UNIVERSITY OF TECHNOLOGY, SYDNEY

Robotic Manipulation by Pushing at a Single Point with Constant Velocity: Modeling and Techniques

by
Michael James Behrens

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and IT
Intelligent Mechatronic Systems Group

October 2013

Declaration of Authorship

I, Michael James Behrens, declare that this thesis titled, Robotic Manipulation by Pushing at a Single Point with Constant Velocity: Modeling and Techniques, and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Abstract

In many mobile robotic manipulation tasks it is desirable to interact with the robot's surroundings without grasping the object being manipulated. Non-prehensile manipulation allows a robot to interact in situations which would otherwise be impossible due to object size or mass. This thesis investigates the most general pushing mode, that of a single contact point, formed either as a fixed point at a vertex or as a single rolling contact between two curved surfaces with a view to enable the manipulation of common household objects such as bins or coffee tables by a simple mobile robot. The investigation is limited to objects which possess a flat base and are able to slide on a flat, horizontal support surface. The derivation of a mathematical model is presented for an object pushed under these conditions, where the system accelerations influence the object motion through the dynamic effects of inertia and friction rendering the quasi-static assumption invalid. Numerical simulations explore the system behavior under a variety of configurations revealing the effect of the dynamic forces on the object motion and the existence of stable configurations, under certain conditions, where an object is pushed by a curved fence. The stable pushing behavior is confirmed experimentally. The mathematical model is utilized to generate near time-optimal pushing trajectories to manipulate an object to a desired goal location and control strategies to compensate for uncertainty in the physical parameters.

Acknowledgements

I would like to thank my supervisor Prof. Gamini (*Dissa*) Dissanayake for continually assisting me throughout the course of this project and providing hours of intelligent and receptive interactions which have ultimately led to a more complete, quality thesis.

I would also like to thank the rest of the team at CAS, especially my co-supervisor Dr Shoudong Huang, for always being willing to render assistance whenever it was requested.

I am grateful to the UTS Faculty of Engineering for providing me with computers to work on and access to the High Performance Computing Cluster for my simulations.

Thanks goes also to my family; John, Lueen and Liesl Behrens; Chris Thiele and my good friend Desmond Hokin for their continuous support and tireless proof reading.

Finally, a special thanks goes to my lovely wife, Sharona Behrens, for her unwavering support and encouragement which has undoubtedly enabled the creation of this thesis. Fear not, my dear, you shall be a thesis widow no longer.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Manipulating our surroundings is a task undertaken by humans easily and frequently during daily activities. As robots become increasingly ubiquitous it will be necessary to equip them with manipulation capabilities equivalent to human beings, enabling the robot to operate seamlessly in human environments. The manipulation capabilities required can be broadly divided into two categories, prehensile manipulations such as grasping, where the object motion can be assumed to be fully constrained by the manipulator and non-prehensile manipulations such as pushing, palming or striking, where the object motion is determined by both the manipulator actions and the object's interaction with the environment. No single manipulation technique is applicable to every situation that may be encountered. Instead, a robot must possess a suite of both prehensile and non-prehensile manipulation actions to enable effective operation in a wide variety of scenarios. This thesis focuses on algorithms and approaches to enable non-prehensile manipulation of objects by pushing. In particular, on using the most general interaction case of a single contact point between the pushing robot and the sliding object rather than relying on a specific manipulator geometry or arrangement of contact points to simplify the task. A detailed investigation is undertaken to explore the effect of dynamic forces on the motion of a sliding object and to understand the influence of various parameters, such as the pushing angle, on the object motion. Furthermore, the investigation explores the effect of the geometry of the contact between the object and the robot and examines the potential existence of stable pushing configurations. This thesis also presents techniques for generating sequences of pushing actions to manipulate an object to a desired goal location on a support surface and compensating for variations in the parameters such as the support friction distribution as a pushing action is executed. The outcomes of this thesis are evaluated with a series of numerical experiments simulating a cuboid object being pushed by a cylindrical robot on a flat surface.

1.1 Background

The ability to manipulate the human environment is an essential competency for "social" robots designed to assist in both the home and the workplace. While many of the fundamental issues associated with mobile robotics such as localization and mapping have received significant attention in the past decade [17] and are now well understood, manipulation in unstructured environments is still seen as a significantly challenging task [41]. One of the principle obstacles to successful manipulation in unstructured environments is the wide variety of tasks and scenarios which can be encountered during operation. Whereas a manufacturing robot is usually designed to repeat a limited set of tasks which are very well understood, a mobile robot operating in an unstructured environment may encounter a multitude of different tasks and environmental conditions under which these tasks must be performed [22]. This level of task variability precludes the use of preprogrammed actions to achieve the objectives. Instead, the robot must reason about the task mechanics during operation and generate a sequence of actions to achieve the objectives as the situation demands it.

In many situations encountered by a mobile robot, the primary method of manipulating the surrounding environment is by utilizing a prehensile manipulation technique such as grasping. The key benefit of using a grasping strategy is that once a stable grasp capable of resisting all expected external forces has been achieved, the manipulator actions required to move the object to the goal can be computed without considering the object-environment interaction, effectively decoupling the two phases of the manipulation action [34]. It is inevitable, however, that a robot will eventually encounter a situation where the objectives cannot be achieved by grasping alone. This problem can arise in situations where the object being manipulated is too large, heavy or awkward to grasp and lift, the robot gripper is already holding something else [9] or is partially disabled. In these cases, non-prehensile manipulation techniques are able to fill the gap in the robot capability and enable the robot to achieve an objective that would be otherwise impossible [40]. Sometimes, a task which can be accomplished using a prehensile "pick and place" action may be completed more effectively using non-prehensile manipulation actions. An example [20] is using sweeping actions to arrange scattered pencils rather than picking up each individual pencil in turn. Recent work by Dogar and Srinivasa [13–15], combines a non-prehensile pushing phase with a traditional grasping action to reduce uncertainty in the object pose and improve the successful grasp rate. These authors also present a framework for relocating clutter by pushing thus enabling a grasping action to be undertaken.

Manipulation of objects on the micro and nano scale using both pushing and grasping strategies are active research areas. However, at this scale the object behavior is not dominated by gravity and friction but instead is influenced greatly by short range forces

such as Van-der-Waals and electrostatic forces [50], and as such fall outside the scope of the work presented in this thesis.

1.2 Motivation

This thesis investigates a subset of non-prehensile manipulation, namely, manipulation of an object free to slide on a support surface by pushing with a contact between the robot and object made at a single point. One of the key challenges involved with this manipulation technique is the fact that the motion of the sliding object is not completely defined by the motion of the robot. Instead, the object motion is determined by a combination of the forces exerted by the robot and the interaction forces between the object and the environment [37, 38]. Early work towards solving this problem, particularly in the manufacturing context, focused on designing an end-effector with a specialized geometry so that the task mechanics will ensure that the object being manipulated reaches a stable configuration where it is stationary relative to the robot [31]. In the context of mobile robotics, a coordinated team of robots [18] has also been used for non-prehensile manipulation by making contact at multiple locations around the object perimeter, thus allowing the pose of the object to be constrained.

In contrast, the focus of the work presented in this thesis is to investigate the motion of a sliding object when pushed at a single point, to develop algorithms that generate near time optimal trajectories to manipulate this object to a desired goal position and orientation and to formulate strategies to compensate for uncertainty in parameters such as the support friction distribution so as to ensure successful delivery of the object to the goal.

The accurate and controlled manipulation of a sliding object with a single point of contact is a very difficult task. The difficulty of this task is largely due to the under-actuated nature of the system formed and the fact that a pushing action cannot be undone by simply performing the action in reverse. To overcome these challenges, it is essential that the robot possesses a good understanding of how an object will respond when acted upon. This information may be used to assemble a sequence of actions which will result in the object moving to a desired position and orientation. Unfortunately, due to the uncertainty in the support friction distribution, it is highly unlikely that an object will respond precisely in the manner predicted prior to the manipulation being commenced. To account for this, the robot must sense deviations from the planned trajectory and respond appropriately to ensure that the object can still be successfully manipulated to the desired goal.

1.3 Scope

This research aims to endow a mobile robot with the ability to manipulate a sliding object to a desired goal by pushing with a single point of contact under a variety of environmental conditions and manipulation speeds. To achieve this, variations in the motion of an object with a flat base sliding on a flat surface are studied for various object masses, moments of inertia, support locations, support friction coefficients and pushing surface radii. Particular emphasis is given to investigating the effect of varying pushing speeds with a view to enable rapid and time efficient manipulations. This understanding of the expected object response to a pushing action is used to determine near time optimal trajectories for manipulating a sliding object to a desired goal position and orientation in the support plane. The final aims of this research are to represent the near time optimal trajectories such that the information may be readily accessed for use in real time control applications and to implement techniques to sense and respond to variations in object motion due to uncertainty in the environmental parameters.

There are a number of assumptions made to limit the scope of this thesis.

1. It is assumed that the object being manipulated is of known mass and geometry and that it is free to slide on a horizontal support surface.
2. It is also assumed that the interaction between the object and the supporting surface will obey Coulomb's Law of friction.
3. The robot is assumed to contact the object at a single point, that being a sharp vertex on either the robot or object perimeter or a point of contact between two curved surfaces.
4. It is assumed that the object is being pushed by the robot, that is, the center of friction of the object is ahead of the pushing contact point in the direction of pushing.
5. It is assumed that the robot and the object are in contact at all times.
6. The robot is assumed to move with constant speed and be of sufficient power to maintain the requested pushing trajectory.
7. It is also assumed that there are no obstacles in the support plane that must be avoided.

1.4 Contributions

In the following, a list of contributions arising from the thesis is presented:

- Investigation of the behavior of an object with a flat base sliding on a flat surface when pushed at a point by a mobile robot under both quasi-static and dynamic conditions.
- Discovery and investigation of regions of "stable" motion when an object with a flat base sliding on a flat surface is pushed by a mobile robot with a curved fence.
- Synthesis of near time-optimal trajectories for delivering an object with a flat base sliding on a flat surface to a goal position and orientation.
- Development of a novel strategy for controlling the motion of an object with a flat base sliding on a flat surface for real time execution which is robust to variation in the environmental parameters.

1.5 Publications

The following publications by the author relate to material presented in Chapters 2 and 3 of the thesis respectively:

- "Models for pushing objects with a mobile robot using single point contact," Behrens, M., Shoudong, H., Dissanayake, G., *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2010)*, Taipei, Taiwan, Oct. 2010.
- "Stability Analysis of Pushing Objects with a Curved Fence," Behrens, M., Dissanayake, G., Manuscript to be submitted for publication in Robotica

Other publications by the author:

- "Designing SANDRA: An Autonomous Tour Guide Robot for the University of Technology, Sydney," Behrens, M., Carmichael M., Patel M., *Proc. of the Australasian Conference on Robotics and Automation (ACRA 2008)*, Canberra, Australia, Dec. 2008.

1.6 Thesis Overview

Chapter 2 investigates the motion of an object sliding on a flat support surface in response to a constant velocity push at a single contact point with a view to enabling non-prehensile manipulation by a mobile robot. The investigation explores two system topologies: arrangements where the pushing contact is made at a single sharp point and arrangements where the pushing contact is made between two curved surfaces free to roll along one another. The investigation also examines system topologies displaying self-stabilizing behaviors where pushing contact is made between two curved surfaces. These stable regions are assessed for suitability for application using currently available techniques for manipulation by stable pushing. The full dynamic equations of motion are derived and implemented in simulation to explore the system response to a range of parameter variations including pushing velocity, object mass, moment of inertia, support friction coefficient and support point location. The causes and ramifications of the observed effects are analyzed and discussed. The stable pushing behavior is verified experimentally.

