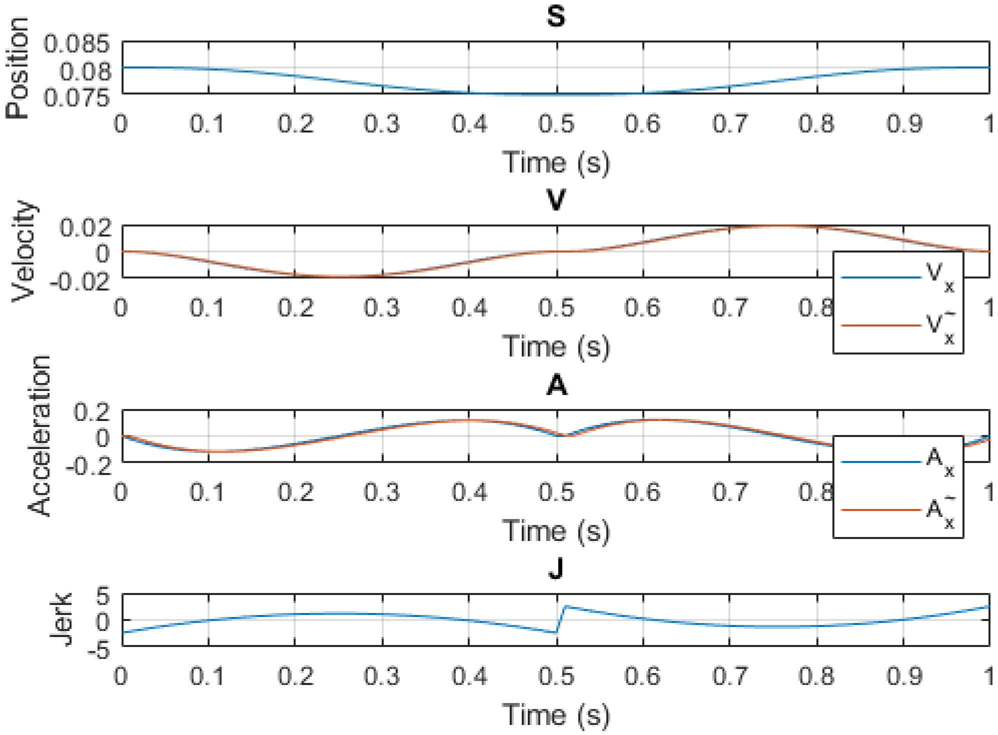


Figure 2.16: Rolling Motion Primitive Desired Trajectory

Figure 2.17 shows the torques at the first and second joints. The second plot shows the relationship between the two torques $\frac{T_2}{T_1}$ which is the transmission coefficient c.

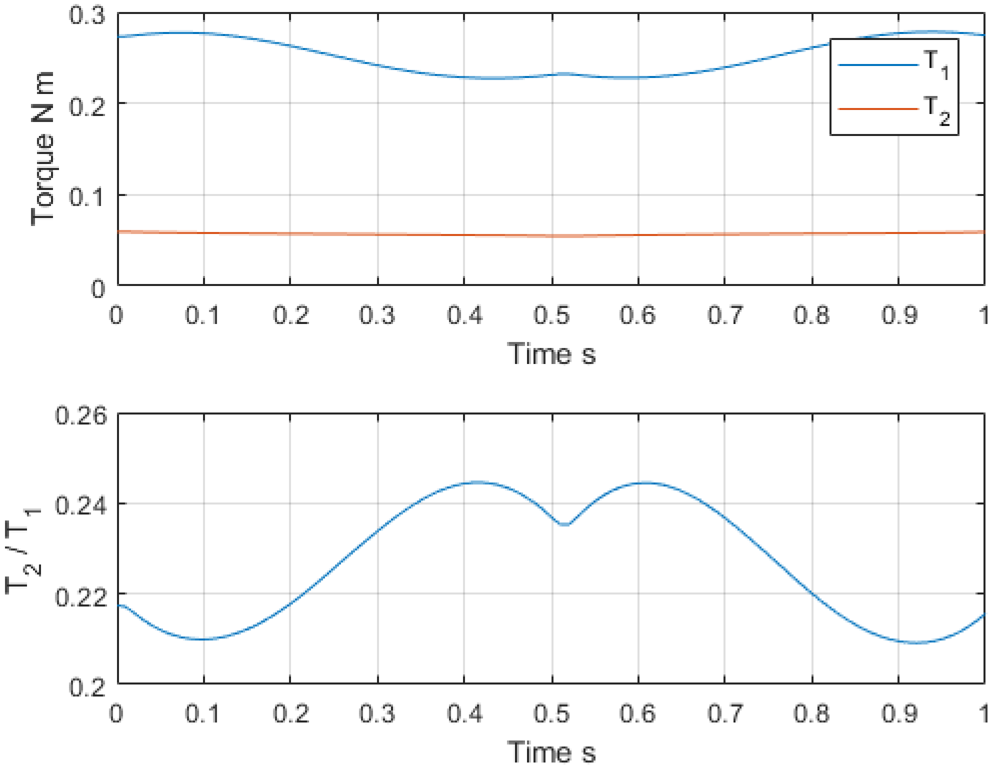


Figure 2.17: Rolling Motion Primitive Torque and Transmission Coefficient

Figures 2.18 2.19 show the x and y components at the joints, the contact between the finger and the object and the object and the ground.

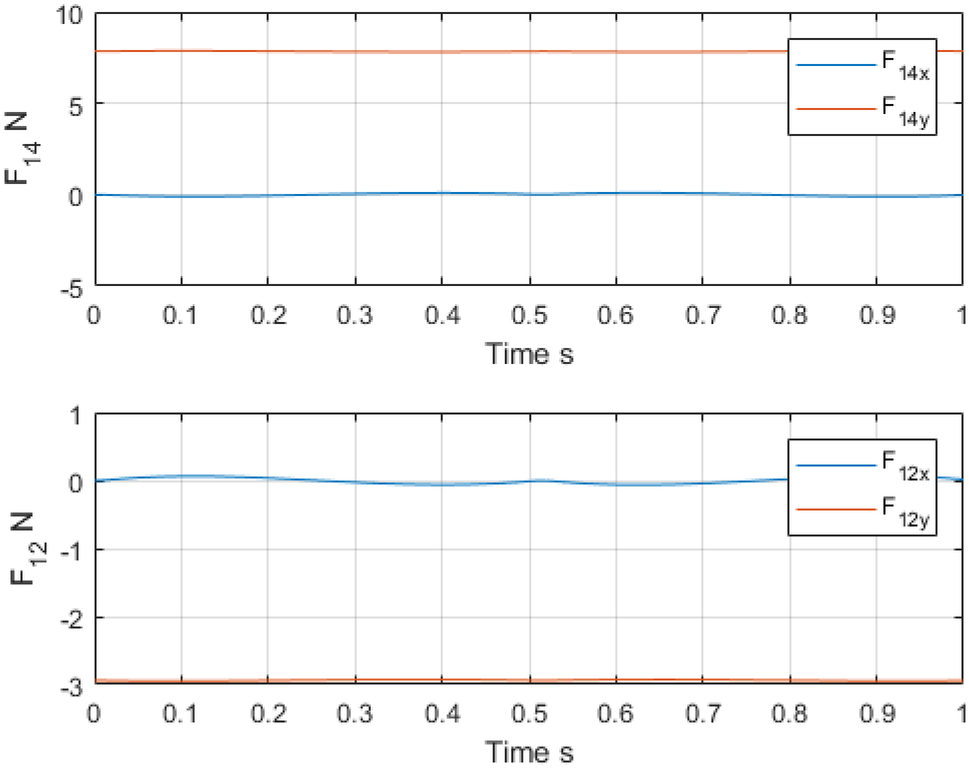


Figure 2.18: Rolling Motion Primitive Link 1 Forces

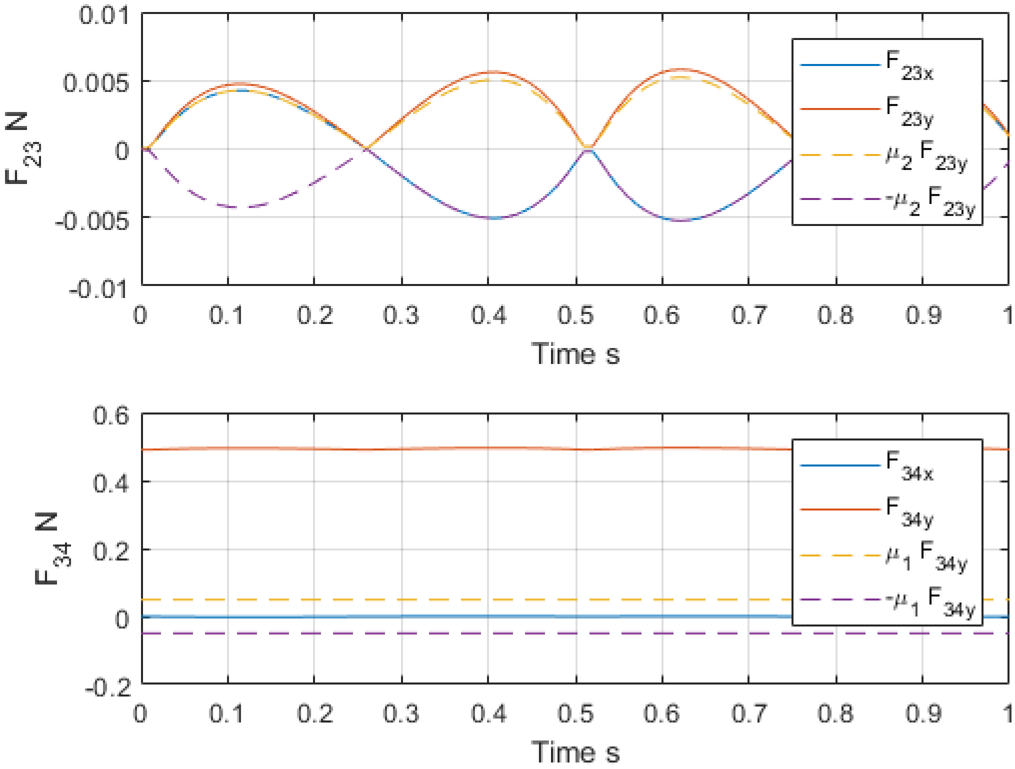


Figure 2.19: Rolling Motion Primitive Links 2 and 3 Forces

Figure 2.20 checks to ensure that the calculated forces satisfy the friction cone constraints. The values are binary either true or false.

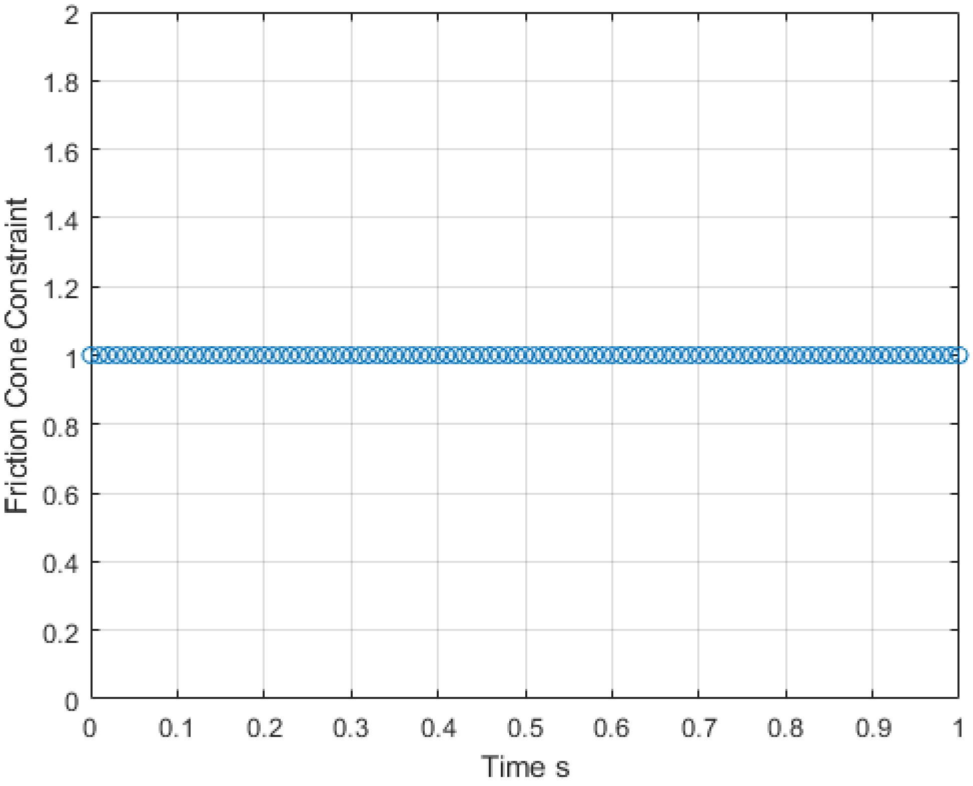


Figure 2.20: Rolling Motion Primitive Friction Cone Constraint

Figures 2.21 2.22 2.22 2.23 show the kinematic data corresponding to the force and torque values from above.

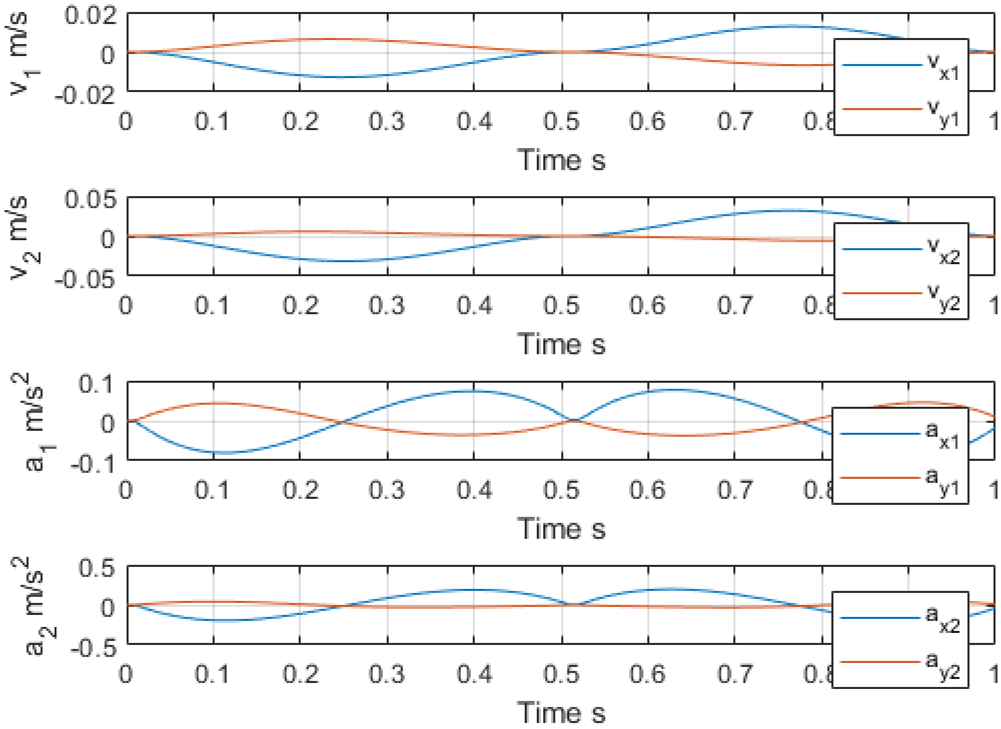


Figure 2.21: Rolling Motion Primitive Linear Velocity and Accelerations for Links 1 and 2