Chapter 3 presents a methodology for deriving near time optimal trajectories that overcome the limitations imposed by the under-actuated nature of the system and enable the delivery of sliding objects to a desired goal position and orientation. The planning task is posed as a constrained optimization problem and is solved using the full dynamic system model derived in Chapter 2. Selected numerical case studies explore the impact of the number of partitions used in the piecewise constant approximation of the input vector, the effect of variation in pushing velocity and the effect of variation in the support friction distribution when planning near time optimal trajectories. The effect of the pushing robot radius when planning trajectories for systems with contact formed between two curved surfaces is also explored. A sensitivity study analyzes the effect of variation in key system parameters on the performance of the open loop trajectories generated by the optimization.

Chapter 4 implements a methodology for the rapid selection of control actions in a manner suitable for real time applications and explores techniques which compensate for uncertainty in environmental parameters such as friction distribution in practical applications. The optimizer presented in Chapter 3 is used to generate many near time-optimal trajectories to position an object at the origin from a wide range of starting poses. A control law that interpolates between these trajectories using Delaunay triangulation to generate an instantaneous control action that corresponds to the current object pose is obtained. The proposed system is implemented in simulation to determine the capabilities and limitations of the technique.

Chapter 5 concludes with a summary of outcomes presented and a discussion of the future of related research.

Chapter 2

Behavior of a Sliding Object Pushed at Constant Velocity

There are many situations where a robot needs to be able to manipulate the surrounding environment and it is not always possible or desirable to achieve this by grasping the target object. Non-prehensile manipulation can serve to expand the capabilities of a robot system in many ways, such as enabling the manipulation of objects too large or heavy to grasp and lift, performing certain actions such as clearing space on a shelf more effectively or providing limited functionality in the event that a gripper fails.

The most significant challenge when performing non-prehensile manipulation is that the movement of the object is often not completely defined by the robot motion. Instead, the object motion is usually influenced by both the robot motion and the interaction between the object and the environment. As a result, any robotic manipulation system which aims to provide general non-prehensile manipulation capability needs to include an understanding of how the object-environment interaction influences the object motion which is the purpose of this chapter. In particular, it investigates the behavior of an object sliding on a uniform flat support surface while being pushed at constant velocity by a mobile robot. While an uneven friction distribution on the support surface would more closely match what may be expected in a practical scenario, the variable support friction would mask the dynamic effects which are the focus of this investigation.

This chapter begins with a review of the previous work in non-prehensile manipulation which is largely concerned with stable pushing under quasi-static conditions. Following this, the full dynamic models describing the motion of an object as it is pushed by a robot are derived. Several configurations including situations where contact between the robot and object is at a single fixed point and the more general case where contact is formed by two surfaces and the object is free to roll along the robot edge are considered. Finally, the mathematical models are implemented in simulation and a variety of case studies

simulated and analyzed to investigate the influence of the environmental parameters on the object motion.

2.1 Literature Review

When a side of a polygonal object is aligned and in contact with a flat fence it is possible to control both the position and orientation of the object by manipulating the pose of the fence. Mason [39] was the first to propose a simple rule for determining the direction of rotation of an object when there is a single point of contact between the flat fence and the object. Using knowledge of the object geometry, the pushing location and the center of friction of the support surface, Mason predicted whether an object would rotate towards a stable orientation with a flat fence or not. Akella, Mason and Lynch [4, 31] went on to develop open loop pushing plans to reduce the orientation uncertainty of a sliding part, so as to guarantee that a part may be fed into the system with an arbitrary orientation but will always exit in a known configuration. Brokowski, Peshkin and Goldberg [7] proposed that a curved section be added to the end of the flat fence to reduce the orientation uncertainty of the object as it breaks contact with the fence. These techniques are particularly suited to the task of parts feeding in industrial applications [2, 3]. They may also be applied to objects being manipulated by a mobile robot equipped with a flat fence. Again, manipulation is achieved by executing a sequence of stable pushes to manipulate the position and orientation of the sliding object. Unlike the parts feeding application, where the object was delivered to the same goal orientation every time without knowledge of the initial orientation, the mobile robot application usually requires the object to be delivered to a variety of goal locations from a starting position and orientation which is known, but may have some uncertainty. An analysis of the part geometry allows the reachable regions to be determined [32]. Further work, again by Lynch and Mason, [36] [33] [30] extended the flat fence methodology to use a number of point contacts to create a virtual fence and allow objects with curved or irregular surfaces to be manipulated. The strength of this methodology is that suitable margins of error may be included so that the effect of support uncertainty is not significant in the determination of the object motion allowing open loop plans to be generated which may be executed without any sensing or feedback. These systems, however, must be designed for a specific task and rely on a fixed object geometry. If a different part needs to be manipulated then an entirely new sequence of stable pushes must be developed.

Another technique for dealing with the unpredictability of the object orientation is the manipulation of objects by coordinated teams of robots [29] [48] [52] [42]. This creates a virtual fence between the contacts and kinematically constrains the motion of the sliding object. Once again a series of stable pushing commands can be generated to manipulate an object to a desired goal configuration. The significant challenges in this

field are related to the task of coordinating and communicating between the robots involved, deciding how decentralized control strategies can be derived and the extent that information needs to be communicated between the robots to accomplish desired behavior.

When the inertial effects are negligible and hence the motion of the object is dominated by the support friction, Kurisu and Yoshikawa [24] hypothesized that an object pushed at a single point will eventually reach a limiting angular velocity where the object motion will be unchanged as long as the pushing speed and direction are maintained constant relative to the object. The limiting angular velocity is a function of the pushing angle, making the object behave in a manner similar to a non-holonomic wheeled vehicle. A validation that the quasi-static model approximates well to the real behavior was not presented. However, in [25] the utility of this approximation was demonstrated through an experimental evaluation of a trajectory following technique for single point pushing.

While the research reviewed here addresses many of the challenges faced in the difficult task of non-prehensile manipulation, there are several research areas which have not, as yet, been studied in the published literature. The remainder of this chapter begins with an investigation of a sliding object's behavior when pushed at a single point under conditions such that the dynamic forces cannot be neglected. The resulting dynamic equations of motion allow a validation of the steady state behavior under quasi-static conditions hypothesized by Kurisu and Yoshikawa to be performed. This chapter also investigates a sliding object's behavior when pushed with a curved fence, in particular, the existence of stable pushing configurations is demonstrated in spite of the fact that contact is made at a single point only.

2.2 Pushing using a Point Contact

Consider a planar object sliding on a support surface in response to a constant velocity push at a single point by a robot as shown in Figure 2.1. The case of pushing with constant velocity is considered because velocity based locomotion is both common and easy to implement and understand. This approach also produces some interesting object behaviors and affords several simplifications during planning and control. It is assumed that the object is being pushed by a robot with sufficient power to impart the required force while maintaining the requested pushing trajectory. As the object is able to rotate relative to the pusher, the object motion is not fully defined by the movement of the pusher. Instead, the motion of the sliding object is governed by the support friction and external forces exerted on it. The pushing force will be determined by the motion of the body and the friction forces imparted by the object environment interface.

A list of notations used in the derivation is given in Table 2.1. For the purpose of the analysis presented in this chapter, as in [24], it is assumed that the support distribution

may be approximated to a finite set of known support points and that the magnitude of the friction force, the support geometry, mass and inertia are fixed. It is also assumed throughout the analysis that all support points are in motion at all times and that the friction conforms to Coulomb's Law of friction. If there are more than three support points the problem becomes statically indeterminate. The individual normal reaction forces at each of the support points and associated friction forces are then computed by assuming that the support pressure distribution is constant. The final assumption is that the pushing contact is a single point and that the pushing contact friction is sufficient such that the contact point does not slide along the perimeter of the body. While it is obvious that all of these conditions will not be met in a practical situation, the focus of this chapter is on obtaining an insight to explain the behavior of an object under these conditions.

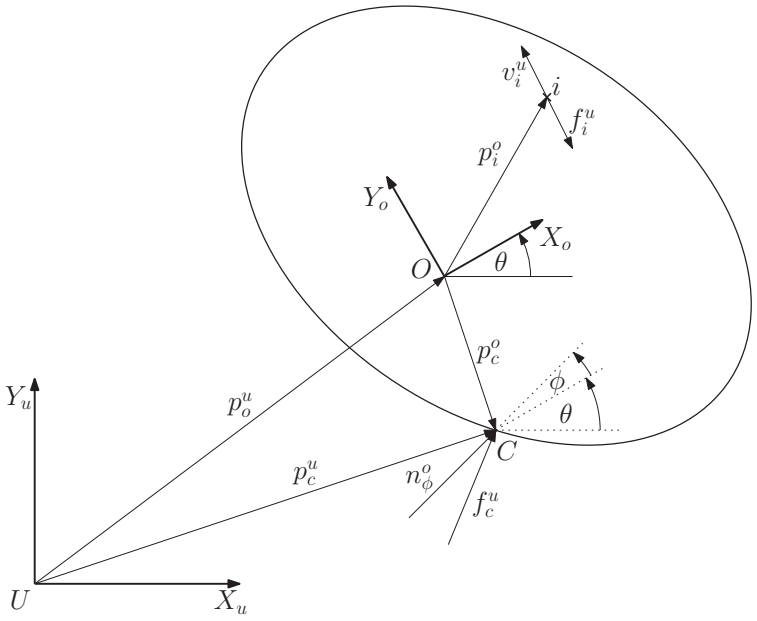


FIGURE 2.1: A sliding object O subject to an input velocity v_c at a single point C

U_{xy} is a global inertial reference frame and O_{xy} is an object coordinate frame, fixed at the center of mass and rotating with the object. The position of O_{xy} and hence the position of the object center of mass relative to U_{xy} is given by

$$\boldsymbol{p}_o^u = \boldsymbol{p}_c^u - R_o^u \boldsymbol{p}_c^o \tag{2.1}$$

where R_o^u is the rotation matrix

$$R_o^u = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.2}$$

to transform vectors from the object frame to the global frame.

Symbol	Description
$\boldsymbol{p}_c^u$	position of the contact point in the global frame
$\boldsymbol{p}_o^u$	origin of the object coordinate frame in the global frame
$\boldsymbol{p}_c^o$	position of the contact point in the object frame
$\boldsymbol{p}_i^o$	position of the ith support point in the object frame
θ	rotation angle of the object coordinate frame in the global frame
ϕ	direction of the pushing velocity in the object frame
v_c	magnitude of the pushing velocity
$\boldsymbol{f}_c^u$	pushing force vector in the global frame
$\boldsymbol{v}_i^u$	velocity vector at ith support point in the global frame
$\boldsymbol{f}_i^u$	friction force vector at ith support point in the global frame
n	number of support points
N_i	normal reaction force at ith support point
μ_i	coefficient of friction at ith support point
m	mass of the sliding object
I_c	moment of inertia about the pushing contact point
ω	angular velocity of the sliding object
$\boldsymbol{n}_\phi^o$	unit vector indicating the pushing velocity direction in the object frame

TABLE 2.1: Notations used in Dynamic and Quasi-static Models

Differentiating (2.1) gives

$$\dot{\boldsymbol{p}}_o^u = \dot{\boldsymbol{p}}_c^u - R_o^u \dot{\boldsymbol{p}}_c^o - \dot{\theta} R_o^u D \boldsymbol{p}_c^o \tag{2.3}$$

with

$$D = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2.4}$$

As the pushing contact point is fixed in the object frame, $\dot{\boldsymbol{p}}_c^o = 0$.
Therefore, (2.3) simplifies to

$$\dot{\boldsymbol{p}}_o^u = \dot{\boldsymbol{p}}_c^u - \dot{\theta} R_o^u D \boldsymbol{p}_c^o \tag{2.5}$$

When there is no slip at the contact point

$$\dot{\boldsymbol{p}}_c^u = v_c R_o^u \boldsymbol{n}_\phi^o \tag{2.6}$$

where $\boldsymbol{n}_\phi^o$ defines the direction of the pushing velocity

$$\boldsymbol{n}_\phi^o = \begin{bmatrix} \cos\phi \\ \sin\phi \end{bmatrix} \tag{2.7}$$

Combining (2.5) and (2.6) provides the velocity of the center of mass

$$\dot{\boldsymbol{p}}_o^u = v_c R_o^u \boldsymbol{n}_\phi^o - \dot{\theta} R_o^u D \boldsymbol{p}_c^o \tag{2.8}$$

Differentiating (2.8) we obtain the acceleration of the center of mass

$$\ddot{\boldsymbol{p}}_o^u = v_c \dot{\theta} R_o^u D \boldsymbol{n}_\phi^o - \ddot{\theta} R_o^u D \boldsymbol{p}_c^o - \dot{\theta}^2 R_o^u D^2 \boldsymbol{p}_c^o \tag{2.9}$$

Considering the friction forces, inertial forces, and the inertial moment, taking a moment balance about the contact point and rearranging produces

$$0 = \sum_{i=1}^{n} (R_o^u (\boldsymbol{p}_i^o - \boldsymbol{p}_c^o) \times \boldsymbol{f}_i^u) + m R_o^u \boldsymbol{p}_c^o \times \ddot{\boldsymbol{p}}_o^u - \ddot{\theta} \begin{bmatrix} 0 \\ 0 \\ I_c \end{bmatrix} \tag{2.10}$$

where I_c is the moment of inertia about the contact point and $\boldsymbol{f}_i^u$ is the friction force at the i^{th} support point given by

$$\boldsymbol{f}_i^u = \frac{-\boldsymbol{v}_i^u}{\|\boldsymbol{v}_i^u\|} \mu_i N_i \tag{2.11}$$

where $\boldsymbol{v}_i^u$ is the velocity of the i^{th} support point in the global frame given by

$$\boldsymbol{v}_i^u = v_c R_o^u \boldsymbol{n}_\phi^o + \omega R_o^u D (\boldsymbol{p}_i^o - \boldsymbol{p}_c^o) \tag{2.12}$$

and

$$\omega = \dot{\theta} \tag{2.13}$$

Combining (2.9 - 2.13) provides a complex function for $\ddot{\theta}$ which, for clarity, is denoted simply as

$$\ddot{\theta} = \dot{\omega} = F(\phi, \omega, \theta) \tag{2.14}$$

The pushing force is given by

$$\boldsymbol{f}_c^u = m \ddot{\boldsymbol{p}}_o^u - \sum_{i=1}^{n} \boldsymbol{f}_i^u \tag{2.15}$$

Combining (2.9) and (2.15) yields

$$\boldsymbol{f}_c^u = m(v_c\dot{\theta}R_o^u D\boldsymbol{n}_\phi^o - \ddot{\theta}R_o^u D\boldsymbol{p}_c^o - \dot{\theta}^2 R_o^u D^2\boldsymbol{p}_c^o) - \sum_{i=1}^{n}\boldsymbol{f}_i^u \tag{2.16}$$

The dynamic equations of the system can be written as

$$\begin{cases} \dot{x} = v_c\cos(\theta+\phi) + \omega(x_c^o\sin\theta + y_c^o\cos\theta) \\ \dot{y} = v_c\sin(\theta+\phi) - \omega(x_c^o\cos\theta - y_c^o\sin\theta) \\ \dot{\theta} = \omega \\ \dot{\omega} = F(\phi,\omega,\theta) \end{cases} \tag{2.17}$$

where $\boldsymbol{p}_o^u = [x\ y]^T, \boldsymbol{p}_c^o = [x_c^o\ y_c^o]^T$

Considering the motion of an object sliding under conditions where the effect of inertial forces due to the object mass and moment of inertia are negligible in comparison to the support friction forces, Kurisu and Yoshikawa [24] began with the assumption that the angular velocity of an object reaches a steady state and used this as the basis for a trajectory planning strategy, although no conclusive proof was presented to support this assumption.

The key conclusion of the analysis in [24] is that the steady state angular velocity is a function of the pushing angle, ϕ and is directly proportional to the pushing speed v_c

$$\dot{\theta} = v_c\psi(\phi) \tag{2.18}$$

with the kinematic equations describing the motion of the object shown to be

$$\begin{cases} \dot{x} &= v_c\cos(\theta+\phi) + v_c\psi(\phi)(x_c^o\sin\theta + y_c^o\cos\theta) \\ \dot{y} &= v_c\sin(\theta+\phi) - v_c\psi(\phi)(x_c^o\cos\theta - y_c^o\sin\theta) \\ \dot{\theta} &= v_c\psi(\phi) \end{cases} \tag{2.19}$$

where $\boldsymbol{p}_o^u = [x\ y]^T, \boldsymbol{p}_c^o = [x_c^o\ y_c^o]^T$.

It is clear that the simplified equations, given in Equation 2.19, are strikingly similar to the full dynamic equations given by Equation 2.17. The difference is that for the full dynamic equations, the angular velocity of the object will evolve as a function of time as expected, while under the quasi-static approximation, angular velocity is only a function of the pushing input angle resulting in an object behavior similar to that of a non-holonomic wheeled robot. It is important to note that the direction of the pushing velocity in the global frame needs to be continuously adjusted to maintain a constant ϕ.

In subsequent sections, a series of numerical simulations are undertaken which demonstrate that a steady state angular velocity is indeed reached if ϕ is maintained to be a constant. Furthermore, as the significance of the inertial forces is reduced, the steady state angular velocity approaches that of the simplified scenario, thus verifying that the hypothesis proposed by Kurisu and Yoshikawa is valid.

2.3 Pushing Objects using a Curved Fence

Pushing objects using a single fixed point of contact can be considered a subset of the more general case where pushing contact is made between two curved surfaces free to roll along each other, as shown in Figure 2.2. Once an initial contact point and pushing direction are selected, it becomes desirable to understand what motions will be experienced by the object during the pushing action. Simple experimentation reveals that in the vast majority of cases the object will begin rotating, rolling around the robot perimeter as the robot advances. Once the object is in motion it may continue to rotate about the robot until it "rolls off" and contact is lost or the object may reach an equilibrium condition at some position around the robot perimeter. In this equilibrium position the object may cease rotating and translate in a direction parallel to the direction of the robot motion.

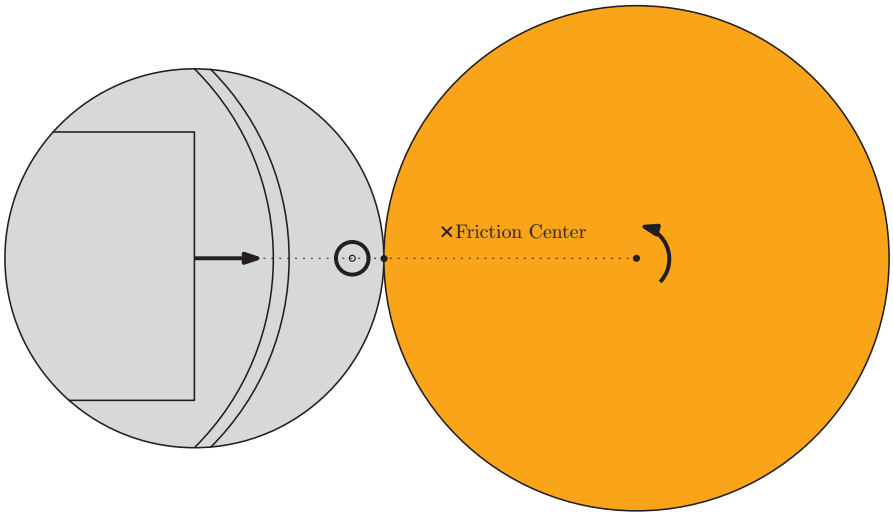


FIGURE 2.2: A sliding object in an initial state

This section investigates whether such equilibrium positions exist, and if so, determines the geometric conditions required to obtain a steady state. Unless stated otherwise, the pushing contact point is assumed to be perfectly rough such that no sliding occurs between the robot and object, as defined by Lynch and Mason [35]. The analytical investigation is limited to the quasi-static case for simplicity. However, both quasi-static and dynamic conditions are evaluated by simulation in Section 2.5. The derivation of the dynamic equations of motion is detailed in Appendix C.

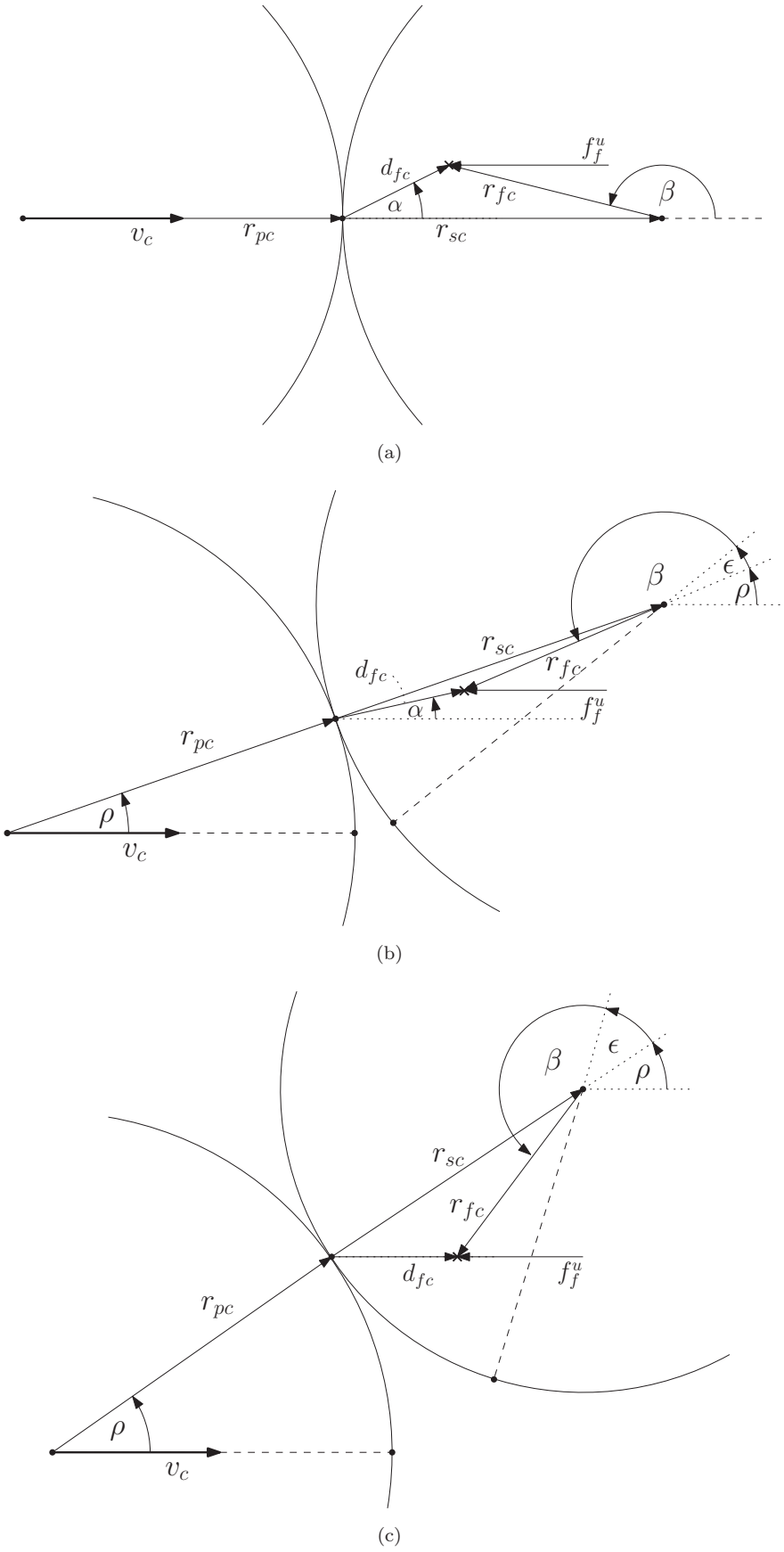


FIGURE 2.3: A pushing system (a) initial (b) intermediate and (c) at rest

Under quasi-static conditions, the rotation of the sliding object is determined by the location of the center of friction and the moment that the support friction force makes around the pushing contact point. The location of the center of friction may be determined analytically by calculating the centroid of the support friction distribution, if it is known [24], or the centroid of the support pressure distribution if the co-efficient of the support friction is uniform [47]. If the location of the center of friction is assumed to be constant relative to the object frame it is possible to determine its location experimentally by performing several pushing actions around the object perimeter and observing the resulting object motion [53].