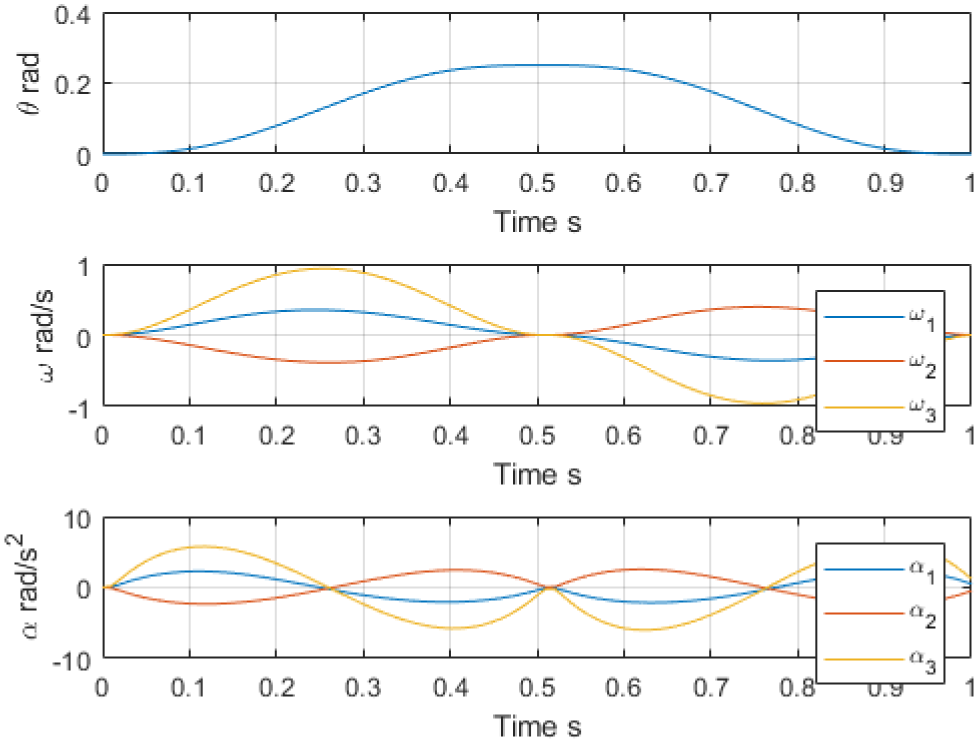


Figure 2.22: Rolling Motion Primitive Angular Velocity and Accelerations for Links 1 and 2

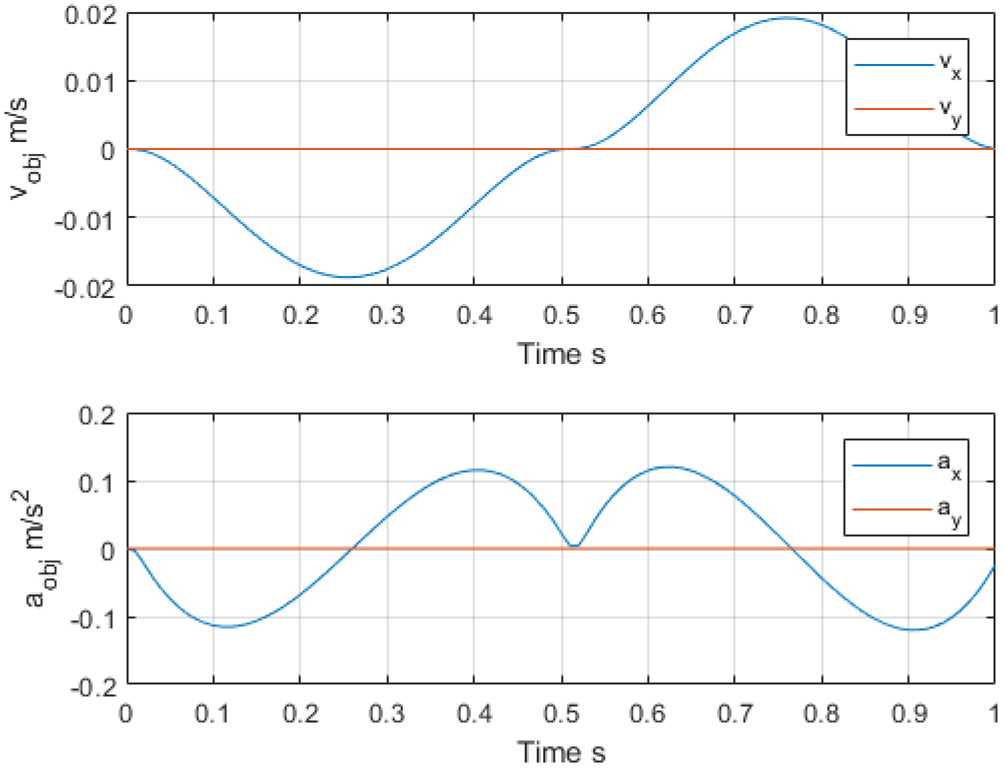


Figure 2.23: Rolling Motion Primitive Linear Velocity and Accelerations for the Object

2.3.3 Tipping Motion Primitive

The *Tipping Motion Primitive* reorients an object by a rotation about a tipping point. This task is frequently used in IHM tasks and is quite difficult to model. The system can be approximated as a four bar linkage rotating about the tipping point.

Table 2.3 lists the parameters used and their values within this example.

Variables	**Values**	**Units**
Link 1 Length	0.08	m
Link 2 Length	0.05	m
Link 1 Mass	0.5	kg
Link 2 Mass	0.3	kg
Link 1 Moment of Inertia	$2.667 * 10^{-4}$	kg m^2
Link 2 Moment of Inertia	$6.25 * 10^{-5}$	kg m^2
Object Mass	0.05	kg
Object Height	0.03	m
Object Length	0.05	m
Object Moment of Inertia	$1.42 * 10^{-5}$	kg m^2
Finger Object Coefficient of Friction	.9	%
Object Ground Coefficient of Friction	.5	%
Acceleration of Gravity	9.80665	$\frac{m}{s^2}$
Gravity Direction	$\frac{3\pi}{2}$	radians
Force of Gravity x - direction	$-1.8015 * 10^{-15}$	$\frac{m}{s^2}$
Force of Gravity y - direction	-9.8066	$\frac{m}{s^2}$
Global Time Step	0.01	s
Finger Contact Position wrt Object Frame	[-0.025 0.03]	m
Motion Primitive Time Steps	[0.8 0.8]	s
Motion Primitive Desired x - Coordinates	[0 $\frac{pi}{4}$ 0]	radians

Table 2.3: Tipping Motion Primitive Parameters

The links' moments of inertia are calculated the same manner as the previous two examples. The object inertia is considered during tipping due to the object rotating and having an angular acceleration component. This can be calculated from the following equation.

$$I_{cube} = \frac{1}{12}m(l^2 + w^2) \tag{2.69}$$

Figure 2.24 shows the free body diagram of the links and the object. The contact point now for the tipping motion primitive is now the left most corner, the object is tipping towards the finger's base, and the reaction force and its radius must be considered.

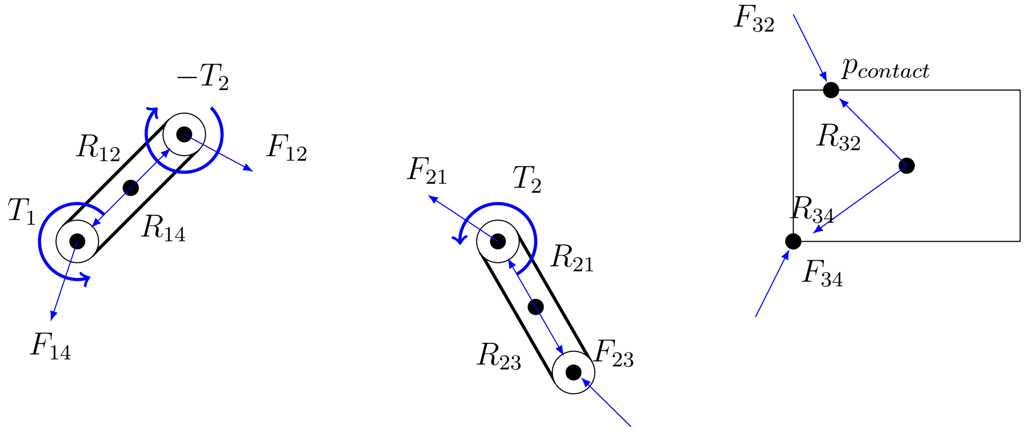


Figure 2.24: Tipping Free Body Diagrams

The equations from the free-body diagram above can be written as follows.

Link 1

$$F_{14x} + F_{12x} = m_1 a_{1x} - F_{g1x} \tag{2.70}$$

$$F_{14y} + F_{12y} = m_1 a_{1y} - F_{g1y} \tag{2.71}$$

$$T_1 + (R_{14x}F_{14y} - R_{14y}F_{14x}) + (R_{12y}F_{12y} - R_{12x}F_{12x}) - T_2 = I_1\alpha_1 \tag{2.72}$$

Link 2

$$F_{23x} - F_{12x} = m_2 a_{2x} - F_{g2x} \tag{2.73}$$

$$F_{23y} - F_{12y} = m_2 a_{2y} - F_{g2y} \tag{2.74}$$

$$T_2 + (R_{23y}F_{23y} - R_{23x}F_{23x}) - (R_{21x}F_{12y} - R_{21y}F_{12x}) = I_2\alpha_2 \tag{2.75}$$

Link 3

$$F_{34x} - F_{23x} = m_3 a_{3x} - F_{g3x} \tag{2.76}$$

$$F_{34y} - F_{23y} = m_3 a_{3y} - F_{g3y} \tag{2.77}$$

$$(R_{34x}F_{34y} - R_{34y}F_{34x}) - (R_{32x}F_{23y} - R_{32y}F_{23x}) = I_3\alpha_3 \tag{2.78}$$

The equations can now be formed into linear set of equations and a linear optimization program can be written in order to minimize the torques required to perform the motion.

$$A_{eq} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -R_{14y} & R_{14x} & -R_{12y} & R_{12x} & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_{21y} & -R_{21x} & -R_{23y} & R_{23x} & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_{32y} & -R_{32x} & -R_{34y} & R_{34x} & 0 & 0 \end{bmatrix} \tag{2.79}$$

$$x = \begin{bmatrix} F_{14x} \\ F_{14y} \\ F_{12x} \\ F_{12y} \\ F_{23x} \\ F_{23y} \\ F_{34x} \\ F_{34y} \\ T_1 \\ T_2 \end{bmatrix} \tag{2.80}$$

$$b_{eq} = \begin{bmatrix} m_1 a_{1x} - F_{g1x} \\ m_1 a_{1y} - F_{g1y} \\ I_1 \alpha_1 \\ m_2 a_{2x} - F_{g2x} \\ m_2 a_{2y} - F_{g2y} \\ I_2 \alpha_2 \\ m_3 a_{3x} - F_{g3x} \\ m_2 a_{3y} - F_{g3y} \\ I_3 \alpha_3 \end{bmatrix} \tag{2.81}$$

$$\begin{aligned} \text{minimize} \quad & \tau \\ \text{subject to} \quad & A_{eq} x = b_{eq} \\ & F_{23x} - \mu_2 F_{23y} && \leq 0 \\ & -F_{23x} - \mu_2 F_{23y} && \leq 0 \\ & F_{34x} - \mu_1 F_{34y} && \leq 0 \\ & -F_{34x} - \mu_1 F_{34y} && \leq 0 \\ & F_{23y} && \geq 0 \\ & F_{34y} && \geq 0 \end{aligned} \tag{2.82}$$

A simulation of the motion primitive was performed to verify the dynamics of the motion primitive.

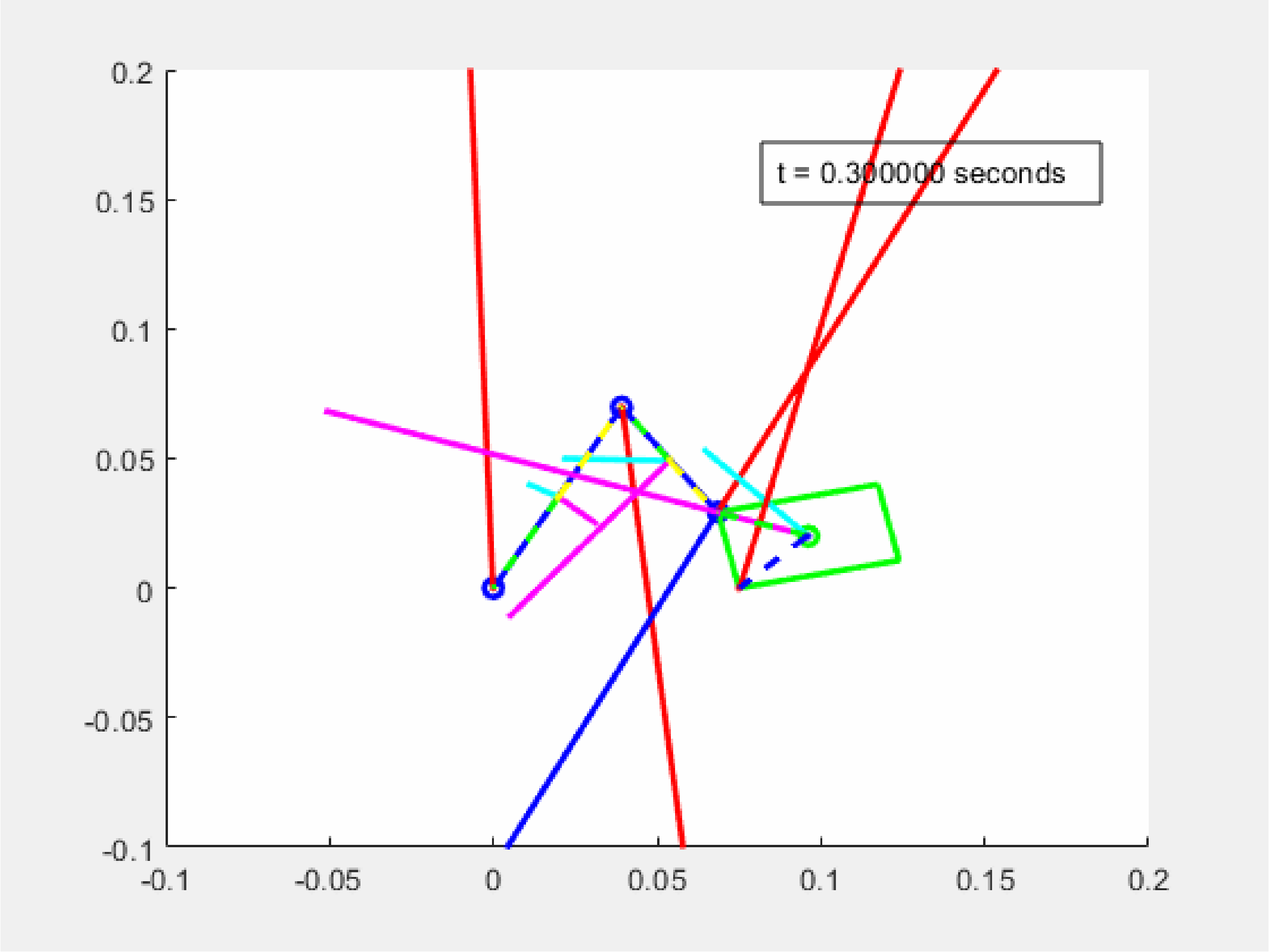


Figure 2.25: Tipping Motion Primitive

Figure 2.26 shows the desired motion of the object. For the tipping motion primitive, the desired trajectory is slightly different than in the previous two motion primitives. The boundary conditions are angle of tilt from the horizontal of the frame attached to the object at its center of gravity.

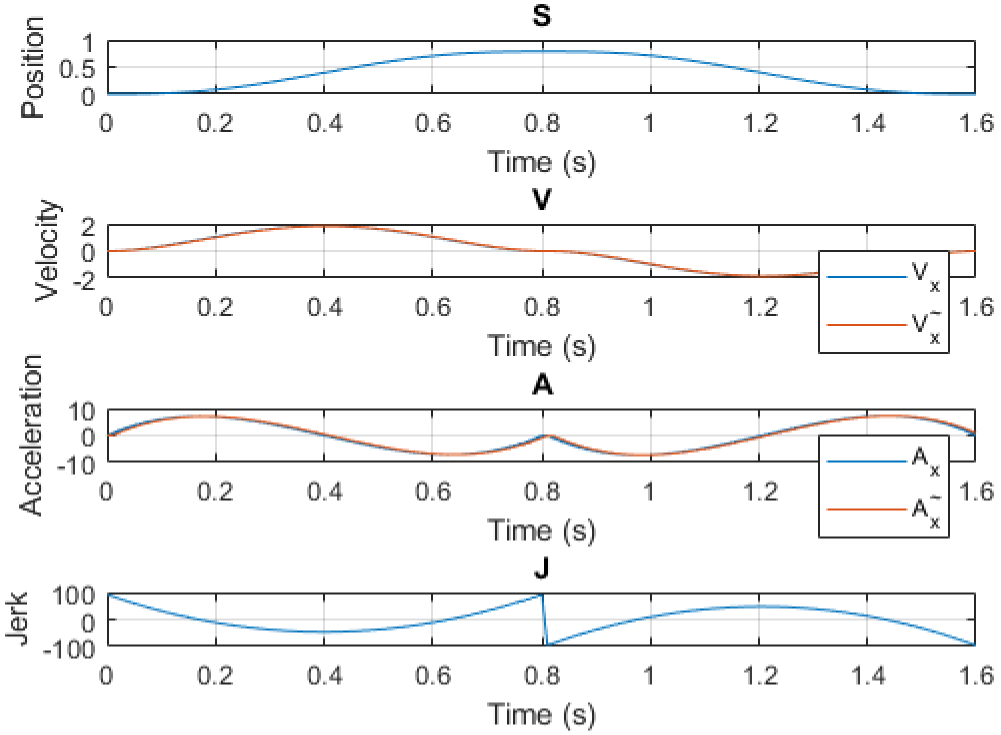


Figure 2.26: Tipping Motion Primitive Desired Trajectory

Figure 2.27 plots the torques and respective transmission coefficient for the tipping motion primitive.