Figure 2.3 illustrates a scenario where an object of radius $\boldsymbol{r}_{sc}$ is pushed by a circular robot of radius $\boldsymbol{r}_{pc}$. The top figure illustrates the initial condition of the pushing system where the robot is pushing directly from left to right. Since the friction center is located above the pushing line, a counter-clockwise moment is generated by the support friction forces which results in the object rotating counter-clockwise about the robot as the pushing action progresses. The middle figure illustrates an intermediate condition where the object has begun to move relative to the robot but has not yet reached an equilibrium condition. The bottom figure illustrates the final steady state condition of the pushing system. In this arrangement, the pushing contact point and the friction center are collinear in the direction of pushing resulting in a zero moment acting on the object. Furthermore, any small disturbance in the object rotation will cause a restoring moment to be generated which will drive the system back to the steady state.

The moment acting on the object is given by

$$M_c = \boldsymbol{d}_{fc} \sin\alpha \times \|\boldsymbol{f}_f^u\| \tag{2.20}$$

where $\boldsymbol{d}_{fc}$ is the vector from the contact point to the friction center, α is the angle between this vector and the pushing vector and $\|\boldsymbol{f}_f^u\|$ is the magnitude of the total support friction force.

From this equation it is apparent that the sign of the moment can be determined simply by considering the angle formed between a vector from the contact point to the friction center and the pushing velocity vector, denoted α in Figure 2.3. Furthermore, an equilibrium condition is formed when both the contact point and the friction center are co-linear in the direction of pushing, resulting in an α value of zero and a corresponding zero moment about the contact point as shown in Figure 2.3(c).

An expression for α can be calculated as follows.

The vector from the contact point to the friction center is given by

$$\boldsymbol{d}_{fc} = \boldsymbol{r}_{sc} + \boldsymbol{r}_{fc} \tag{2.21}$$

$$\begin{bmatrix} d_{fc}\cos\alpha \\ d_{fc}\sin\alpha \end{bmatrix} = \begin{bmatrix} r_{sc}\cos\rho + r_{fc}\cos(\rho+\epsilon+\beta) \\ r_{sc}\sin\rho + r_{fc}\sin(\rho+\epsilon+\beta) \end{bmatrix} \tag{2.22}$$

where

$$\epsilon = \frac{\|\boldsymbol{r}_{pc}\|}{\|\boldsymbol{r}_{sc}\|}\rho \tag{2.23}$$

Since

$$\tan\alpha = \frac{d_{fc}\sin\alpha}{d_{fc}\cos\alpha} = \frac{r_{sc}\sin\rho + r_{fc}\sin(\rho+\epsilon+\beta)}{r_{sc}\cos\rho + r_{fc}\cos(\rho+\epsilon+\beta)} \tag{2.24}$$

Therefore

$$\alpha = \arctan\left(\frac{r_{sc}\sin\rho + r_{fc}\sin(\rho+\epsilon+\beta)}{r_{sc}\cos\rho + r_{fc}\cos(\rho+\epsilon+\beta)}\right) \tag{2.25}$$

The value of ρ at equilibrium can be computed by setting α to zero in Equation 2.25 and solving the resulting nonlinear equation.

Stable pushing systems, which result in a slider motion that is robust to variations in the friction distribution thus producing a constant or stable motion, have been known to exist where two flat surfaces contact one another or where two points contact with a surface. As described in Section 2.1, these systems have been studied extensively and techniques for assembling strings of "stable pushes" have been developed to allow objects to be manipulated to a desired position and orientation without sensing equipment. If this phenomenon can be predicted in systems where pushing contact is made with a curved fence, these same stable pushing techniques may be exploited in a larger suite of applications. To determine whether a particular equilibrium point is stable or unstable, the response to a small disturbance $\delta\rho$ must be inspected. If the resulting couple acts to restore the system to the rest state then the equilibrium point may be considered stable.

Disturbance Tolerance

To investigate the level of disturbance which may be tolerated, the initial pushing location and direction is selected so that the system begins in a stable equilibrium condition with the friction center and the pushing contact point aligned in the direction of pushing as shown in Figure 2.4. If the object experiences a small disturbance, a restoring moment will be generated according to Equation 2.20. If the disturbance is sufficiently large it is possible that the moment generated will no longer act to restore the system to rest and the system stability will be lost. This system behaves in a manner similar to an eccentrically weighed wheel which is free to roll on an arched track under the influence of gravity as shown in Figure 2.5.

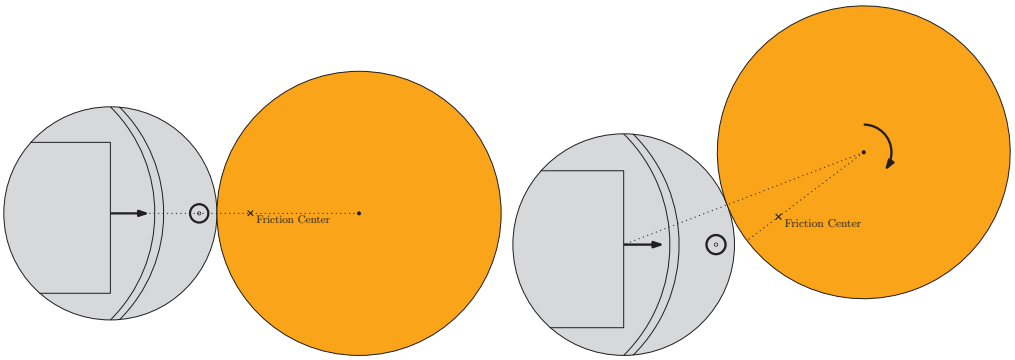


FIGURE 2.4: A sliding object in a (a) steady state and (b) disturbed state

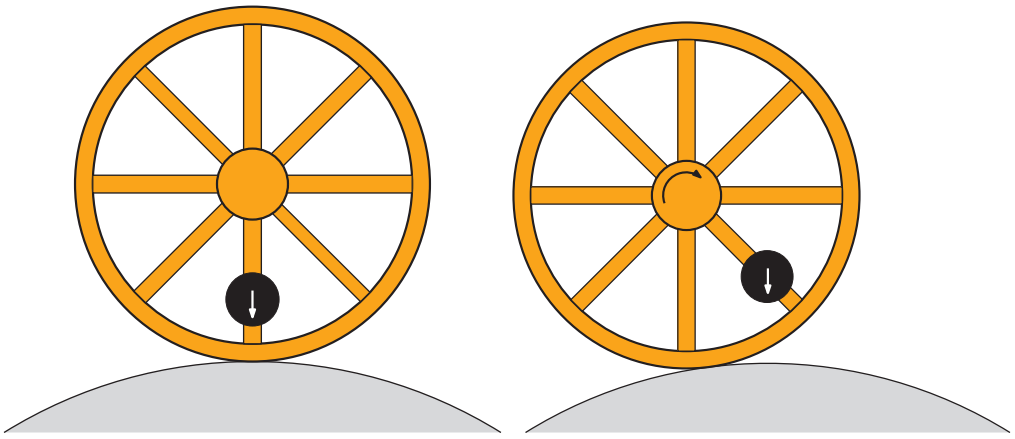


FIGURE 2.5: A rolling wheel in a (a) steady state and (b) disturbed state

As discussed earlier, the sign of the restoring moment is governed by the value of α. Figure 2.6 shows the variation of α as the slider is rolled about the pusher to various values of ρ for a specific geometry as defined in the figure caption. These parameters represent a $400mm$ diameter, eccentrically weighted object being manipulated by a mobile robot with a gently curving bumper of $500mm$ radius. While the object radius is small relative to the robot radius in this example, this is not a requirement for stability

as will be demonstrated by the subsequent manipulation of a flat sided (infinitely large radius) object. Three equilibrium points can clearly be seen as zero crossings at $\rho = -0.5$, $\rho = 0$ and $\rho = 0.5$. In order for an equilibrium point to be stable, the slope of the curve at that point must be negative, such that a small positive displacement $\delta\rho$ results in a negative α while a small negative displacement $-\delta\rho$ results in a positive α. If these conditions are satisfied then the object will return to the equilibrium condition when disturbed. In Figure 2.6, this condition holds true for the equilibrium point at $\rho = 0$, while the other equilibrium points are unstable. For the region $-0.5 \leq \rho \leq 0.5$, the moment generated will act to restore the system to the stable equilibrium point at $\rho = 0$. Although this is somewhat counter-intuitive, if the object is disturbed close to, but not past, the unstable equilibrium point it will return to the stable equilibrium point. In this example, if the object is disturbed to $\rho = -0.4$ a positive restoring moment will be generated which will drive the object back to $\rho = 0$, slowly at first and then more rapidly as the local maximum at $\rho = -0.25$ is reached before slowing again to settle at $\rho = 0$. For the region outside $-0.5 \leq \rho \leq 0.5$, the moment generated acts to drive the system further away from the stable equilibrium. As such, if the object is disturbed to or beyond either of the unstable equilibrium positions, control of the object will be lost.

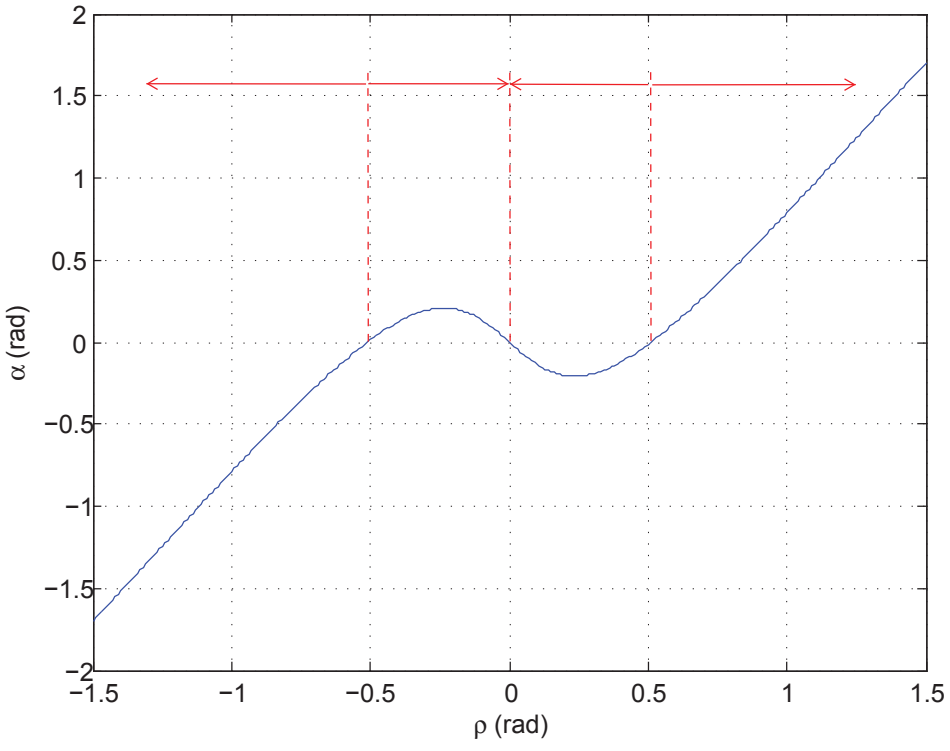


FIGURE 2.6: Variation of α with respect to ρ for system with $\boldsymbol{r}_{pc} = 0.5m$, $\boldsymbol{r}_{sc} = 0.2m$, $\boldsymbol{r}_{fc} = 0.1m$ and $\beta = \pi$. Arrows indicate the direction of object rotation for a given ρ

If the radius of the pusher is sufficiently large relative to the slider radius, there is the potential for multiple stable equilibrium points to exist, as demonstrated in Figure 2.7. Seven equilibrium points can be seen as zero crossings at $\rho = -0.75$, $\rho = -0.66$, $\rho = -0.25$, $\rho = 0$, $\rho = 0.25$, $\rho = 0.66$ and $\rho = 0.75$ with $\rho = -0.66$, $\rho = 0$ and $\rho = 0.66$ satisfying the requirements for stable equilibrium points. In this case, each stable equilibrium point is separated by an unstable equilibrium point and the stable regions for each stable equilibrium point will extend up to the unstable equilibrium point. As a result, the overall stable region will encompass several individual stable regions.