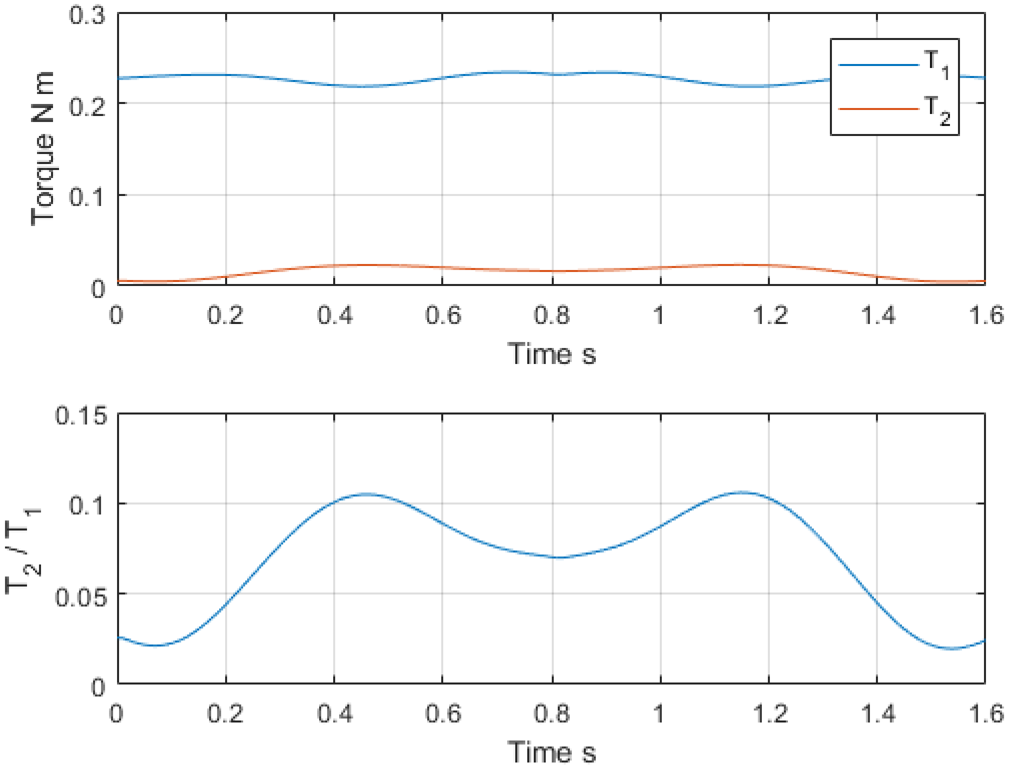


Figure 2.27: Tipping Motion Primitive Torque and Transmission Coefficient

Figures 2.28 and 2.29 show the resulting forces at the joints of the finger as well as the contact between the finger and the object and the object and the ground. Visually the friction cone constraints can be seen to be satisfied at each step. It is vital that the there is no slip at either of the contacts for the tipping motion primitive. Ensuring the no slip condition at the tipping point is necessary to ensure the object does not move unpredictably during the motion due to sliding.

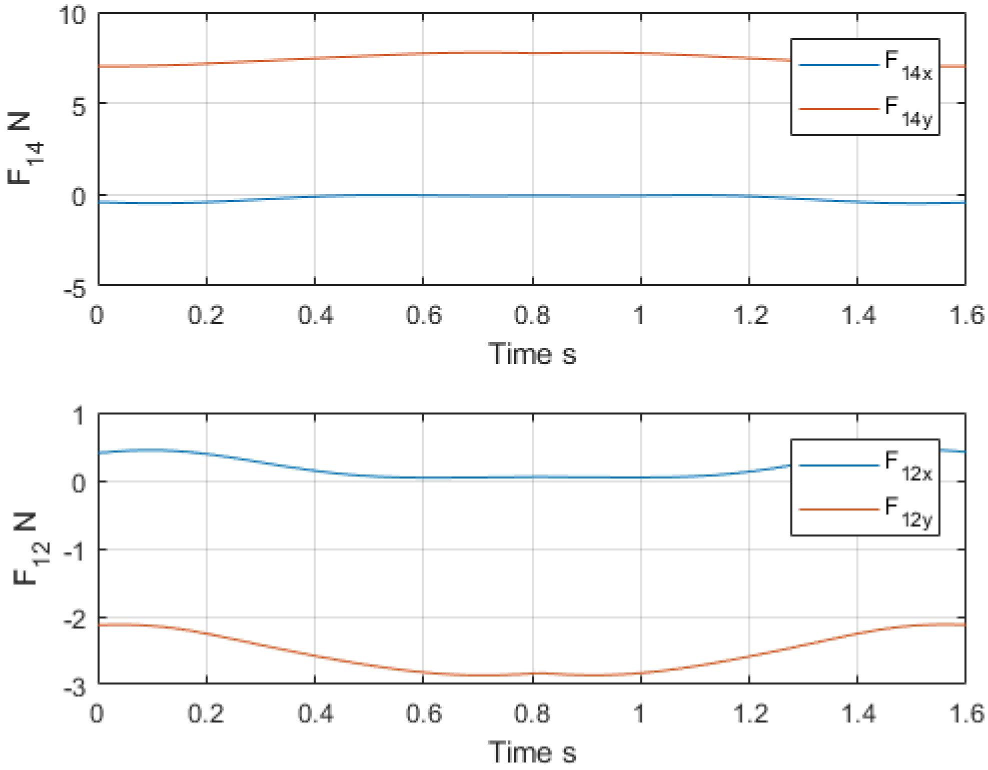


Figure 2.28: Tipping Motion Primitive Link 1 Forces

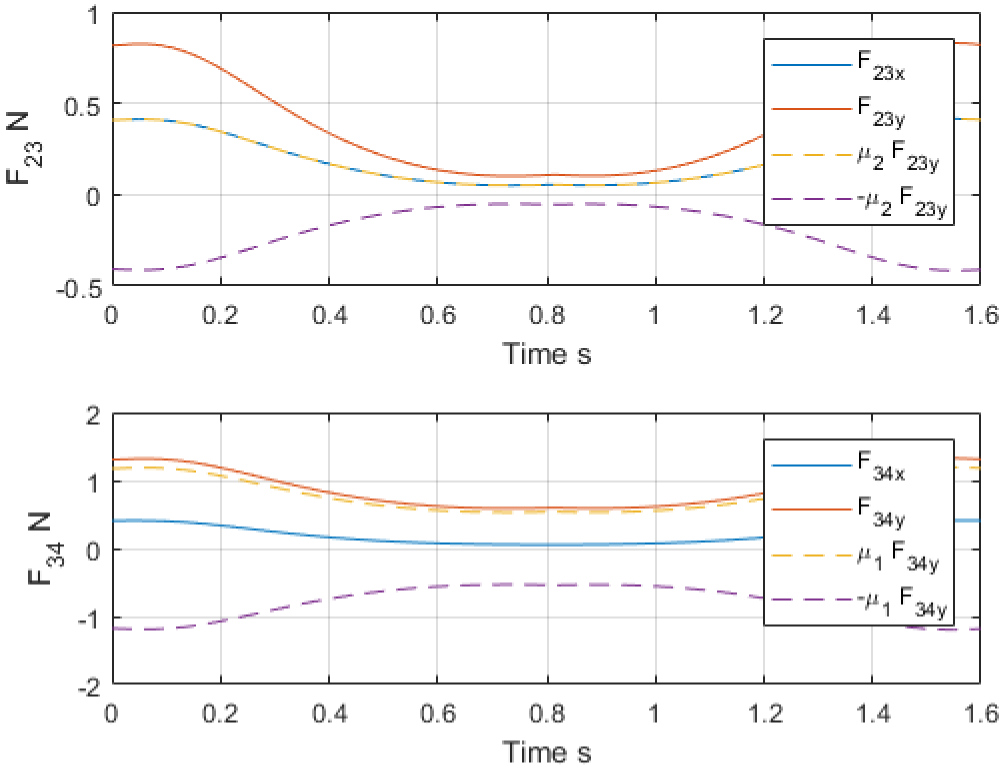


Figure 2.29: Tipping Motion Primitive Links 2 and 3 Forces

Figure 2.30 shows, per step, if the calculated forces for the desired motion satisfy the friction cone constraints.

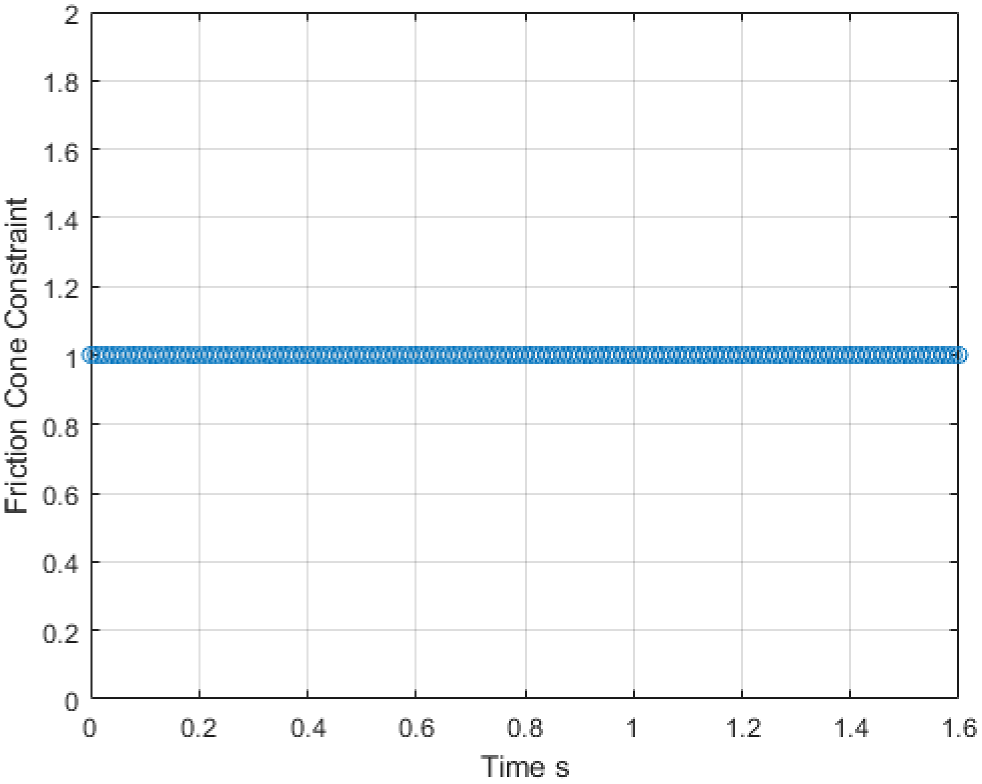


Figure 2.30: Tipping Motion Primitive Friction Cone Constraint

Figures 2.31, 2.32, 2.32 and 2.33 show the kinematic data corresponding to the force and torque values from the solution above.

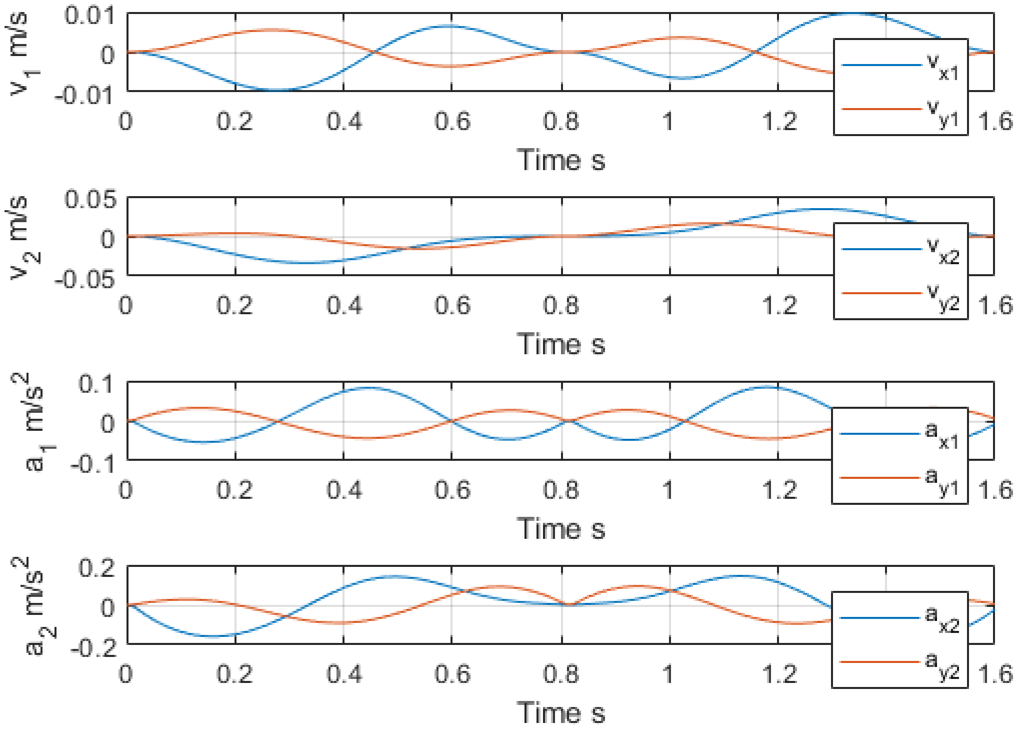


Figure 2.31: Tipping Motion Primitive Links 1 and 2 Linear Velocity and Accelerations

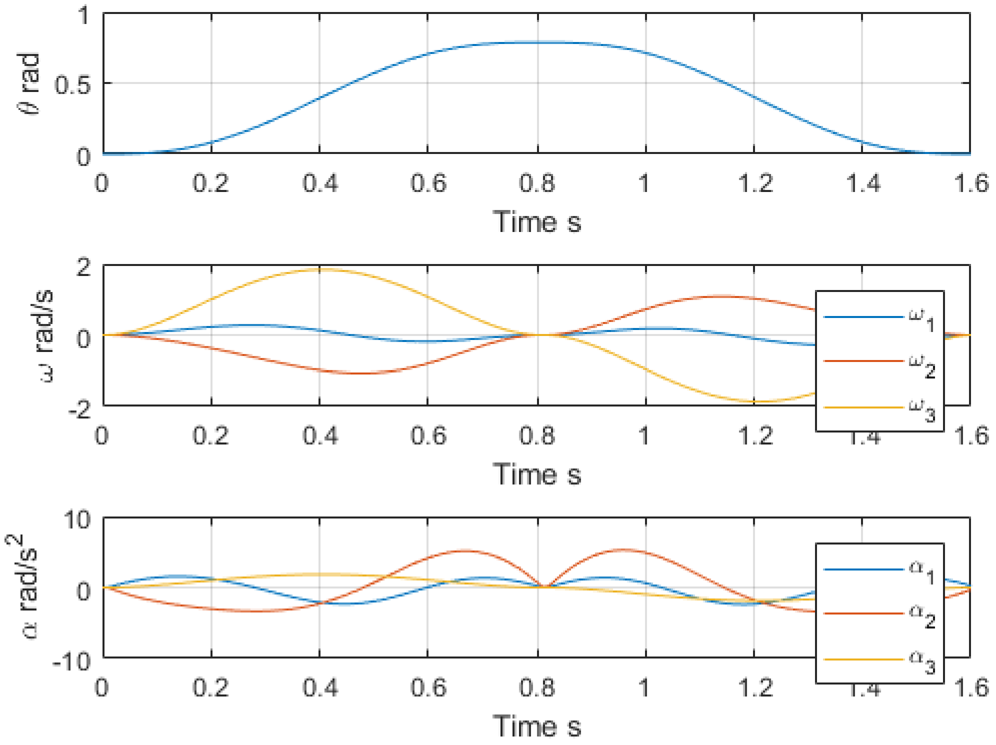


Figure 2.32: Tipping Motion Primitive Links 1 and 2 Angular Velocity and Accelerations