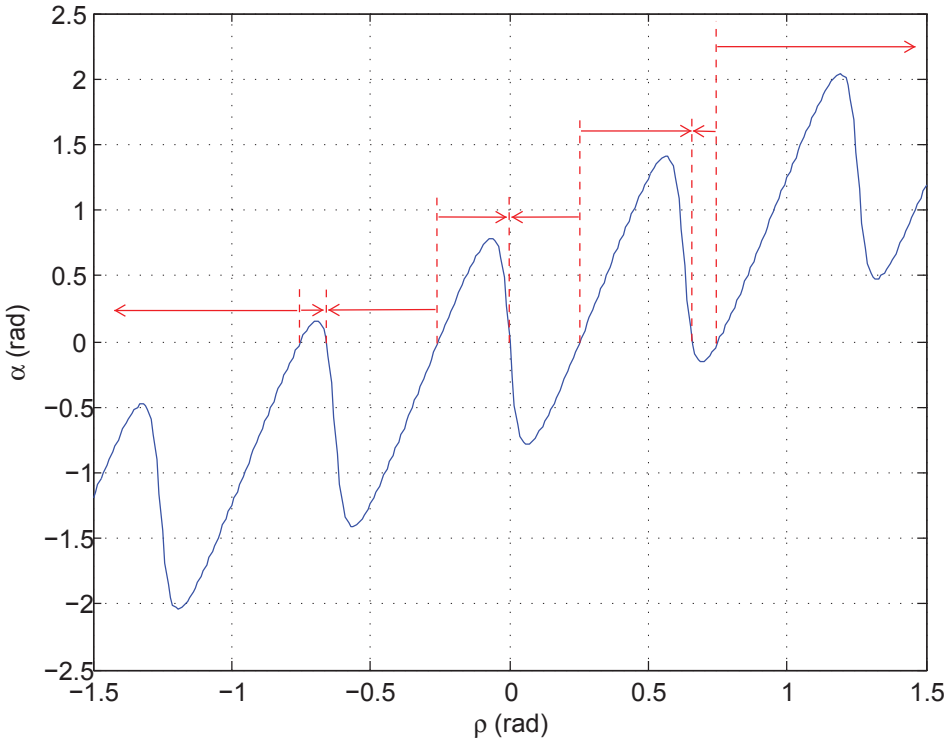


FIGURE 2.7: Variation of α with respect to ρ for system with $\boldsymbol{r}_{pc} = 2.0m$, $\boldsymbol{r}_{sc} = 0.2m$, $\boldsymbol{r}_{fc} = 0.15m$ and $\beta = \pi$. Arrows indicate the direction of object rotation for a given ρ

Effect of Contact Coefficient of Friction

All of the analysis thus far assumes a perfectly rough contact between the pusher and slider ensuring that a rolling contact is maintained until the pushing direction becomes tangential to the slider surface. If the contact friction is limited to some finite value, the rolling contact will transition to a sliding contact at some earlier point as shown in Figure 2.8. Any equilibrium points located outside the rolling contact range will no longer be reachable and the stable regions, if any, will extend only to the limit of the rolling contact range.

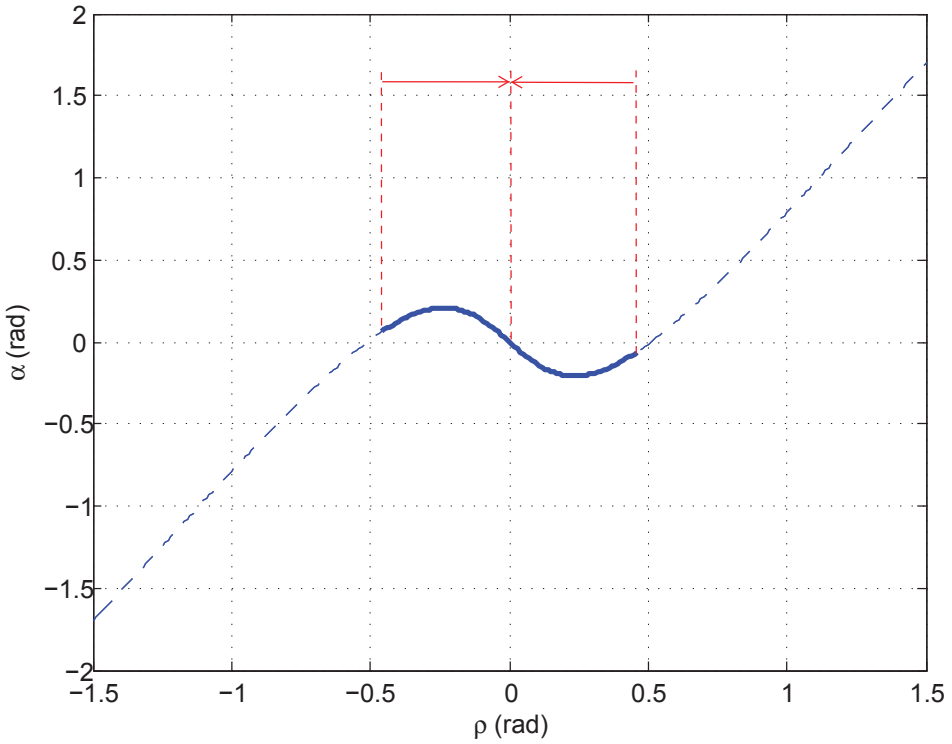


FIGURE 2.8: Variation of α with respect to ρ with a contact friction coefficient of 0.5 showing the rolling (solid) and sliding (dashed) regions.
$\boldsymbol{r}_{pc} = 0.5m$, $\boldsymbol{r}_{sc} = 0.2m$, $\boldsymbol{r}_{fc} = 0.1m$ and $\beta = \pi$.
Arrows indicate the direction of object rotation for a given ρ

Manipulation using a Flat Fence

When the pusher radius is infinite, that is, the object is pushed with a flat fence, ρ is always zero. In this situation, the linear displacement along the fence defines the location of the contact point. Equation 2.25 now simplifies to

$$\alpha = \arctan\left(\frac{-r_{fc}\sin(\epsilon+\beta)}{r_{sc} - r_{fc}\cos(\epsilon+\beta)}\right) \tag{2.26}$$

where

$$\epsilon = \frac{\boldsymbol{d}_{pc}}{\|\boldsymbol{r}_{sc}\|} \tag{2.27}$$

Figure 2.9 shows the variation in α as the object rolls along the fence, with equilibrium points at $\boldsymbol{d}_{pc} = -1.26$, $\boldsymbol{d}_{pc} = -0.62$, $\boldsymbol{d}_{pc} = 0$, $\boldsymbol{d}_{pc} = 0.62$ and $\boldsymbol{d}_{pc} = 1.26$, of which $\boldsymbol{d}_{pc} = -1.26$, $\boldsymbol{d}_{pc} = 0$ and $\boldsymbol{d}_{pc} = 1.26$ are stable equilibrium points. If there is no unstable equilibrium point between a stable equilibrium and the end of the pushing fence, the stable region will extend up to the end of the pushing fence. However, if the stable equilibrium point and the end of the fence are separated by an unstable equilibrium point, the stable region will extend up to the unstable equilibrium point only. It is interesting to note that due to the cyclic nature of the response, it is guaranteed that there will be at least one stable equilibrium position provided that the fence is at least one slider circumference in length. If the friction center is located at the exact geometric center of the sliding object then the system will always be marginally stable as no restoring moments will be generated. As a result, the system will not recover from a disturbance but neither will it continue to rotate away.

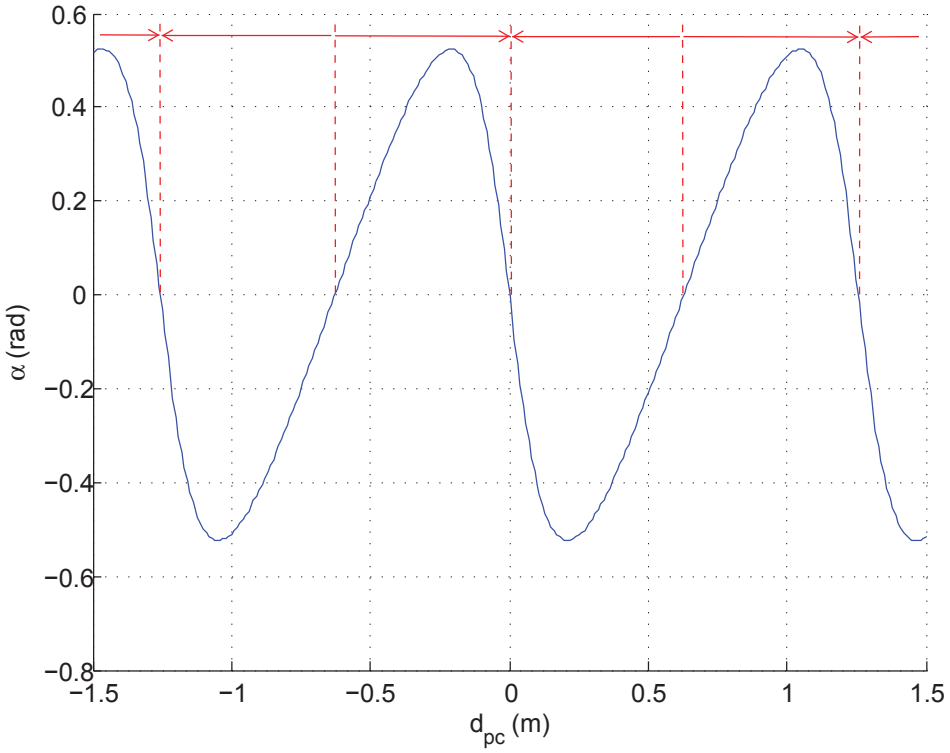


FIGURE 2.9: Variation of α with respect to $\boldsymbol{d}_{pc}$ for pushing with a flat fence $\boldsymbol{r}_{sc} = 0.2m$, $\boldsymbol{r}_{fc} = 0.1m$ and $\beta = \pi$.
Arrows indicate the direction of object rotation for a given ρ

Manipulation of a Flat Sided Object

Another case which can be considered is pushing a flat face of an object with a curved fence as illustrated in Figure 2.10. In this configuration we can consider the object face to have an infinite radius of curvature.