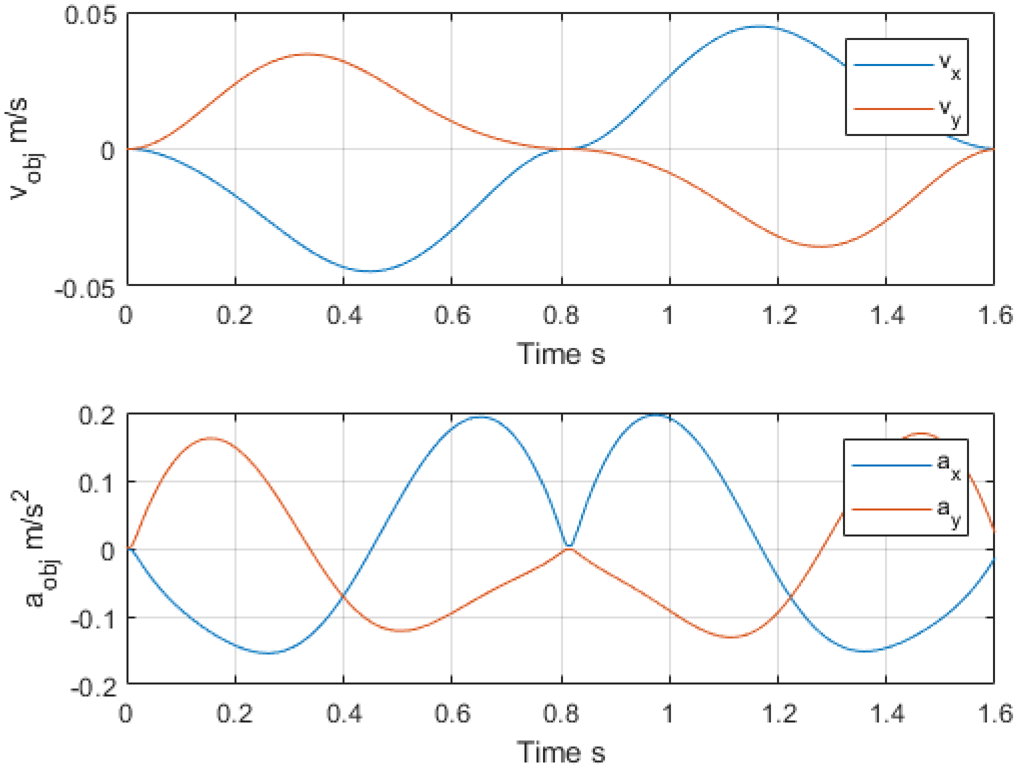


Figure 2.33: Tipping Motion Primitive Object Linear Velocity and Accelerations

Chapter 3

In-Hand Manipulation Model Feedback Control Loop

This chapter describes a method to utilize the motion primitives in a feedback control loop. In any real-life system, there are errors present which requires active control to provide corrections. Ideally, the theory and code can be written and tested on hardware. However, as shown in this chapter, dynamic simulations to model real-life systems can be used as a valid stepping stone. In this paper [89], a formulation is presented to model line and surface contact with point contacts. Their results were shown to be consistent with physics-based models of surface and line contacts. Utilizing the dynamic simulation, motion primitives are planned and then the torques utilized to manipulate an object. The simulation then provides state feedback, as sensors would in a hardware implementation, and then a new plan is generated from that state. These corrections drive the object to the desired configuration and perform the desired motion primitive. This formulation is similar to a model predictive controller (MPC), where the behavior of a dynamic system can be optimized for a prediction horizon while satisfying a set of constraints. The main advantage of this method is that the MPC is able to anticipate future events and can take control actions accordingly. Other classical controllers, like PID controllers, do not have this predictive ability. The MPC performs its finite-horizon optimization each iteration cycle and provides the proper control signals in order to satisfy the constraints defined in the model optimization. The new states based on the plant feedback is supplied to the MPC and the prediction is performed repeated. The motion primitives developed in this work provide the dynamic model for the finger and the linear optimization program that is implemented can be used in an MPC online feedback control loop.

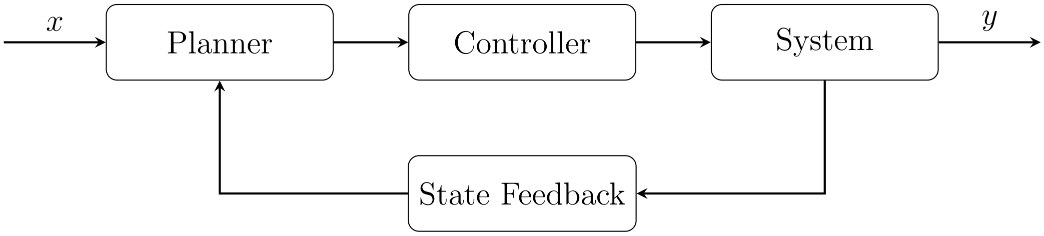


Figure 3.1: Closed-Loop In-Hand Manipulation Model Feedback Control Loop.

3.1 Rigid Body Dynamic Simulation with Line and Surface Contacts

The rigid body dynamic simulation with line and surface contacts will be explained in this section. The main goal of the paper [89] is the development of a method to consider line and surface contacts during dynamic simulations. Contact detection and dynamic simulation are simultaneously solved by forming the problem as a mixed nonlinear complementarity problem. The center of pressure, forces acting at the center of pressure, configurations and velocities of rigid objects can be determined from the mixed nonlinear complementarity problem. The authors form a geometrically implicit time-stepping scheme for dynamic simulation of contact between two rigid objects to include 2D line contact and 3D line/surface contact. The dynamic model is written by writing the Newton-Euler equations of motion, kinematic map, friction model and contact constraints.

The Newton-Euler equations can be written in this form

$$M(q)\dot{v} = W_n\lambda_n + W_t\lambda_t + W_o\lambda_o + W_r\lambda_r + \lambda_{app} + \lambda_{vp} \tag{3.1}$$

where $M(q) = \text{diag}(m, m, m, I')$ which is a 6 x 6 mass matrix. The inertia matrix in the body frame is $I = \text{diag}(I_x, I_y, I_z)$ while the inertia matrix in the world frame is $I' = RIR^T$. The rotation matrix ,R, is a 3 x 3 rotation matrix from body to world frame. The external forces and gravity, λ_{app}, is a 6 x 1 vector. The Coriolis and centripetal forces is represented in the 6 x 1 vector λ_{vp}. The wrenches of each of the normal contact forces are represented by W_n, W_t, W_o and W_r. Let (n, t, o) be unit vectors of the contact frame and r be the vector from the center of gravity to the contact point expressed in the world frame. The normal contact forces may be calculated as follows.

$$W_n = \begin{bmatrix} n \\ r \times n \end{bmatrix} \tag{3.2}$$

$$W_t = \begin{bmatrix} t \\ r \times t \end{bmatrix} \tag{3.3}$$

$$W_o = \begin{bmatrix} o \\ r \times o \end{bmatrix} \tag{3.4}$$

$$W_r = \begin{bmatrix} 0 \\ n \end{bmatrix} \tag{3.5}$$

To discretize of equation 3.1 a backward Euler time-stepping scheme is used. This can be written as follows.

$$M^l v^{l+1} = M^l v^l + W_n^{l+1} p_n^{l+1} + W_t^{l+1} p_t^{l+1} + W_o^{l+1} p_o^{l+1} + W_r^{l+1} p_r^{l+1} + p_{app}^l + p_{vp}^l \tag{3.6}$$

All of the applied forces have become impulses in this equation. Next, the kinematic map is defined as follows.

$$\dot{q} = G(q)v \tag{3.7}$$

Where G is the matrix mapping the generalized velocity of the body to the time derivative of the position and the orientation. The kinematic map is then discretized and can be written as follows.

$$\dot{q} = \frac{q^{l+1} - q^l}{h} \tag{3.8}$$

$$q^{l+1} = q^l + hGv^{l+1} \tag{3.9}$$

The friction model is then defined which is based on the maximum power dissipation principle. Using Coulomb's friction law the friction model can be written as follows.

$$\max \quad -(v_t p_t + v_o p_o + v_r p_r) \tag{3.10}$$

$$\text{s.t.} \quad (\frac{p_t}{e_t})^2 + (\frac{p_o}{e_o})^2 + (\frac{p_r}{e_r})^2 - \mu^2 p_n^2 \leq 0 \tag{3.11}$$

The velocities v_t and v_o are the tangential components of the relative velocity at the contact. The velocity v_r is the relative angular velocity about the normal at the contact. The variables e_t, e_o and e_r are given positive constants that define the friction ellipsoid. The coefficient of friction at the contact is given by μ. The model states that the possible contact forces and moments that lie within the friction ellipsoid maximize the power dissipation at the contact due to friction.

The normal contact constraint is then defined as

$$0 \leq \lambda_n \perp \psi_n(q,t) \geq 0 \tag{3.12}$$

where $\psi_n(q,t)$ is the gap function between two objects. If $\psi_n(q,t) \geq 0$ the objects separate. When $\psi_n(q,t) = 0$ the objects are touching. If $\psi_n(q,t) \leq 0$ the objects penetrate one another. The two objects are described by convex inequalities $f_i(x) \leq 0$ and $g_j(x \leq 0)$. The variables a_1^{l+1} and a_2^{l+1} are the position vectors of the contact points or closest points on the two objects at the end of the time step. Equation 3.12 can be rewritten into this form

$$a_1^{l+1} - a_2^{l+1} = -l_k^{l+1}[\nabla f_k(a_1^{l+1}) + \sum_{i=1, i\neq k}^{m} l_i^{l+1} \nabla f_i(a_1^{l+1})] \tag{3.13}$$

$$\nabla f_k(a_1^{l+1}) + \sum_{i=1, i\neq k}^{m} l_i^{l+1} \nabla f_i(a_1^{l+1}) = -\sum_{j=m+1}^{n} l_j^{l+1} \nabla g_j(a_2^{l+1}) \tag{3.14}$$

$$0 \leq l_i^{l+1} \perp -f_i(a_1^{l+1}) \geq 0 \qquad i = 1, \ldots, m \tag{3.15}$$

$$0 \leq l_j^{l+1} \perp -g_j(a_2^{l+1}) \geq 0 \qquad i = m+1, \ldots, n \tag{3.16}$$

$$0 \leq \lambda_n^{l+1} \perp \max_{i=1,\ldots,m}\{-f_i(a_2^{l+1})\} \geq 0 \tag{3.17}$$

where l_i^{l+1} is the Lagrange multipliers and the index k is the index of a active constraint on the body. If the equations above hold, the two objects will satisfy the contact constraints given in equation 3.12.

The work presented in [89] is presented here in brevity. Please refer to original paper to gain a deeper understanding of the inner workings of dynamic simulation. In the following section, the dynamic simulation described in this section is utilized in a feedback control loop with the motion primitive planning for underactuated fingers.

3.2 In-Hand Manipulation Model Feedback Control Loop Example

This section will overview the IHM model feedback control loop and provide an example using the sliding motion primitive. In order for the motion primitives to be useful they must be shown to be consistent with the dynamics of the system. Applying planned torques in the presence of errors will lead to undesired results so a feedback control loop was developed. Initially, it was thought by simply re-planning from the current step after the simulation had provided the state feedback would be enough to suffice but the planner needed another controller to ensure that the planned torque inputs to the system were as expected. This was due to an overshooting error caused by overfitting of the trajectory generation polynomials once the object was close to the goal position. This is due to the degree of the polynomials used to design the trajectories. An ad-hoc controller was developed to help deal with these issues. The controller applies the torques provided by the planner multiplied by a gain which takes into account the velocity of the block, direction of travel and it's error in the position. In the following subsection, the results are shown for the sliding motion primitive.