The vector from the contact point to the friction center is given by

$$\boldsymbol{d}_{fc} = \boldsymbol{n}_{fc} + \boldsymbol{t}_{fc} - \boldsymbol{t}_{pc} \tag{2.28}$$

where $\boldsymbol{t}_{fc}$ is the tangent and $\boldsymbol{n}_{fc}$ is the normal to the friction center, and

$$\|\boldsymbol{t}_{pc}\| = \|\boldsymbol{r}_{pc}\|\rho \tag{2.29}$$

Expanding Equation 2.28 gives

$$\begin{bmatrix} d_{fc}\cos\alpha \\ d_{fc}\sin\alpha \end{bmatrix} = \begin{bmatrix} n_{fc}\cos\rho + t_{fc}\cos(\rho+\frac{\pi}{2}) - r_{pc}\rho\cos(\rho+\frac{\pi}{2}) \\ n_{fc}\sin\rho + t_{fc}\sin(\rho+\frac{\pi}{2}) - r_{pc}\rho\sin(\rho+\frac{\pi}{2}) \end{bmatrix} \tag{2.30}$$

Since

$$\begin{aligned} \tan\alpha &= \frac{d_{fc}\sin\alpha}{d_{fc}\cos\alpha} \\ &= \frac{n_{fc}\sin\rho + t_{fc}\sin(\rho+\frac{\pi}{2}) - r_{pc}\rho\sin(\rho+\frac{\pi}{2})}{n_{fc}\cos\rho + t_{fc}\cos(\rho+\frac{\pi}{2}) - r_{pc}\rho\cos(\rho+\frac{\pi}{2})} \\ &= \frac{n_{fc}\sin\rho + t_{fc}\cos\rho - r_{pc}\rho\cos\rho}{n_{fc}\cos\rho - t_{fc}\sin\rho + r_{pc}\rho\sin\rho} \end{aligned} \tag{2.31}$$

Therefore

$$\alpha = \arctan\left(\frac{n_{fc}\sin\rho + t_{fc}\cos\rho - r_{pc}\rho\cos\rho}{n_{fc}\cos\rho - t_{fc}\sin\rho + r_{pc}\rho\sin\rho}\right) \tag{2.32}$$

The perpendicular distance from the pushing surface to the center of friction and the radius of the robot are key to determining the stability of the system. If the perpendicular distance to the center of friction is less than the robot radius of curvature the system will possess a stable equilibrium point and an associated stable region, as shown in Figure 2.11. As the perpendicular distance to the friction center increases relative to the robot radius, the stable region will reduce until only a marginally stable system is formed at the equilibrium point, as shown in Figure 2.12. This will occur when the perpendicular distance to the friction center is equal to the robot radius. If the distance

to the friction center is larger than the robot radius the system will not exhibit any stable equilibrium points, as shown in Figure 2.13.

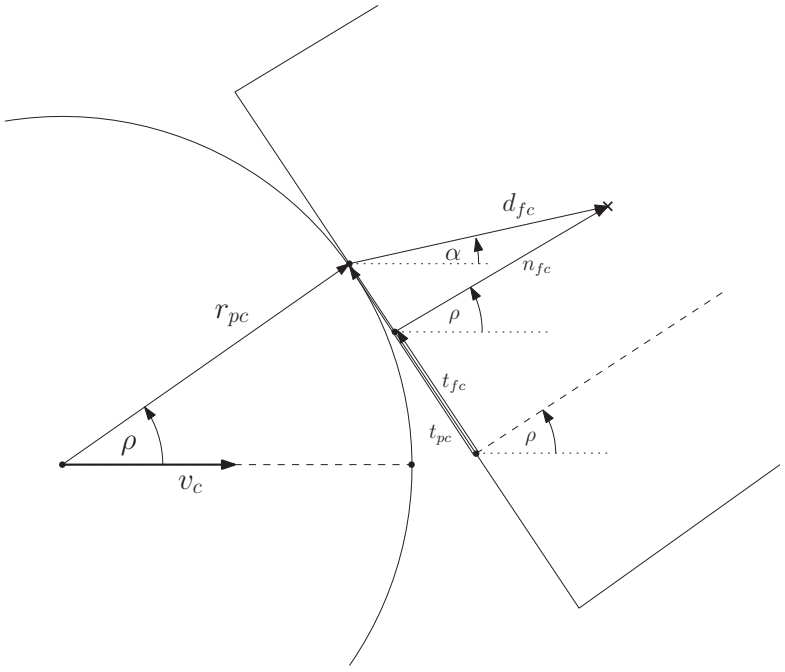


FIGURE 2.10: Calculation of α for pushing a flat faced object with a curved fence

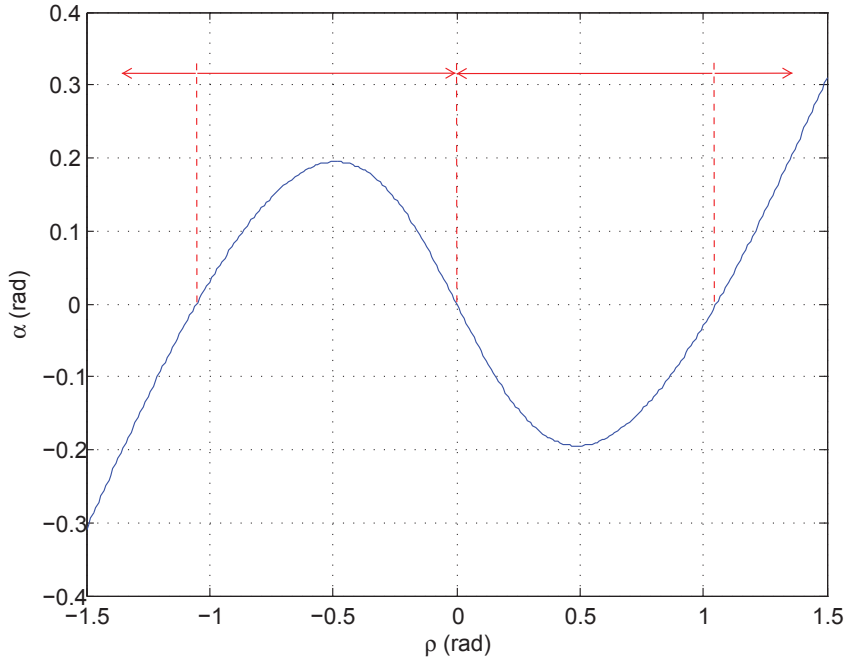


FIGURE 2.11: Variation of α with respect to ρ for pushing a flat object with $\boldsymbol{r}_{pc} = 0.5m$, $\boldsymbol{n}_{fc} = 0.3$ and $\boldsymbol{t}_{fc} = 0$.
Arrows indicate the direction of object rotation for a given ρ

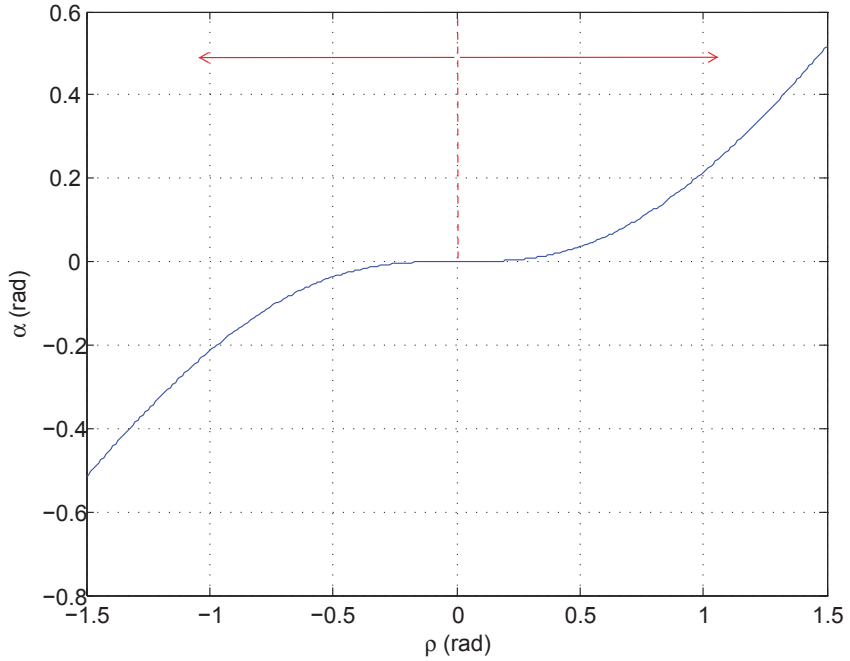


FIGURE 2.12: Variation of α with respect to ρ for pushing a flat object with $\boldsymbol{r}_{pc} = 0.5m$, $\boldsymbol{n}_{fc} = 0.5$ and $\boldsymbol{t}_{fc} = 0$.
Arrows indicate the direction of object rotation for a given ρ

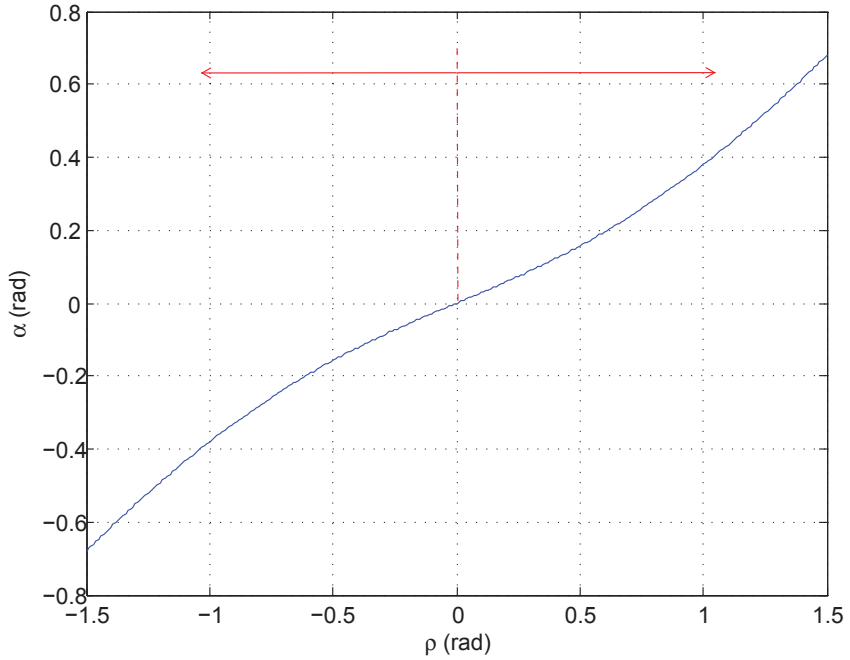


FIGURE 2.13: Variation of α with respect to ρ for pushing a flat object with $\boldsymbol{r}_{pc} = 0.5m$, $\boldsymbol{n}_{fc} = 0.7$ and $\boldsymbol{t}_{fc} = 0$.
Arrows indicate the direction of object rotation for a given ρ

Pushing objects with Offset Friction Centers

If the friction center is offset from the initial pushing line, as shown in Figure 2.3(a), the equilibrium points will no longer be symmetrically arranged about an equilibrium point at $\rho = 0$. Instead the α *vs* ρ curve will be offset, as shown in Figure 2.14, with the magnitude and direction of the offset dependent on the location of the friction center. In this situation, the effective slope of the pushing surface will be different for a positive and negative disturbance. This causes the stable region to obtain different extents in each direction, resulting in an asymmetrical response. In this case, the stable equilibrium point is located at $\rho = 0.25$ *rad*. It is interesting to note that the stable region extends significantly further in the negative ρ direction, up to $\rho = -0.43$ *rad*, compared to the positive ρ direction which only extends to $\rho = 0.40$ *rad*.