3.2.1 Sliding Feedback Control Loop

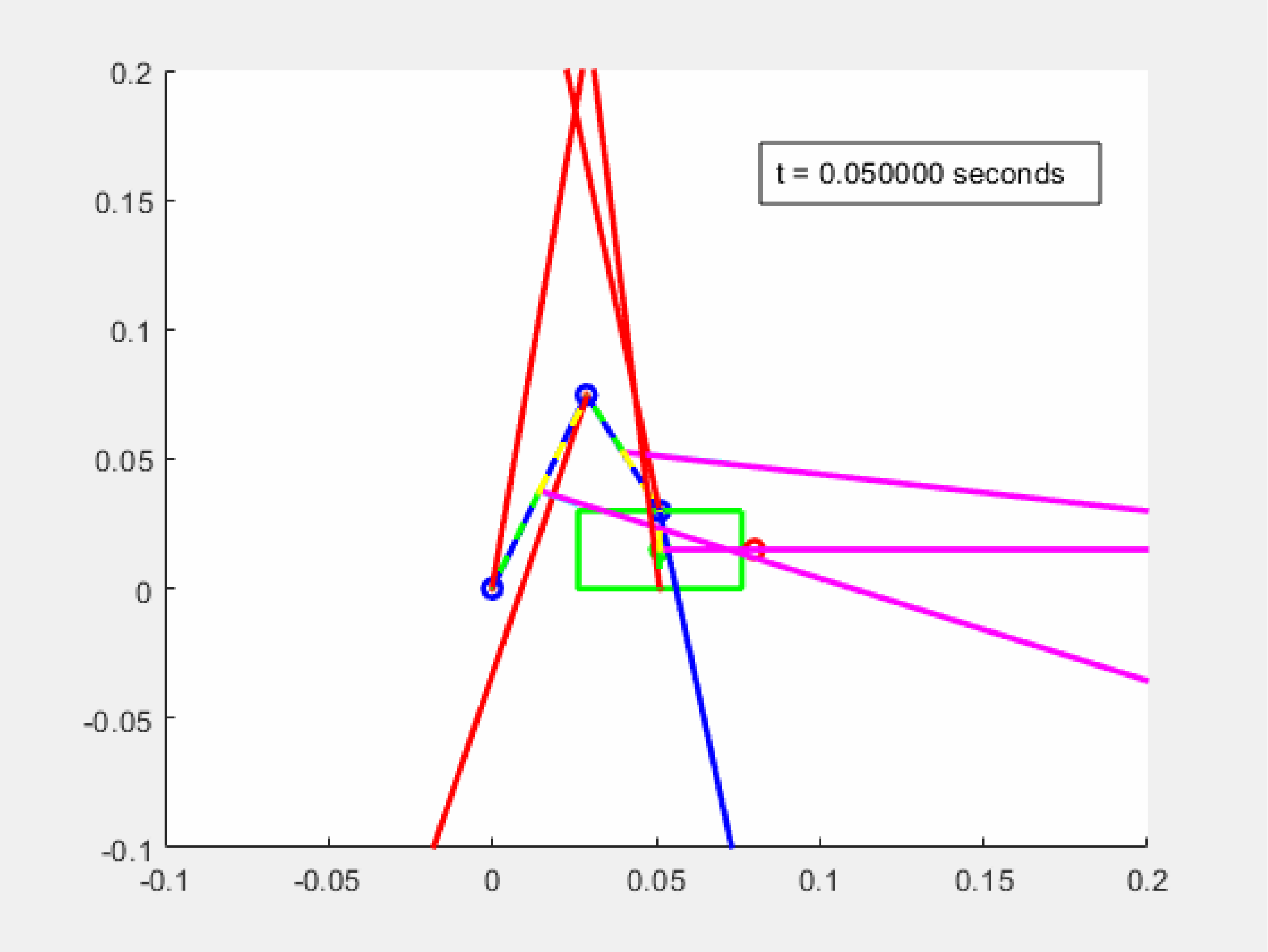


Figure 3.2: Sliding Motion Primitive Simulation

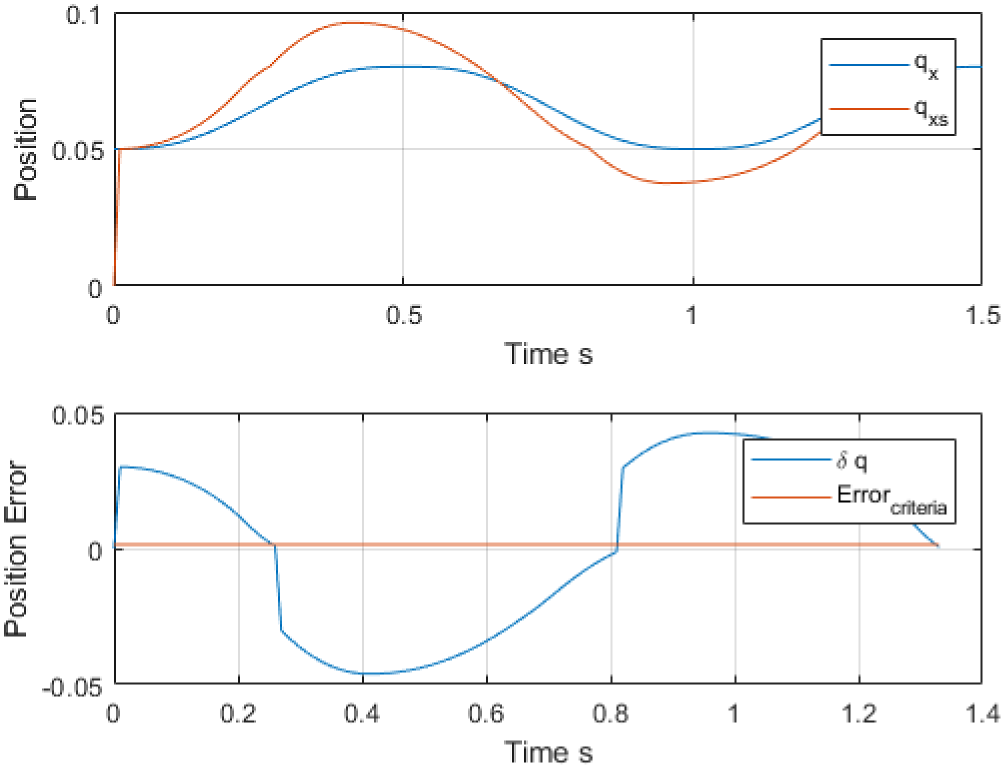


Figure 3.3: Sliding Motion Primitive Simulation Position and Positional Error

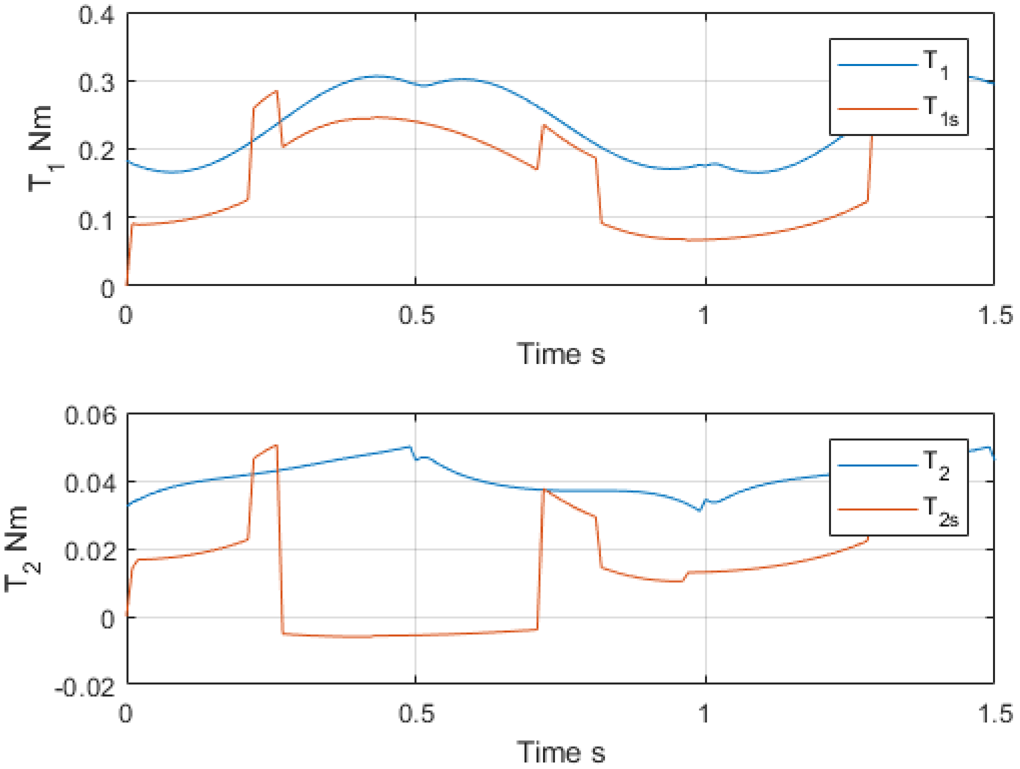


Figure 3.4: Sliding Motion Primitive Simulation Torque

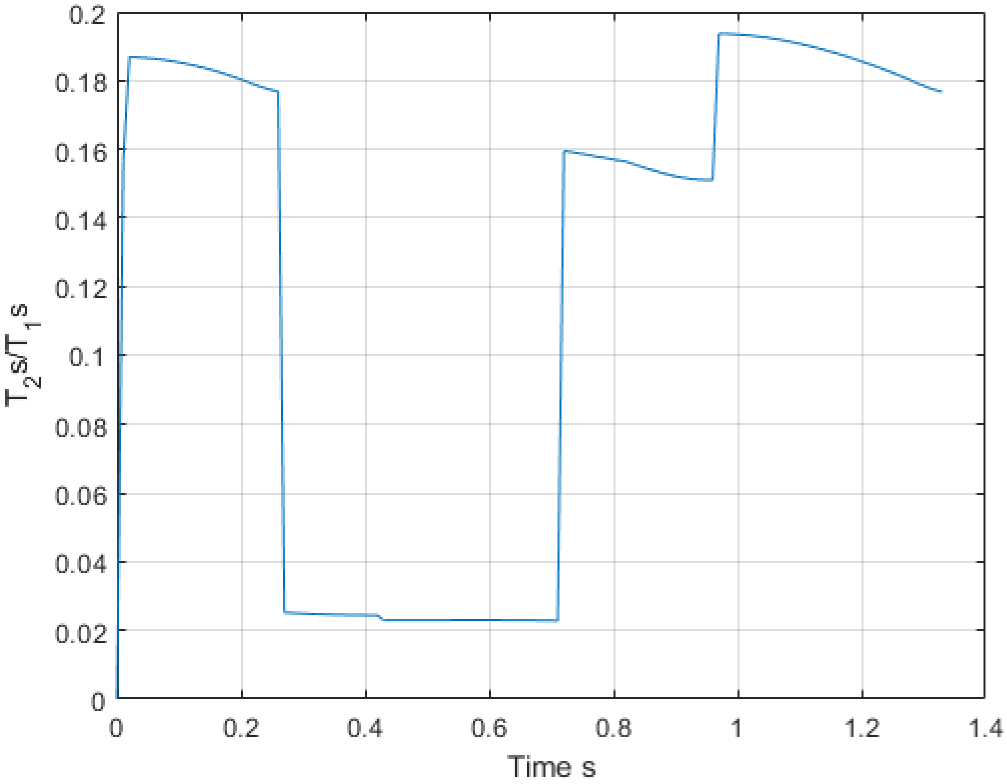


Figure 3.5: Sliding Motion Primitive Simulation Transmission Coefficient

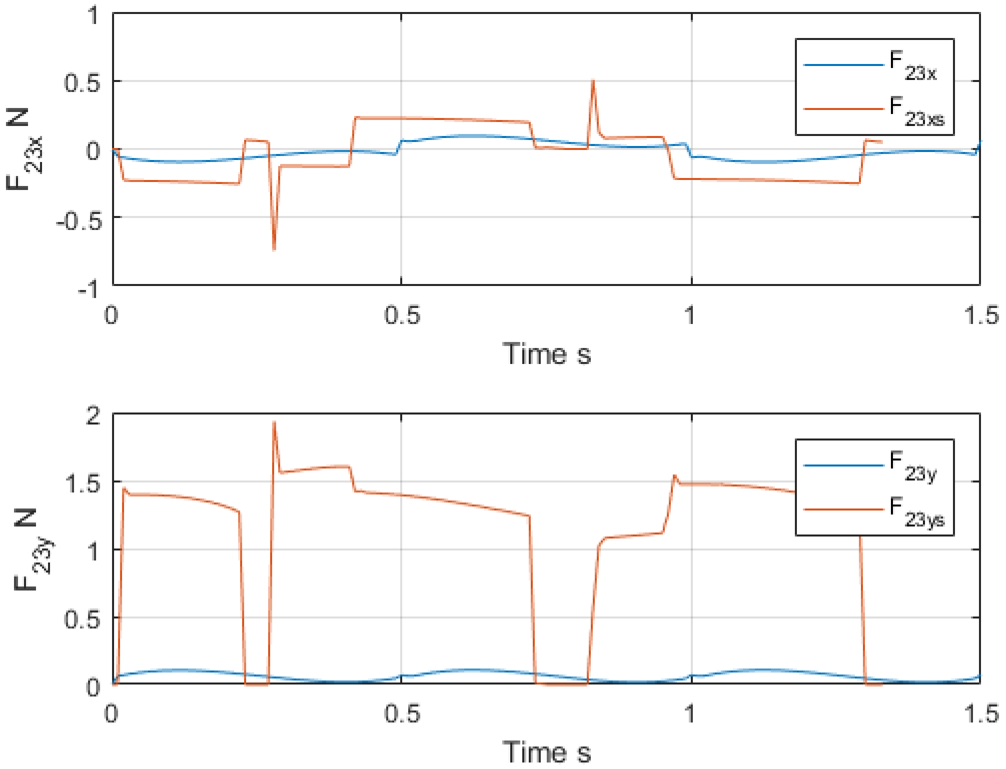


Figure 3.6: Sliding Motion Primitive Simulation Finger - Object Contact Forces

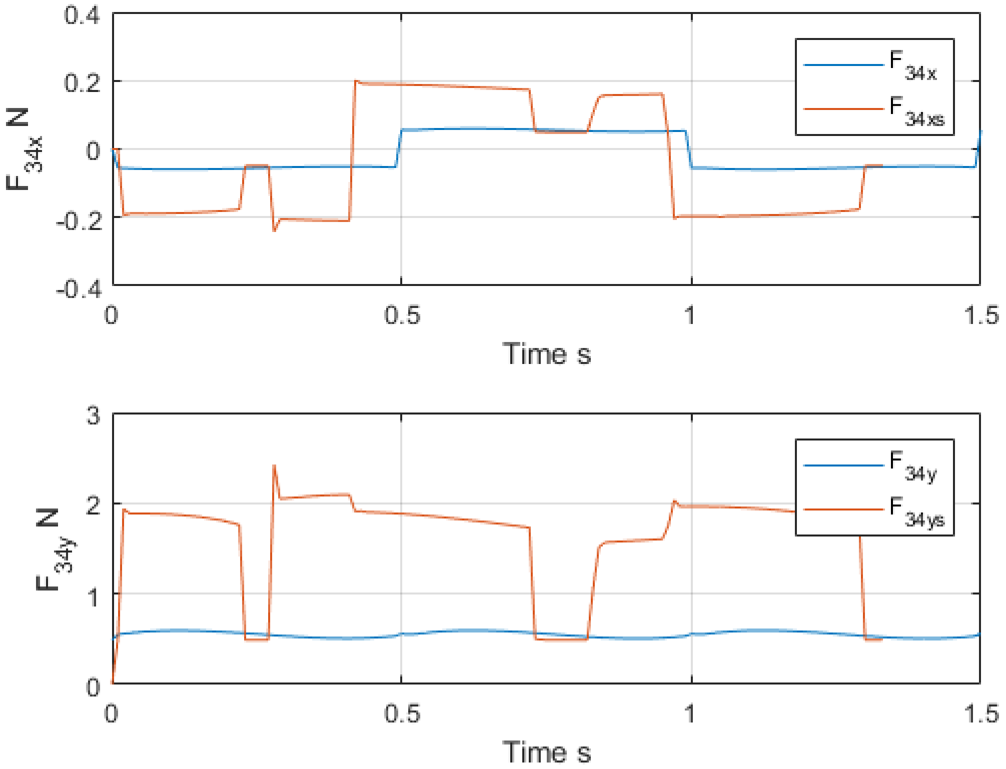


Figure 3.7: Sliding Motion Primitive Simulation Object - Ground Contact Forces

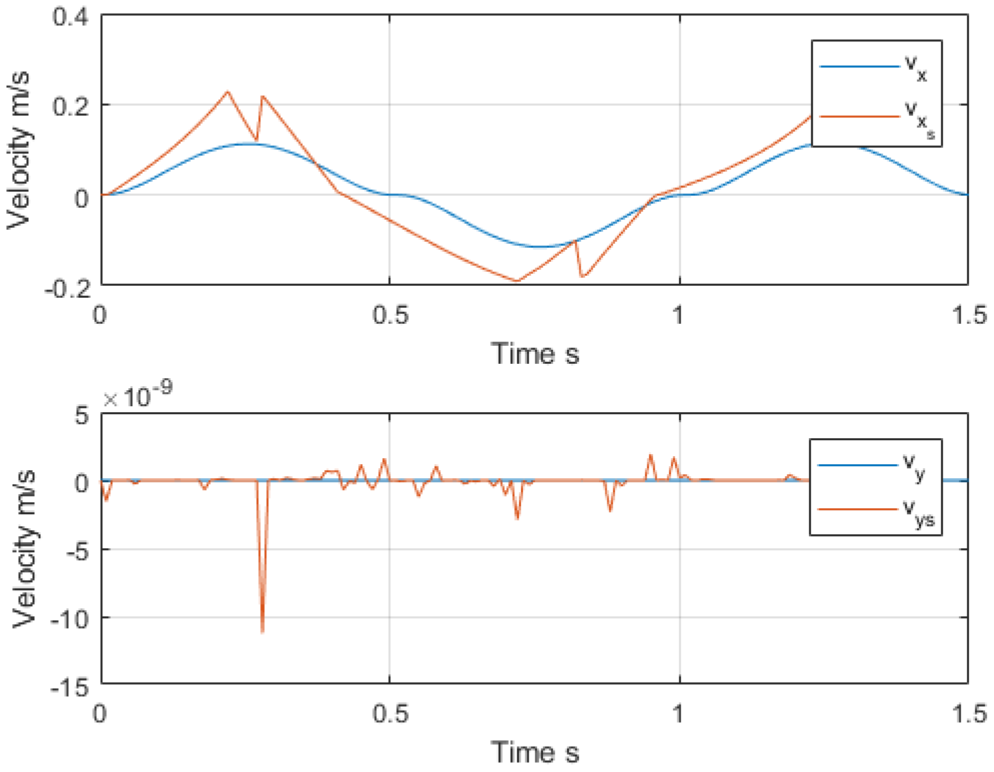


Figure 3.8: Sliding Motion Primitive Simulation Object Linear Velocity

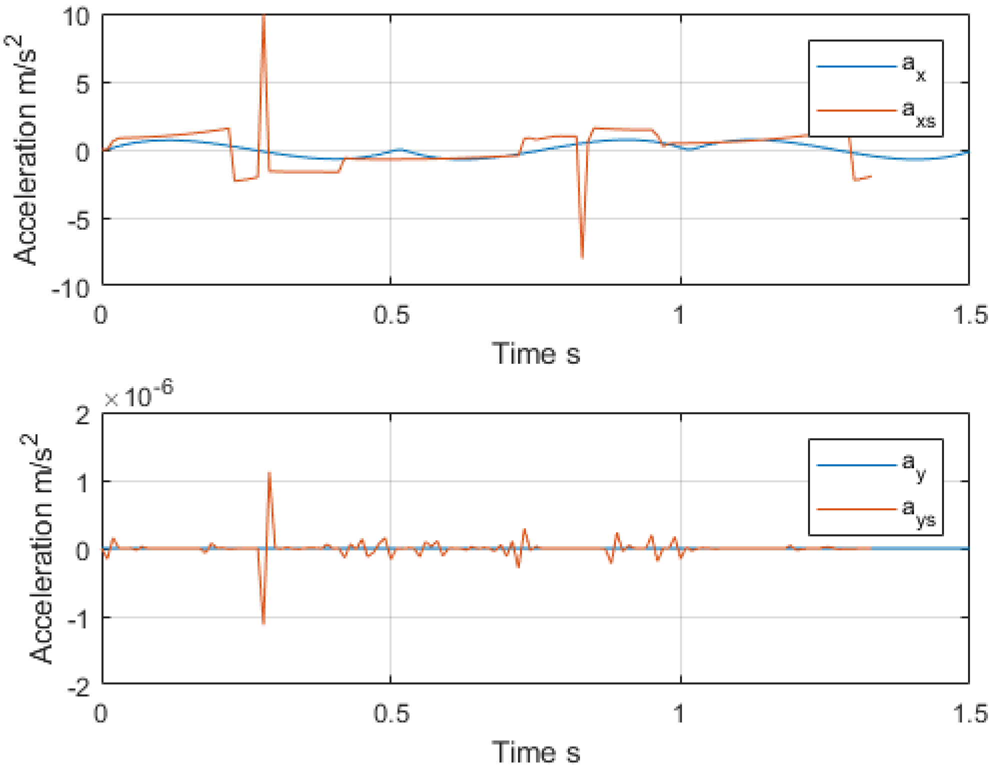


Figure 3.9: Sliding Motion Primitive Simulation Object Linear Acceleration

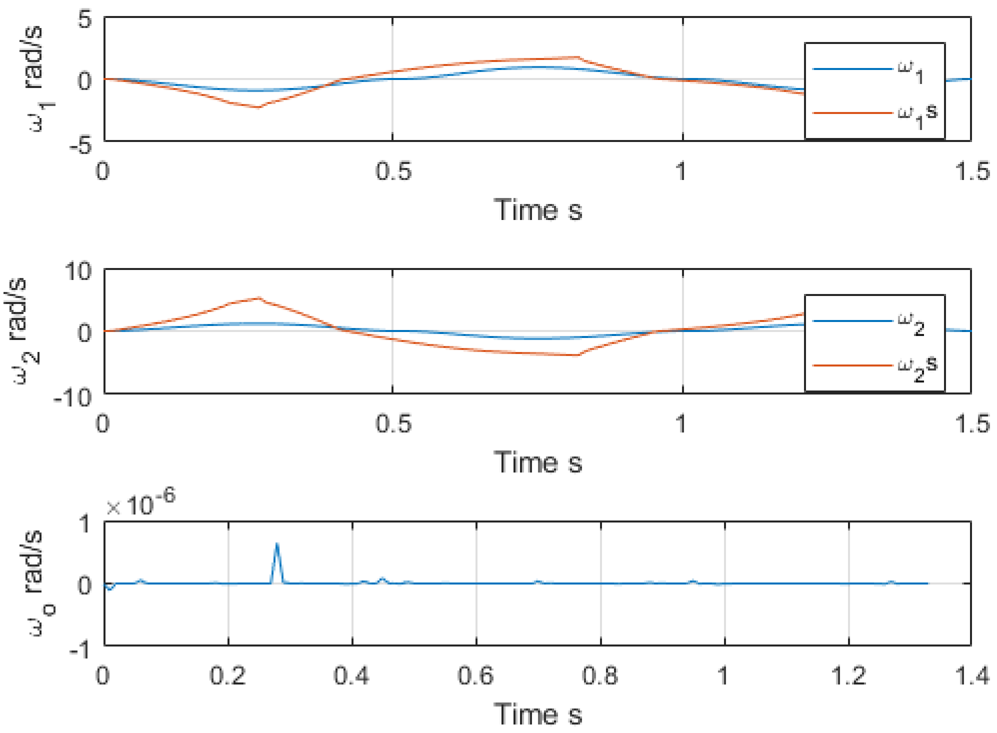


Figure 3.10: Sliding Motion Primitive Simulation Link 1, Link 2 and Object Angular Velocities

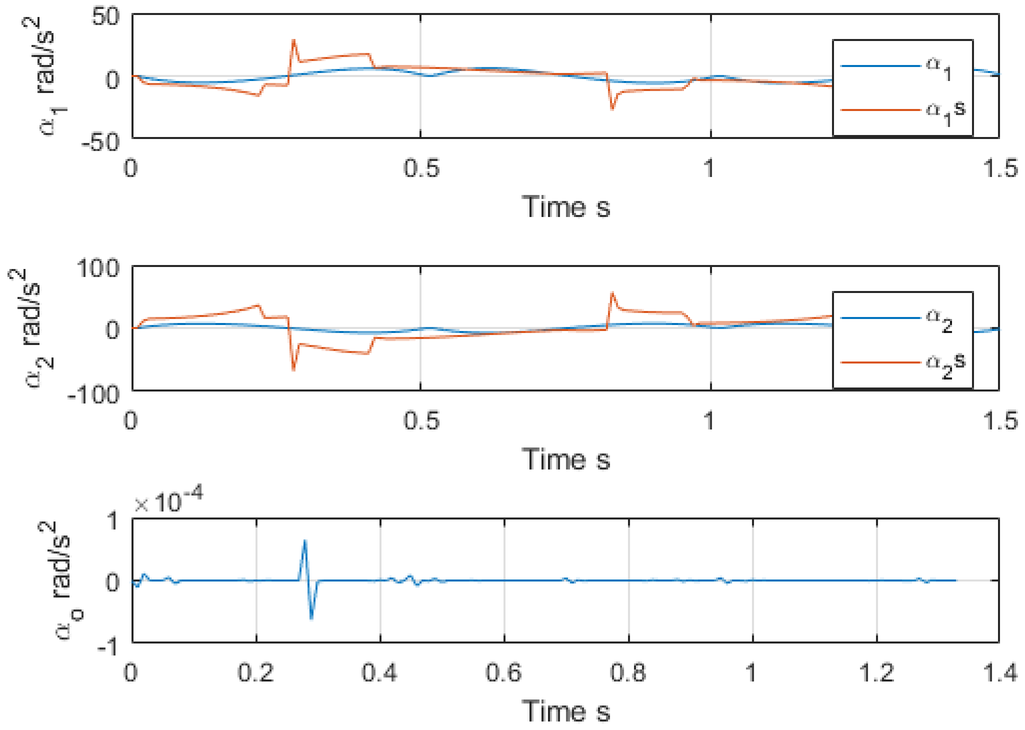


Figure 3.11: Sliding Motion Primitive Simulation Link 1, Link 2 and Object Angular Accelerations

Chapter 4

Multi-Fingered In-Hand Manipulation Motion Primitives

In this chapter, the motion primitive formulation shown previously in 2D is expanded to 3D. Multi-fingered IHM can be composed of a 2D motion primitive for a desired 3D IHM task. One finger can be used as the manipulator while the other fingers provide the necessary object wrenches to ensure the manipulator finger can perform the motion as expected. The finger performing the manipulation is using the desired 2D motion primitive transformed into the desired 3D configuration. The reaction forces are then solved for and used in a second-order cone program (SOCP) in order to calculate the minimal forces that resist an external wrench. By providing the reaction forces from the motion primitive as the external wrench in the SOCP the support finger contact forces can be calculated. This can then be used in a controller to develop a control scheme to perform IHM with underactuated fingers.

4.1 Motion Primitive 3D Representation

First a 2D motion primitive is converted to its 3D frame. This ensures all of the reference frames are handled accordingly, especially for the contact forces. The contact forces solved for in the motion primitive are with respect to the base frame of the finger in the motion primitive. Those forces may then be transformed to the world frame since the transformation matrix between the two frames is known.

The motion of fingers are assumed to be restricted to a plane since each of the fingers only contain 2 DoF. Each of the fingers act in their respective yz-planes. Forces from the local 2D motion primitive can be rewritten as follows.

$$F^l_{MP} = \begin{bmatrix} F^x_{MP} \\ F^y_{MP} \\ F^z_{MP} \end{bmatrix} = \begin{bmatrix} F^x_{MP} \\ F^y_{MP} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ F^x_w \\ F^y_w \end{bmatrix} \tag{4.1}$$

In the local frame the z component is the orthogonal component to the plane and is thus 0 throughout the motion primitive. Due to the fingers acting in the yz-plane, the local x-components become global y-components and the local y-components become global z-components. The forces then need to be properly rotated as well in order to complete the transformation from the local to the global frame. The rotation matrix is given from the known transformation matrix between the local finger frame and the global frame.

4.2 Support Finger Contact Forces

In this section the support finger contact forces for the 3D IHM motion primitive is calculated. A SOCP named CVX developed by Professor Boyd in [12] where he presented a method to solve for the contact forces of a hand for given grasp locations can be used to calculate contact forces on an object to resist external wrenches. These contact forces are subjected to the friction cone constraints which ensure that tangential contact forces that were greater than the normal force would not be considered valid.

We formulate the problem of solving for the contact forces as a force optimization problem (FOP). We define M as the number of contacts with positions $p^{(i)} \in \mathrm{R}^3$ written in the global coordinate system. The force applied at contact point $p^{(i)}$ is $f^{(i)} \in \mathrm{R}^3$ written in the local coordinate system. Each contact frame is defined with the z-axis pointed inwards and orthogonal to the object surface at the point of contact. Thus the x and y components of the forces are exactly tangential at the point of contact to the surface. The rotation matrix that transforms the force vectors from the local at $p^{(i)}$ to the global frame is $Q^{(i)} \in SO(3)$. Writing the force balance equation with respect to the global frame

$$Q^{(1)}f^{(1)} + \ldots + Q^{(i)}f^{(i)} + f^{ext} = 0 \tag{4.2}$$

Similarly the torque balance equation with respect to the global frame may be written

$$p^{(1)} \times Q^{(1)}f^{(1)} + \ldots + p^{(i)} \times Q^{(i)}f^{(i)} + \tau^{ext} = 0 \tag{4.3}$$

The cross product may also be written using matrix multiplication by using the skew symmetric matrix. Let $S^{(i)}$ be defined as

$$S^{(i)} = skew(p^{(i)}) = \begin{bmatrix} 0 & -p_z^{(i)} & p_y^{(i)} \\ p_z^{(i)} & 0 & -p_x^{(i)} \\ -p_y^{(i)} & p_x^{(i)} & 0 \end{bmatrix} \tag{4.4}$$

The contact matrix $A_i \in R^{6x3}$ is defined as

$$A_i = \begin{bmatrix} Q^{(i)} \\ S^{(i)}Q^{(i)} \end{bmatrix} \tag{4.5}$$

which determines the resulting object wrench given a contact force at contact point $p^{(i)}$. The total contact matrix $A \in \mathrm{R}^{6x3M}$ can be written as

$$A = \begin{bmatrix} A_1 & \cdots & A_i \end{bmatrix} \tag{4.6}$$

The total force vector ,$f \in \mathrm{R}^{3M}$, is a column vector which is concatenated of the individual contact force column vector.

$$f = \begin{bmatrix} f_x^{(1)} \\ f_y^{(1)} \\ f_z^{(1)} \\ \vdots \\ f_x^{(M)} \\ f_y^{(M)} \\ f_z^{(M)} \end{bmatrix} \tag{4.7}$$

The total external wrench is $\omega^{ext} = (f^{ext}, \tau^{ext}) \in R^6$ which acts at the center of gravity of the object. This is the wrench that is to be resisted by the fingers. The problem may be written as a set of 6 linear equations in the following form.

$$Af + \omega^{ext} = 0 \tag{4.8}$$

The product of Af gives the total wrench on object due to contact forces at postions $p^{(i)}$. In this problem, one of the finger contact forces is known from the desired motion primitive. When calculating the contact forces for the other two supporting fingers, the known finger contact forces are written into the total contact matrix A and force vector f as follows

$$A = \begin{bmatrix} A_{(1)} & \dots & A_{(M)} & A_{(mp)} \end{bmatrix} \tag{4.9}$$

$$f = \begin{bmatrix} f^{(1)} \\ \vdots \\ f^{(M)} \\ f^{(mp)} \end{bmatrix} \tag{4.10}$$

where $A_{(mp)}$ and $f^{(mp)}$ are the contact matrix and the contact forces of the finger performing the motion primitive. This simplifies the choice of external wrench ω^{ext} to just the force of gravity that acts at the object's center of gravity.

$$\omega^{ext} = \begin{bmatrix} 0 \\ 0 \\ F_z^g \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.11}$$

where F_z^g is the force of gravity in the -z direction. By adding the third finger as a known constant, thus applying an object wrench to perform the desired motion, the supporting fingers will handle providing the reaction forces. Another method includes adding the object wrench caused by the finger performing the motion primitive to the external wrenches vector. An object wrench can be determined from a contact force through the adjoint matrix.

$$F_c = Ad^T_{g_{bc}} F_b \tag{4.12}$$

$$Ad^T_{g_{bc}} = \begin{bmatrix} R_{bc} & 0 \\ \hat{p_{bc}} R_{bc} & R_{bc} \end{bmatrix} \in \mathrm{R}^{6x6} \tag{4.13}$$

$$F = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \in \mathrm{R}^6 \tag{4.14}$$

The contact wrench at the local contact frame at p_{bc} is transformed to an equivalent object wrench. With this formulation the total contact matrix A and force vector f are 6 x 6 and 6 x 1 respectively. The equivalent object wrench is added to the external wrench ω^{ext}.

The friction cone constraints may be defined as

$$\begin{aligned} ||(f_x^{(i)}, f_y^{(i)})|| = & \sqrt{(f_x^{(i)^2} + f_y^{(i)^2})} \leq \mu_i f_z^{(i)} \\ & f^{(i)} \in K_i \end{aligned} \tag{4.15}$$

where K_i is the 3D friction cone and f_x, f_y, f_z are the x, y and z components of the contact force in the contact frame. The contact assumption is currently a point contact model which means the fingers can only exert normal and tangential forces but not torques at contact points.

The force optimization problem aims to find a set of contact force vectors $f^{(i)}$ that are as small as possible while satisfying the friction cone constraints and the force and torque equilibrium constraints. Let F^{max} be the maximum contact force magnitude of the grasp.

$$\begin{aligned} F^{max} &= max(||f^{(i)}|| \cdots ||f^{(M)}) \\ &= max_{i=1 \ldots M} \sqrt{f_x^{(i)^2} + f_y^{(i)^2} + f_z^{(i)^2}} \end{aligned} \tag{4.16}$$

The problem is formulated to minimize the maximum contact force of the grasp

$$\begin{aligned} \text{minimize} \quad & F^{max} \\ \text{subject to} \quad & Af + w^{ext} = 0 \\ & f^{(i)} \in K_i \end{aligned} \tag{4.17}$$

The SOCP is solved for each step of the motion primitive and provides the necessary contact forces of the supporting finger to perform IHM in 3D. These contact forces can then be converted into respective joint torques using the following

$$\tau = J^T f \tag{4.18}$$

where f is the wrench applied to the tip of the finger, J is the Jacobian matrix of the finger and τ is the vector of torques at the joints.

4.2.1 Sliding Support Contact Forces

This section provides an example of calculating the sliding motion primitive support finger contact forces. The 2D sliding motion primitive from Chapter 2 was used in the calculation for the forces of the supporting fingers. At each time step, the supporting contact forces are calculated given the necessary object-ground forces from Chapter 2. This can be repeated for the other two motion primitives in order to calculate their respective support finger contact forces.