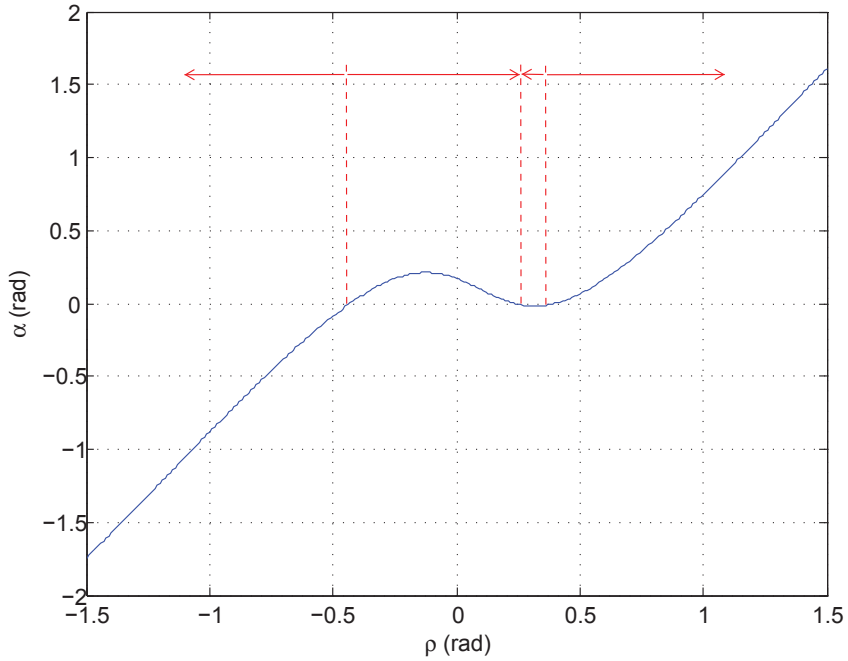


FIGURE 2.14: Variation of α with respect to ρ for system with $\boldsymbol{r}_{pc} = 0.5m$, $\boldsymbol{r}_{sc} = 0.2m$, $\boldsymbol{r}_{fc} = 0.085m$ and $\beta = 2.9$. Arrows indicate the direction of object rotation for a given ρ

In a system where dynamic forces are significant, although the system will settle in the same stable equilibrium position, there will often be some overshoot as the object reaches the stable equilibrium position as shown in Figure 2.16. If this overshoot causes the object to rotate beyond the limits of the stable region the object will not return to the stable equilibrium position. Instead, the object will continue to rotate about the robot until contact is lost, as shown in Figure 2.17. However, multiple simulations using conditions typically experienced in practice show that, in most cases, the object will remain in the stable region and will settle to a steady state. Thus, the dynamic forces result in a very minor narrowing of the stable region. Considering this, the stability analysis of the non-linear dynamic system is excluded from this current investigation.

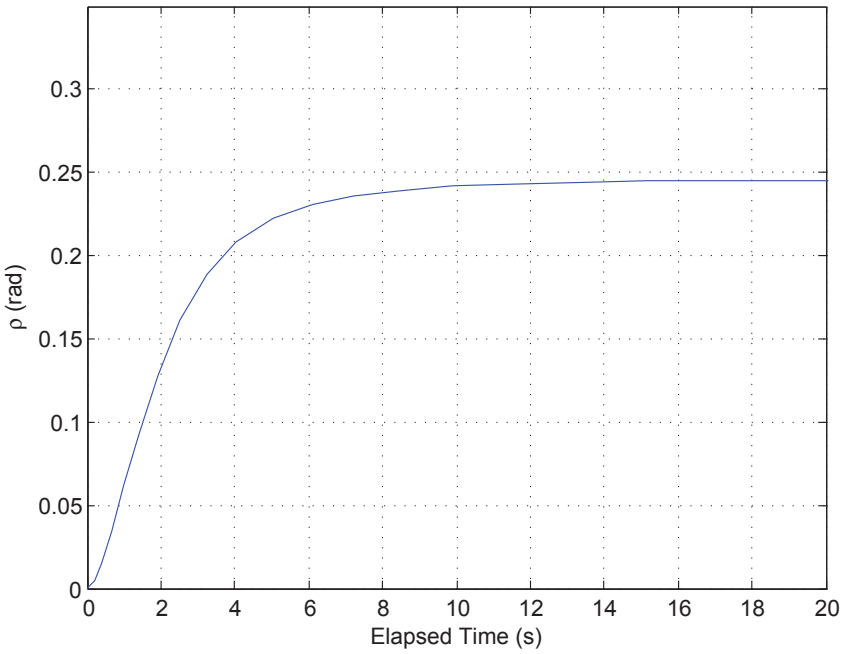


FIGURE 2.15: Variation of ρ with $v_c = 0.5m/s$

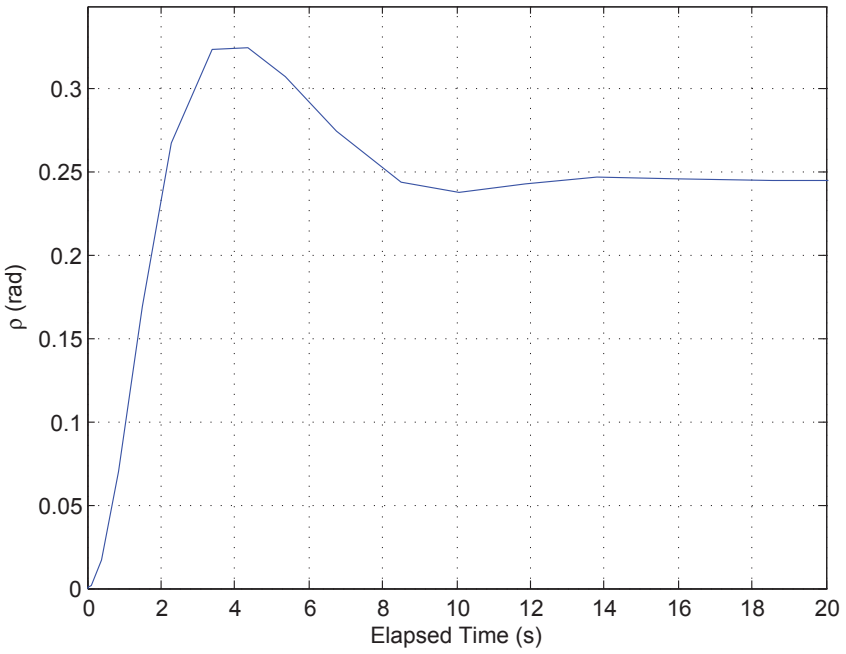


FIGURE 2.16: Variation of ρ with $v_c = 1.5m/s$

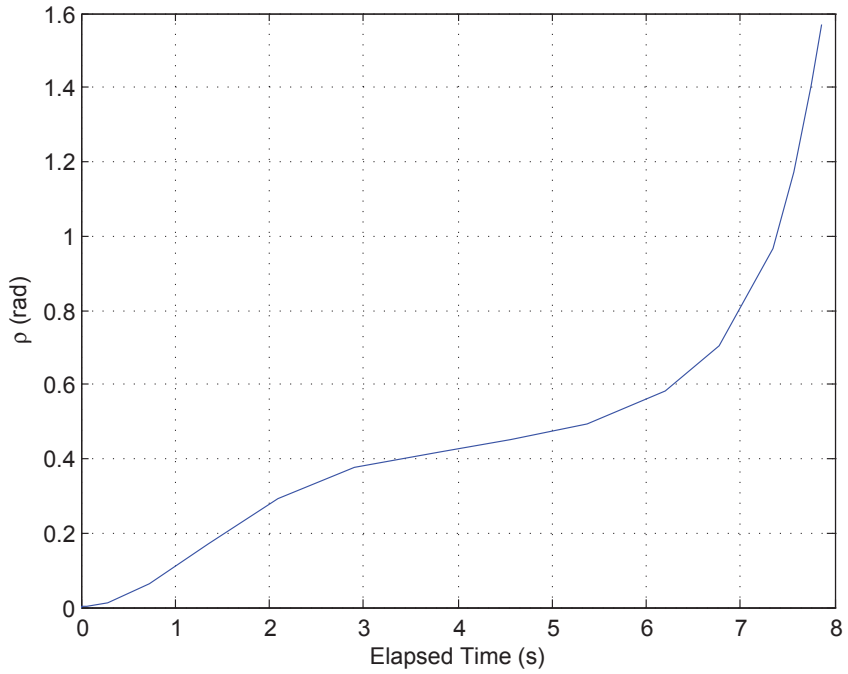


FIGURE 2.17: Variation of ρ with $v_c = 2.5m/s$

2.4 Case Study 1: Pushing with Constant Velocity at a Fixed Point

This section presents the results of a series of computer simulations employing a full dynamic model, which accounts for all of the dynamic effects of motion, to further evaluate the behavior of an object being pushed by a robot as presented in Section 2.2. A cuboid object sliding on a uniform flat surface while being pushed at constant velocity by a cylindrical robot as shown in Figure 2.18 with the default parameters given in Table 2.2 is used in the study. As the robot is pushing at one of the object corners and a perfectly rough pushing contact is assumed, the system will have a single fixed point of contact throughout the manipulation when considered as a 2D system.

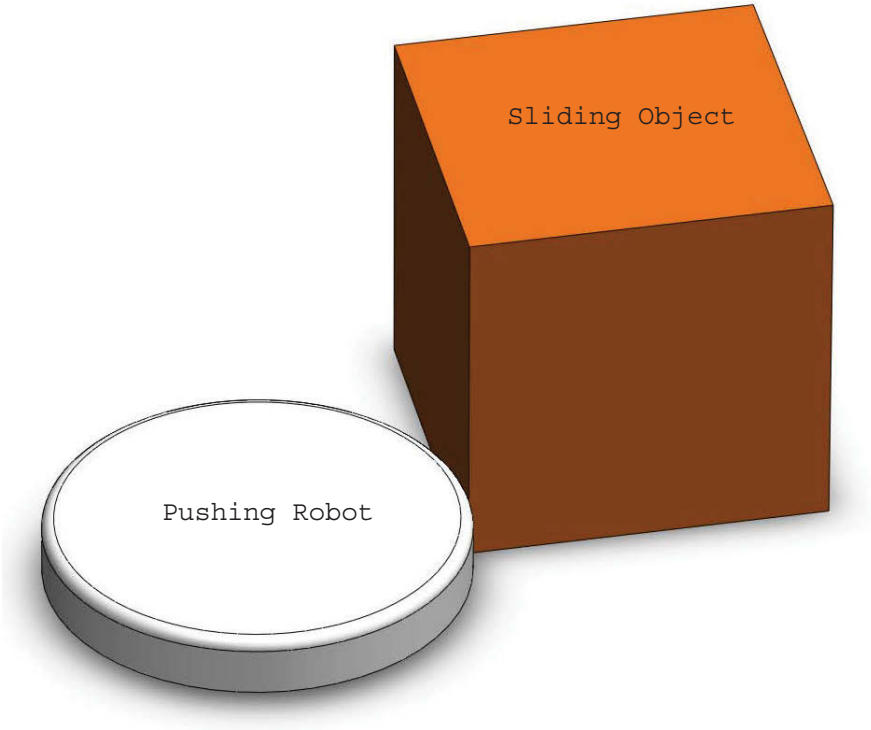


FIGURE 2.18: Round robot pushing a cuboid object at a corner resulting in a point contact

Object Parameter	Value
Friction Coefficient Between Object and Support Surface	0.3
Support Contact Pattern Side Length	$0.3m$
Number of Support Contacts	4
Object Mass	$2kg$
Object Side Length	$0.3m$
Object Moment of Inertia	$0.03kg/m^2$

TABLE 2.2: Default Physical Parameters for Numerical Simulations

2.4.1 Effect of maintaining a constant pushing angle relative to the object coordinate frame

For the configuration where the robot pushes at one of the object corners, the object will undergo pure translation if the direction of the pushing passes directly through the center of friction and the center of mass. However, for a more likely configuration where the pushing direction does not exactly pass through both the center of friction and the center of mass, the object will begin to rotate as well as translate as it is pushed. An interesting observation from the quasi-static analysis culminating in Equation 2.19 is that if the pushing velocity angle ϕ is maintained constant relative to the object coordinate system then the object eventually reaches a steady state condition and will travel in a circular trajectory. Figure 2.20 shows that this is true. The fact that a steady state will eventually be reached when ϕ is maintained constant is the key to enabling practical control strategies for the manipulation of objects. In situations where the quasi-static assumption is valid the steady state angular velocity varies linearly with respect to the pushing velocity. Whereas, in systems where the inertial effects are significant there is a period of transient response before the steady state is reached and the steady state angular velocity varies non-linearly with respect to the input velocity. Figure 2.19 compares the steady state angular velocities for the example system. It can be seen that the approximation based on the quasi-static assumption gives a linear change in angular velocity. While the dynamic calculation corresponds to the quasi-static analysis closely at very low pushing speeds but becomes increasingly divergent as the pushing speed increases.