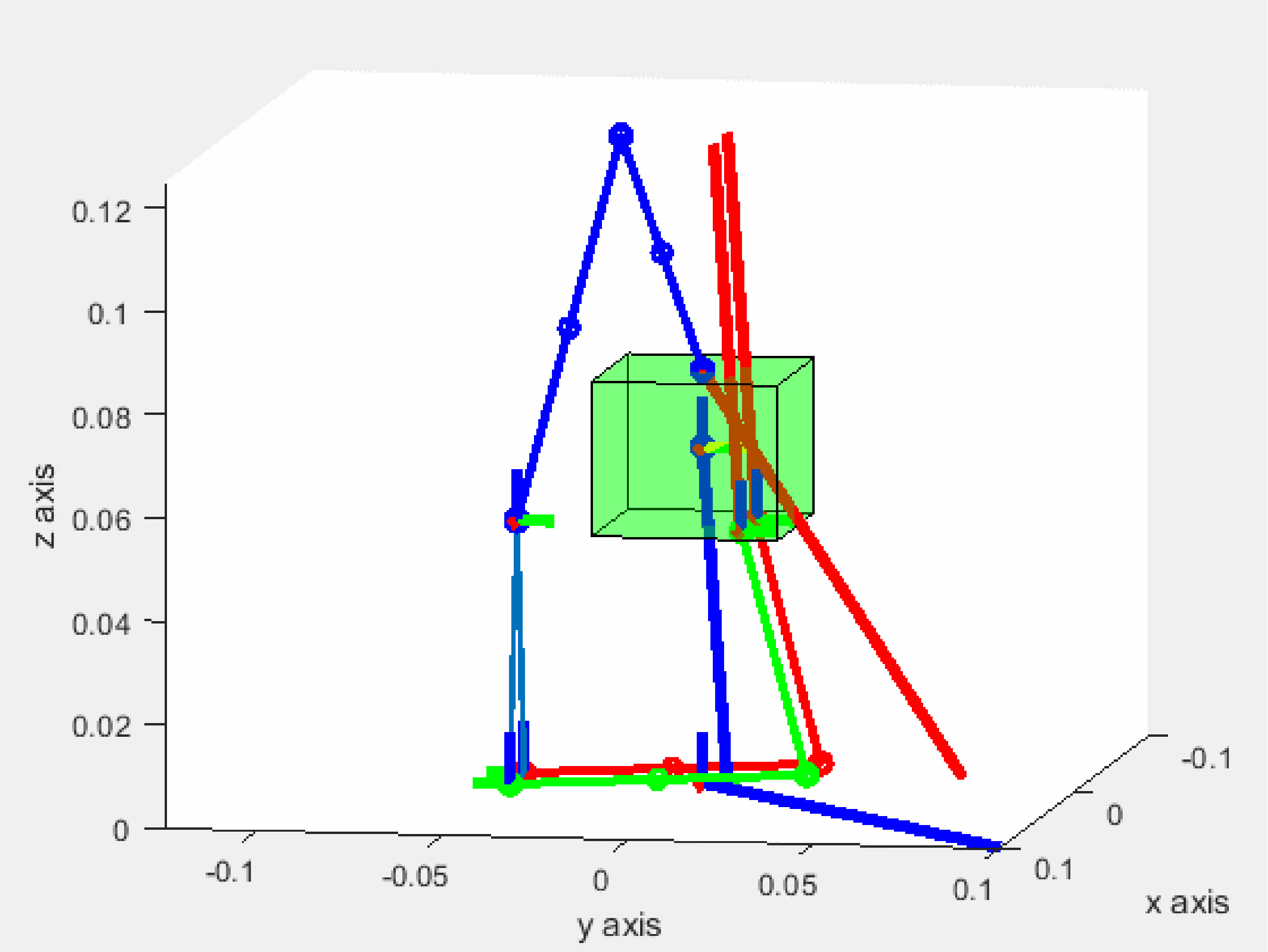


Figure 4.1: Sliding Motion Primitive Contact Forces

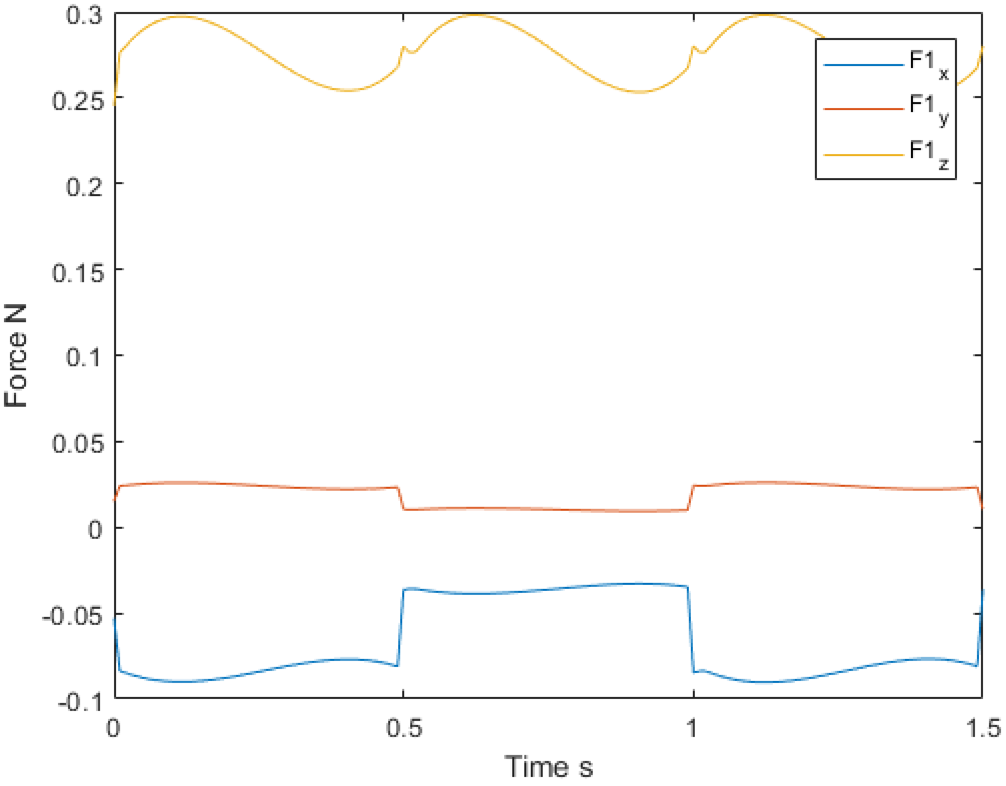


Figure 4.2: Sliding Motion Primitive Contact Forces Finger 1

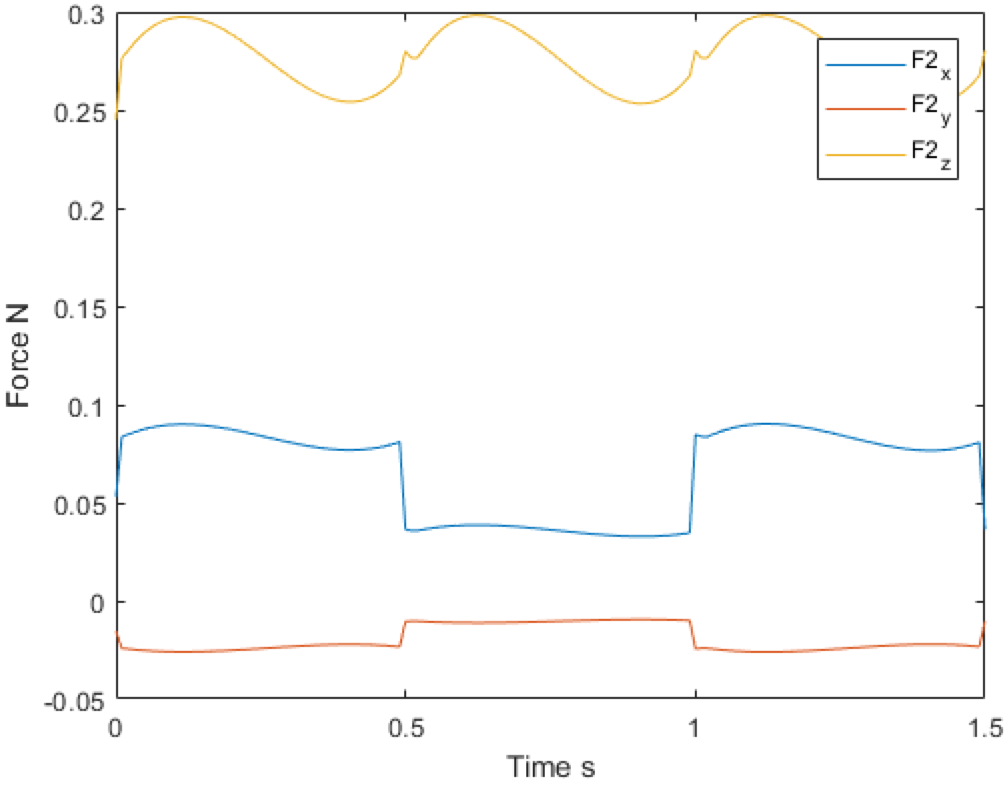


Figure 4.3: Sliding Motion Primitive Contact Forces Finger 2

Chapter 5

Conclusion

This chapter will provide a section on a discussion of the work and results presented in this thesis and a section outlining future work for the research. Care was taken to separate my interpretation of the results until this chapter as to not provide any pre-existing bias to the readers. The reader will be able to make their own interpretations of the results and then compare their interpretations with mine. Hopefully this process provides effective academic discussions in the future. The future work section will describe what I have determined to be the best next steps in continuing with the research presented in this thesis. These are only meant to be my interpretations as the reader is free and supposed to have their own interpretation of how to continue the work. All comments, corrections and suggestions are recommended.

5.1 Discussion

In this section, the results and my interpretations of the thesis will be discussed in detail. First, I will discuss the results in Chapter 2 regarding each of the motion primitive examples. Then the validation of the method and results presented in Chapter 3 will be discussed. All recommendations for improvements will be described in the future work section.

The results from each of the examples, I believe, are consistent to what a real-life system would demonstrate although this should be verified carefully. The design of the desired trajectories are mostly straight-forward and intuitive however, I find the large jerks possibly troublesome in real-life systems. While there are no aggressive spikes in velocities or accelerations in these given examples, it is shown in Chapter 3 when testing on a dynamic simulation some of these can occur. Limitations on hardware, especially motor and motor controller response rates, can incur unforeseen errors. These may lead to undesirable finger behavior causing the object to move in unpredictable ways. However, this does not seem to be an issue with the planning, as shown in Chapter 2, but with how the feedback control method is currently being utilized. This will be discussed further in the future work section below. Each of the examples satisfies their respective constraints in consistent manners. For the sliding motion primitive, object y-component velocities and object angular velocities and accelerations are zero. This is consistent with the desired motion of the object. Likewise, y-component of velocity for the rolling motion primitive is zero while there are angular velocities and accelerations present. For tipping all components of the motion are present in the motion primitive. For the forces and torques, physical intuitions for the system are difficult to analyze without first testing on hardware with different sensors to verify the accuracy of the solved forces and torques. I expect errors due to attempts to model the physical system however, when considering the magnitudes of the forces and torques provided from the motion primitives, it seems that they are consistent with the input parameters to each of the motion primitive examples. During each of the motion primitives, the finger-object contact forces seem to make sense physically. The x-component of each of the contact forces satisfy their respective friction cone constraints. One potential issue that might arise is that the linear optimization program solves for x-component contact forces which lie on the boundary of the friction cone constraint. This compatible with physical intuition as the x-component of the forces are required in order to move the object for each of the motions thus they must be as great as possible. However, my instinct tells me that in real-life systems a safety factor of some sort might be desired or some way of handling potential losses of contact should be explored. This is because any errors in modeling the physics of the system can cause the finger-object contact forces, which already lie on the edges of their respective friction cone constraints, to leave the friction cone thus causing slip at the contact point. While preventing slip may be the desired course of action to remediate this problem in real-life scenarios, there are certainly situations where slipping may be inevitable and different finger manipulation techniques might be used. For example, the contact between the finger and the object can be reestablished and the motion primitive plan recalculated and executed. Another option might be to use different finger gaiting techniques which is something humans do when manipulation objects quite frequently when performing IHM.

The results presented in Chapter 3, I believe, are promising because they show that the motion primitives developed in Chapter 2 are consistent with other physical models and can be utilized in a control scheme with the dynamic simulation acting as the system and its state feedback. The planner provides torque inputs for the dynamic simulation and the dynamic simulation presents state feedback at the end of the time step. The ability to move the object to the desired location while roughly maintaining similar position, velocity and acceleration profiles for the object demonstrates this methods capabilities. All of the calculated parameters are physically intuitive but spikes in the accelerations should be reduced. The simulation validation shows great room for improvements in many areas regarding this research. When initially developing the feedback control loop presented in Chapter 3, utilizing the planner without any external controllers provided errors mainly arising when the object was near the desired goal locations. This was due to overfitting of the SVAJ curves and issues with handling the desired time scale constant for the designed motion. As the object got closer to the goal position the SVAJ curves would provide curves and torques which lead the object to overshoot the goal position. This was handled by the solution described in Chapter 3 which involved the design of a few external controllers to manage these overshoots. While the system results are fairly close to the desired system results, more work can be done to further reduce the overshoot and improve controller design. The method is shown to work with dynamic simulations consistent with real-life systems and thus should be implemented on real hardware in order to continue improving the research presented in the thesis.

5.2 Future Work

In this section I will outline the next steps in further validation of the method and how to proceed in expanding this work. First, I believe that it is important to validate this method and the control scheme on real-life hardware. Dynamic simulations, like all mathematical models of physical phenomena, are subject to errors that might go undetected until they have been tested in the real-life situations. While testing on real-life hardware does involve more difficulty and adds a lot of areas where inconsistencies might arise, the benefits learning the true capabilities of the method is invaluable. This is how I would structure the next steps in the hardware validation stage. A small test fixture with one figure should be constructed. The finger tip, the object and the platform which will act as the ground for the object should be constructed of materials with well known friction coefficients. Many different sensors should be available as well. Encoders for joint positions, force sensors both on the finger-tips and the object platform, torque sensors for the motors, and a sensor to provide object position and orientation feedback. If handled correctly, this test fixture could provide valuable validation data as well as insight into improvements of the method. One important question to handle is the issue of how to design the transmission mechanism for this test fixture. I would suggest bypassing this entirely at this stage. My reasoning is this, first and foremost validation of the method on real hardware is needed before exerting effort on designing experimental torque distribution systems. This is because relevant data can be collected without it by supplying torques to each of the joints individually. The controller can command torques provided by the planner to each of the joints and verify that the torques are valid. As long as the system accurately models the finger in the planner's motion primitive, it should work as expected. Another reason is that development of this transmission mechanism will take time and come with physical limitations. Motion primitives can be designed with transmission coefficients that are not physically realizable. However, by separating the testing and collection of data from this constraint, research can continue to improve the method as research to improve the transmission mechanism may improve in time as well.

Once the method is validated with one finger in the test fixture work can also be done to explore on how to select, design and perform multiple motion primitives to perform IHM tasks. IHM tasks will probably need to be composed of multiple motion primitives. There is no immediately obvious way, at least to me, to perform this planning in a principled manner. This should be explored carefully. Work in this area is imperative however as the capability to approximate IHM skills, which make up the IHM tasks, could improve extremely useful in future iterations of this research. Furthermore, the previous development of the single-fingered test fixture would greatly aide in data collection for this step. Once the work is done to approximate IHM skills in single-fingered scenarios, the motion primitives can then be used to approximate IHM skills in multi-fingered scenarios. As described in Chapter 4, a motion primitive can be designed in 3D by taking a 2D plane and designing the motion primitive given the desired object motions. Reaction forces, instead of being provided by the ground or an external support, must be provided by the supporting fingers. The initial steps for this method have been shown in Chapter 4 but more work must be done to ensure the validity of the method. I believe a similar approach as to the one done for the single-fingered method should work. First, ensure that the theory and model of the motion primitives and its planning for IHM with multi-fingered hands is validated at some level in simulation. Then it is necessary to develop a test fixture. This step might be more complicated as it involves a large amount of work to design the hand, implement the sensors and then ensure the fixture is well controlled for data collection. The next step would then be to compose IHM skills in 3D as previously done in 2D. This step will also be difficult as now the planning space is a bit large. Multiple fingers provide many different combinations of active and supporting fingers which needs to be explored and then selected given the desired IHM task. However, I believe by proceeding in this principled manner satisfactory results will be obtained.

In conclusion, I believe the results provided by the dynamic simulation validate that the method presented in this thesis has the potential to perform IHM tasks with underactuated robotic hands. My interpretations of the results are merely guidelines for future researchers to use when continuing this work. Similarly, the procedure for continuing future work is also just my suggestions on how to proceed with future research. If successful, the future applications and significance of the work will be evident. Underactuated robotic hands, which are currently only being used for power grasps and other small tasks, will have the capabilities to perform a wide array of dexterous functions. This has implications to the improvements of robotic hands in advanced assembly tasks in the manufacturing environment, improvements in robotic hand usage of human tools and the progress of mobile robotics autonomy when performing different functions.

Bibliography

[1] Diogo Almeida and Yiannis Karayiannidis. Dexterous manipulation by means of compliant grasps and external contacts. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1913–1920. IEEE, 2017.

[2] Rika Antonova, Silvia Cruciani, Christian Smith, and Danica Kragic. Rienforcement learning for pivoting tasks. *arXiv preprint arXiv:1703.00472*, 2017.

[3] Hirohiko Arai, Kazuo Tanie, and Susumu Tachi. Dynamic control of a manipulator with passive joints in operational space. *IEEE International Conference on Robotics and Automation*, 1993.

[4] Daniel Aukes, Barrett Heyneman, Vincent Duchaine, and Mark R. Cutkosky. Varying spring preloads to select grasp strategies in an adaptive hand. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[5] Daniel Aukes, Susan Kim, Pablo Garcia, Aaron Edsinger, and Mark R. Cutkosky. Selectively compliant underactuated hand for mobile manipulation. *IEEE International Conference on Robotics and Automation*, 2012.

[6] Francisco Eli Vina Barrientos, Yiannis Karayiannidis, Karl Pauwels, Christian Smith, and Danica Kragic. In-hand manipulation using gravity and controlled slip. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[7] Francisco Eli Vina Barrientos, Yiannis Karayiannidis, Christian Smith, and Danica Kragic. Adaptive control for pivoting with visual and tactile feedback. *IEEE International Conference on Robotics and Automation*, 2016.

[8] Fransisco Eli Vina Barrientos. *Robotic manipulation under uncertainty and limited dexterity*. PhD thesis, KTH Royal Institue of Technology in Stockholm, 2016.

[9] Maria Bauza and Alberto Rodriguez. A probabilistic data-driven model for planar pushing. *IEEE International Conference on Robotics and Automation*, 2017.

[10] Antonio Bicchi. On the closure properties of robotic grasping. *The International Journal of Robotics Research*, 14(4):319–334, 1995.

[11] Lionel Birglen and Clement M. Gosselin. Kinetostatic analysis of underactuated fingers. *IEEE Transactions on Robotics and Automation*, 2004.

[12] Stephen P. Boyd and Ben Wegbreit. Fast computation of optimal contact forces. *IEEE Transactions on Robotics*, 23(6):1117–1132, 2007.

[13] Ian M. Bullock and Aaron M. Dollar. Classifying human manipulation behavior. *IEEE International Conference on Rehabilitation Robotics*, 2011.

[14] J. Alfredo Rosas-Flores Jaime Alvarez-Gallegos Rafael Castro-Linares. Control of an underactuated planar 2r manioulator: Experimental resutls. *IFAC Journal of Systems and Control*, 2002.

[15] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi. Adaptive synergies for the design and control of the pisa/iit softhand. *The International Journal of Robotics Research*, 2014.

[16] Sun-Lai Chang, Jyh-Jone Lee, and Home-Che Yen. Kinematic and compliance analysis for tendon-driven robotic mechanisms with flexible tendons. *Mechanism and Machine Theory*, 2005.

[17] Nikhil Chavan-Dafle and Alberto Rodriguez. Stable prehensile pushing: In-hand manipulation with alternating sticking contacts. *IEEE International Conference on Robotics and Automation*, 2018.

[18] Seyoung Cheon, Woohyeok Choi, Sang-Rok Oh, and Yonghwan Oh. Development of an underactuated robotic hand using differential gear mechanism. *Ubiquitous Robots and Ambient Intelligence*, 2014.

[19] Moez Cherif and Kamal K. Gupta. Planning for in-hand dexterous manipulation. *Third Workshop on the Algorithmic Foundations of Robotics*, 1998.

[20] Vincent C. K. Cheung, Andrea d'Avella, Matthew C. Tresch, and Emilio Bizzi. Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviors. *The Journal of Neuroscience*, 2005.

[21] Matei Ciocarlie and Peter Allen. Data-driven optimization for underactuated robotic hands. *IEEE International Conference on Robotics and Automation*, 2010.

[22] Matei Ciocarlie, Corey Goldfeder, and Peter Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. *International Conference on Intelligent Robots*, 2007.

[23] Matei T. Ciocarlie and Peter K. Allen. Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 2009.

[24] Salvador Cobos, Manuel Ferre, M. A. Sanchez Uran, Javier Ortego, and Cesar Pena. Efficient human hand kinematics for manipulation tasks. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[25] Mark R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 1989.

[26] Nikhil Chavan Dafle, Alberto Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S. Srinivasa, Michael Erdmann, Matthew T. Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. *IEEE International Conference on Robotics and Automation*, 2014.

[27] Raphael Deimel and Oliver Brock. A compliant hand based on a novel pneumatic actuator. *IEEE International Conference on Robotics and Automation*, 2013.

[28] Aaron M. Dollar and Robert D. Howe. The highly adaptive sdm hand: Design and performance evaluation. *The International Journal of Robotics Research*, 2010.

[29] Charlotte E. Exner. The zone of proximal develop in in-hand manipulation skills of nondysfunctional 3- and 4-year-old children. *The American Journal of Occupational Therapy*, 1990.

[30] Charlotte E. Exner. Content validity of the in-hand manipulation test. *American Journal of Occupational Therapy*, 1993.

[31] Nima Fazeli, Samuel Zapolsky, Evan Drumwright, and Alberto Rodriguez. Fundamental limitations in performance and interpretability of common planar rigid-body contact models. *International Symposium of Robotic Research*, 2017.

[32] Nima Fazeli, Samuel Zapolsky, Evan Drumwright, and Alberto Rodriguez. Learning data-efficient rigid-body contact models: Case study of planar impact. *Conference on Robotic Learning*, 2017.

[33] J. Randall Flanagan, Miles C. Bowman, and Roland S. Johanss. Control strategies in object manipulation tasks. *Current Opinion in Neruobiology*, 2006.

[34] Marco Gabiccini, Antonio BIcchi, D. Prattchizzo, and M. Malvezzi. On the role of hand synergies in the optimal choice of grasping forces. *Autonomous Robots Springer*, 2011.

[35] G. Gioioso, G. Salvietti, M. Malvezzi, and D. Prattichizzo. Mapping synergies from human to robotic hands with dissimilar kinematics: an approach in the object domain. *IEEE Transactions on Robotics*, 2013.

[36] Clement Gosselin, Frederic Pellentier, and Thierry Lalierte. An anthropomorphic underactuated robotic hand with 15 dofs and a single actuator. *IEEE International Conference on Robotics and Automation*, 2008.

[37] Abhishek Gupta, Clemens Eppner, Sergey Levine, and Pieter Abbeel. Learning dexterous manipulation for a soft robotic hand from human demonstrations. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[38] Junhu He. *Robotic In-hand Manipulation with Push and Support Method*. PhD thesis, University of Hamburg, 2017.

[39] Francois Robert Hogan and Alberto Rodriguez. Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics. *Workshop on the Algorithmic Foundations of Robotics*, 2016.

[40] Anne Holladay, Robert Paolini, and Matthew T. Mason. A general framework for open-loop pivoting. *IEEE Internatioal Conference on Robotics and Automation*, 2015.

[41] Keum-Shik Hong. An open-loop control for underactuated manipulators using oscillatory inputs: Steering capability of an unactuated joint. *IEEE Transactions on Control Systems Technology*, 2002.

[42] Yifan Hou, Zhenzhong Jia, Aaron M. Johnson, and Matthew T. Mason. Robust planar dynamic pivoting by regulating inertial and grip forces. *The International Workshop on the Algorithmic Foundations of Robotics*, 2016.

[43] Joshua M. Inouye. *Bio-Inspired Tendon-Driven Systems: Computational Analysis, Optimization, and Hardware Implementation*. PhD thesis, University of Southern California, 2012.

[44] Joshua M. Inouye, Jason J. Kutch, and Francisco J. Valero-Cuevas. A novel synthesis of computational approaches enables optimization of grasp quality of tendon-driven hands. *IEEE Transactions on Robotics*, 2012.

[45] Joshua M. Inouye and Francisco J. Valero-Cuevas. Anthropomorphic tendon-driven robotic hands can exceed human grasping capabilities following optimization. *The International Journal of Robotics Research*, 2013.

[46] Jason J. Kutch Joshua M Inouye and Francisco J. Valero-Cuevas. Optimizing the topology of tendon-driven fingers: Rationale, predictions and implementation. *Springer Tracts in Advanced Robotics*, 2014.

[47] James J. Kuffner Katsu Yamane and Jessica K. Hodgins. Synthesizing animations of human manipulaion tasks. *ACM Transactions on Graphics*, 2004.

[48] Hiroaki Kobayashi, Kazuhito Hyodo, and Daisuke Ogane. On tendon-driven robotic mechanisms with redundant tendons. *The International Journal of Robotics Research*, 1998.

[49] Gert A. Kragten and Just L. Herder. The ability of underactued hands to grasp and hold objects. *Mechanism and Machine Theory*, 2010.

[50] Vikash Kumar. *Manipulators and Manipulation in high dimensional spaces*. PhD thesis, 2016.

[51] Vikash Kumar, Abhishek Gupta, Emanuel Todorov, and Sergey Levine. Learning dexterous manipulation policies from experience and imitatation. *Computing Research Repository*, 2016.

[52] Thierry Laliberte, Lionel Birglen, and Clement M. Gosselin. Underactuation in robotic grasping hands. *Journal of Machine Intelligence and Robotic Control*, 2002.

[53] Thierry Laliberte and Clement M. Gosselin. Simulation and design of underactuation mechanical hands. *Mechanism and Machine Theory*, 1998.

[54] Yueh-Hua Lee and Jyh-Jone Lee. Modeling of the dynamics of tendon-driven robotic mechanisms with flexible tendons. *Mechanism and Machine Theory*, 2003.

[55] Zexiang Li and S. Shankar Sastry. Task-Oriented Optimal Grasping by Multifingered Robot Hands. *IEEE Journal of Robotics and Automation*, 1988.

[56] Minas V. Liarokapis and Aaron M. Dollar. Learning task-specific models for dexterous, in-hand manipulation with simple, adaptive robot hands. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[57] Fabio Londi, Ettore Pennestri, Pier Paolo Valentini, and Leonardo Vita. Control and virtual reality simulation of tendon driven mechanisms. *Multibody System Dynamics*, 2004.

[58] Alessandro De Luca, Stfano Iannitti, and Raffaella Mattone andGiuseppe Oriolo. Underactuated manipulators: Control properties and techniques. *Machine Intelligence and Robotic Control*, 2003.

[59] Raymond Ma, Adam Spiers, and Aaron M. Dollar. M 2 gripper: Extending the dexterity of a simple, underactuated gripper. *Advances in reconfigurable mechanisims and robots ll*, 2016.

[60] Raymond R. Ma, Lael U. Odhner, and Aaron M. Dollar. A modular, open-source 3d printed underactuated hand. *IEEE International Conference on Robotics Automation*, 2013.

[61] Arun D. Mahindrakar, Shodhan Rao, and R. N. Banavar. Point-to-point control of a 2r planar horizontal underactuated manipulator. *Mechanism and Machine Theory Elsevier*, 2006.

[62] Monica Malvezzi, Guido Gioioso, Gionata Salvietti, Domenico Prattichizzo, and Antonio Bicchi. Syngrasp: a matlab toolbox for grasp analysis of human and robotic hands. *IEEE International Conference on Robotics and Automation*, 2013.

[63] Carolyn R. Mason, Jose E. Gomez, and Timothy J. Ebner. Hand synergies during reach-to-grasp. *Journal of Neurophysiology*, 2001.

[64] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

[65] Ehtesham Nazma and Suhaib Mohd. Tendon driven robotic hands: A review. *International Journal of Mechanical Engineering and Robotics Research*, 2012.

[66] Van-Duc Nguyen. Constructing force-closure grasps. *The International Journal of Robotics Reasearch*, 1988.

[67] Robert L. Norton. *Design of Machinery An Introduction to the Synthesis and Analysis of Mechanisms and Machines.* McGraw-Hill, 2008.

[68] Lael U. Odhner and Aaron M. Dollar. Dexterous manipulation with underactuated elastic hands. *IEEE International Conference on Robotics and Automation*, 2011.

[69] Lael U. Odhner, Leif P. Jentoft, Mark R. Claffee, Nicholas Corson, Yaroslav Tenzer, Raymond R. Ma, Martin Buehler, Robert Kohout, Robot D. Howe, and Aaron M. Dollar. A compliant, underactuated hand for roboust manipulation. *The International Journal of Robotics Research*, 2014.

[70] Lael U. Odhner, Raymond R. Ma, and Aaron M. Dollar. Precision grasping and manipulation of small objects from flat surfaces using underactuated fingers. *IEEE International Conference on Robotics and Automation*, 2012.

[71] Ryuta Ozawa, Hiroaki Kobayashi, and Kazunori Hashirii. Analysis, classification, and design of tendon-driven mechanisms. *IEEE Transactions on Robotics*, 2014.

[72] Gianluca Palli. *Model and Control of Tendon Actuated Robots.* PhD thesis, Universita degli studi di Bologna, 2006.

[73] Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller Deep-Mind. Data-efficient deep reinforcement learning for dexterous manipulation. *arXiv preprint arXiv:1704.03073*, 2017.

[74] Domenico Prattichizzo, Monica Malvezzi, Marco Gabiccini, and Antonio Bicchi. On the manipulability ellipsoids of underactuated robotic hands with compliance. *Robotics and Autonomous Systems Elsevier*, 2011.

[75] Aravind Rajeswaran, Vikash Kumar, John Schulman Abhishek Gupta, and Sergey Levine Emanuel Todorov. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[76] J. Alfredo Rosas-Flores, Jaime Alvarez-Gallegos, and Rafael Castro-Linares. Control of an underactuated planar 2r manipulator: Experimental results. *International Federation of Automatic Control*, 2002.

[77] Marco Santello, Martha Flanders, and John F. Soechting. Postural hand synergies for tool use. *The Journal of Neuroscience*, 1998.

[78] Jinn-Biau Sheu, Jyun-Jheng Huang, and Jyh-Jone Lee. Kinematic synthesis of tendon-driven robotic manipulators using singular value decomposition. *Robotica*, 2009.

[79] Jinn-Biau Sheu, Tyng Liu, and Jyh-Jone Lee. On the fail-safe design of tendon-driven manipulators with redundant tendons. *Journal of Mechanical Science and Technology*, 2012.

[80] Simon Stepputtis, Yezhou Yang, and Heni Ben Amor. Extrinsic dexteriy through active slip control using deep predictive models. *IEEE International Conference on Robotic Automation*, 2018.

[81] Hannah S. Stuart, Shiquan Wang, Bayard Gardineer, David L. Christensen, Daniel M. Aukes, and Mark Cutkosky. A compliant underactuated hand with suction flow for underwater mobile manipulation. *IEEE International Conference on Robotics Automation*, 2014.

[82] Takahiro Suzuki, Wataru Miyosihi, and Yoshihiko Nakamura. Control of 2r underactuated manipulator with friction. *IEEE Conference on Decision and Control*, 1998.

[83] Lung-Wen Tsai and Jyh-Jone Lee. Kinematic analysis of tendon-driven robotic mechanisms using graph theory. *Journal of Mechanisms, Transmissions, and Automation in Design*, 1989.

[84] M. Uyguroglu and H. Demirel. Kinematic analysis of tendon-driven robotic mechanisms using oriented graphs. *Acta Mechanica*, 2006.

[85] Francisco J. Valero-Cuevas, Felix E. Zajac, and Charles G. Burgar. Large index-fingertip forces are produced by subject-independent patterns of muscle exciation. *Journal of Biomechanics*, 1998.

[86] Herke van Hoof, Nutan Chen, Maximilian Karl, Patrick van der Smagt, and Jan Peters. Stable reinforcement learning with autoencoders for tactile and visual data. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[87] Tucker Hermans Herke van Hoof, Gerhard Neumann, and Jan Peters. Learning robot in-hand manipulation with tactile features. *IEEE-RAS International Conference on Humoanoid Robots*, 2015.

[88] Long Wang, Joseph DelPreto, Sam Bhattacharyya, Jonathan Weisz, and Peter K. Allen. A highly-underactuated robotic hand with force and joint angle sensors. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[89] Jiayin Xie and Nilanjan Chakraborty. Rigid body dynamic simulation with line and surface contact. *IEEE International Converence on Robotic Automation*, 2016.

[90] Wenzeng Zhang, Qiang Chen, and Zhenguo Sun. Passive adaptive grasp multi-fingered humanoid robot hand with high under-actuated function. *IEEE International Conference on Robotics and Automation*, 2004.

[91] Agisilaos G. Zisimatos, Minas V. Liarokapis, Christoforos I. Mavrogiannis, and Kostas J. Kyriakopoulos. Open-source , affordable, modular, light-weight, underactuated robot hands. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.