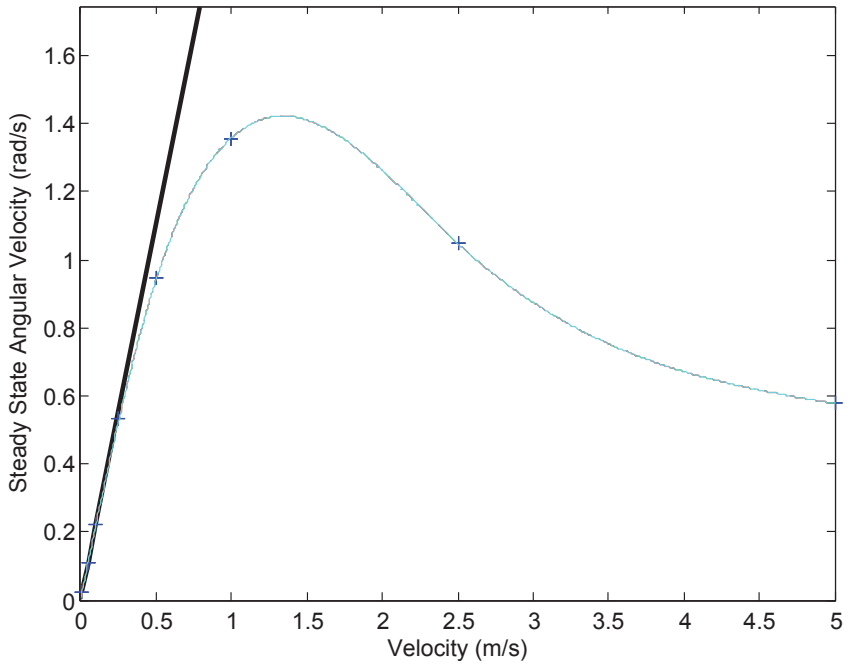


FIGURE 2.19: Comparison of angular velocities predicted under the quasi-static assumption with steady state angular velocities reached when system inertial effects are considered and the pushing angle is maintained constant

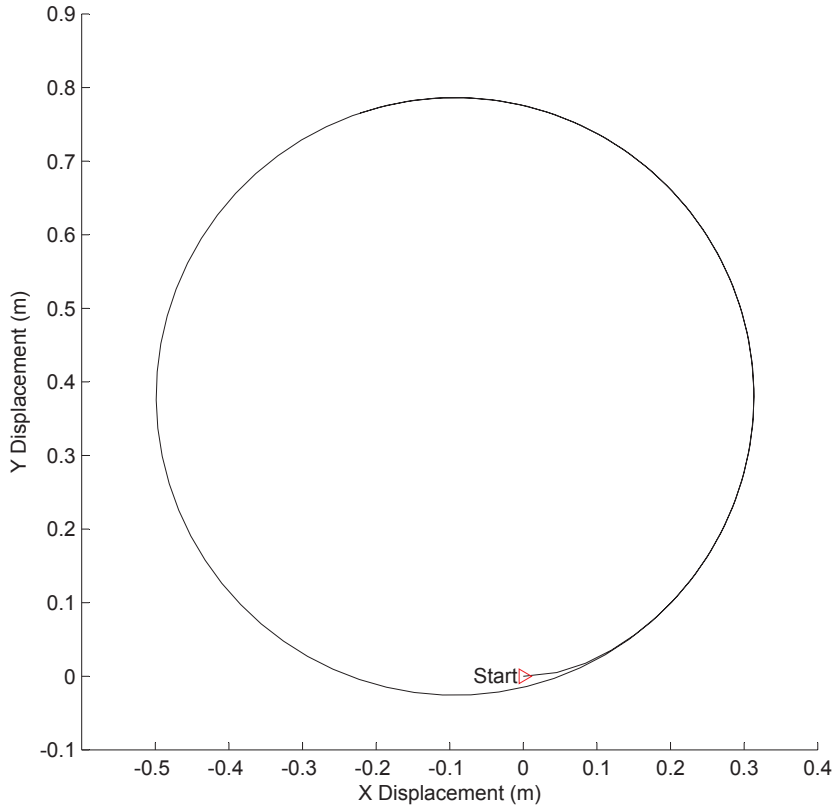


FIGURE 2.20: Motion of an object sliding in the presence of friction when pushed at constant velocity

2.4.2 Effect of Mass Variation

This section considers the variation in the response of an object pushed at constant velocity when the density is varied. A variety of simulations with densities giving object masses between 0.5kg and 3kg are shown in Figure 2.21 and Figure 2.22. It is interesting to note that both the static and dynamic responses are independent of the object mass. Qualitatively, this can be explained by considering the support friction. Since the support friction is assumed to conform to Coulombs Law of friction, the support friction forces and the dynamic forces all increase in proportion. As a result, the moment balance around the contact point is unchanged and hence the angular acceleration is unchanged. Overall this demonstrates that when the object is pushed with a constant velocity the mass of the object has no bearing on the way that the object behaves.

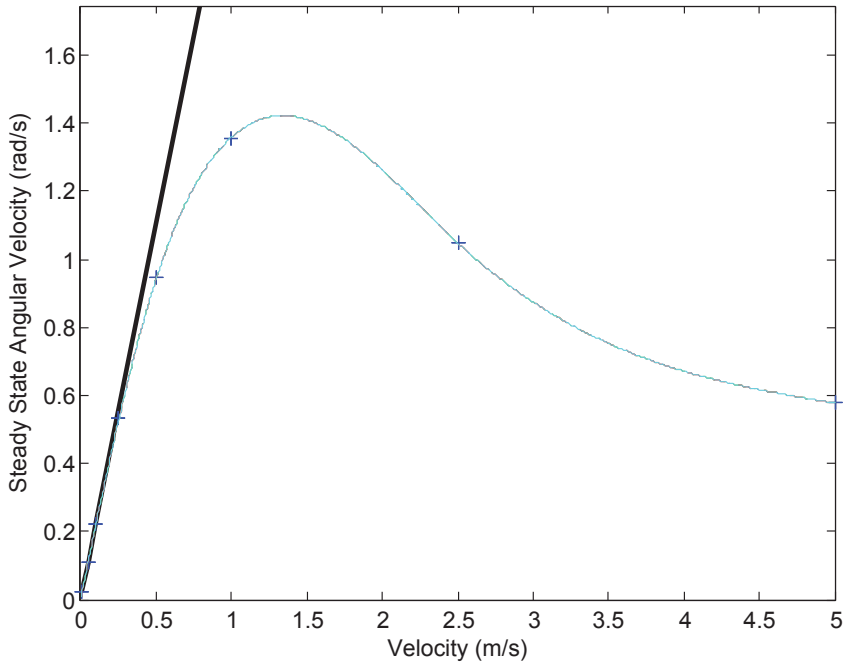


FIGURE 2.21: Comparison of the angular velocity with pushing velocity for masses from 0.5 to 3kg

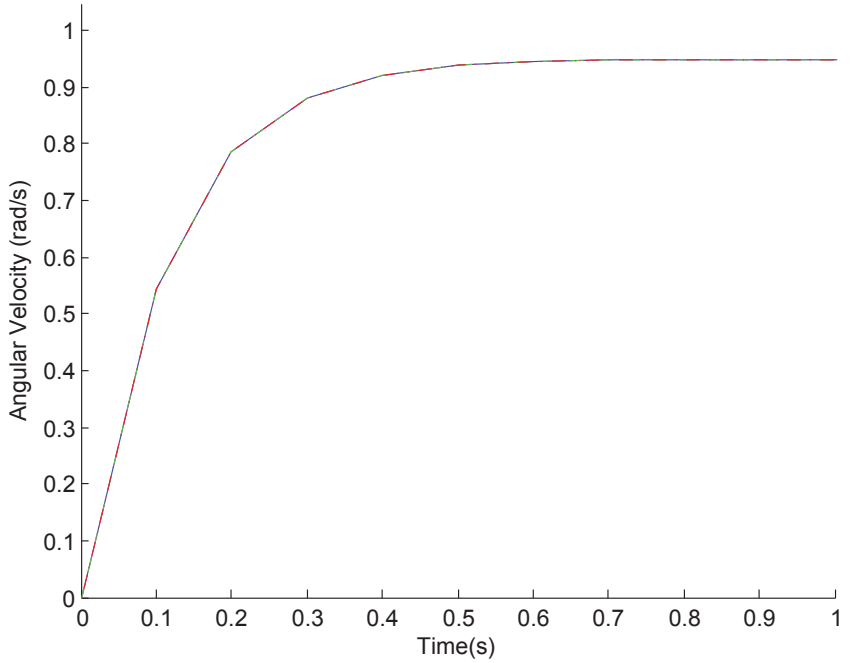


FIGURE 2.22: Transient response of the angular velocity for masses from 0.5 to 3kg

2.4.3 Effect of Friction Coefficient Variation

This section considers the variation of the friction coefficient between an object pushed at constant velocity and the support surface. Figure 2.23 shows the result for a selection of friction coefficients ranging from normal friction levels [43] to almost negligible friction. Again the dynamic response diverges from the quasi-static approximation as the pushing velocity increases and reducing the friction coefficient causes the response to diverge sooner and faster. This can be expected because changing the friction coefficient will alter the support friction forces but leave the inertial forces relatively unchanged. It is also apparent that increasing friction coefficients lead to larger steady state angular velocities for a given pushing velocity.

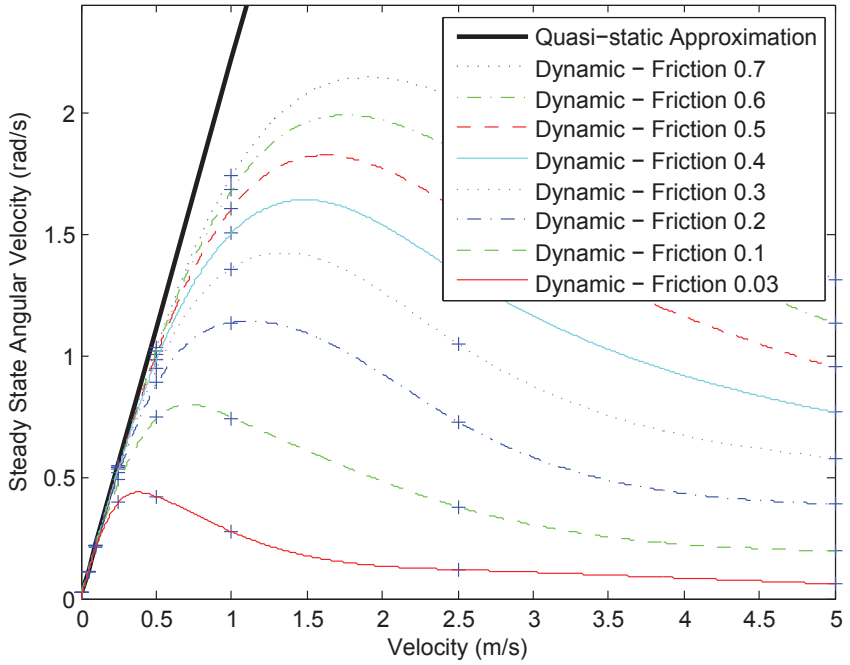


FIGURE 2.23: Comparison of the angular velocity with pushing velocity at various support friction coefficients

2.4.4 Effect of Moment of Inertia Variation

This section investigates the variation in object motion when the moment of inertia is varied by altering the size and density of the object while keeping the location of the support points and the object mass constant. The moments of inertia presented here are computed from objects with side lengths ranging from $0.1m$ to $0.5m$ in $0.1m$ increments with a mass of $3kg$. For a constant pushing velocity the steady state angular velocity response, shown in Figure 2.24, is unchanged. This is understandable, as the rotary inertia of the object does not have an effect on the steady state angular velocity. The object rotary inertia will, however, affect the transient response of the angular velocity as shown in Figure 2.25. As expected, objects with lower moments of inertia are able

to achieve a higher angular acceleration for a given constant velocity input and hence reach a steady state faster than objects with greater moments of inertia.

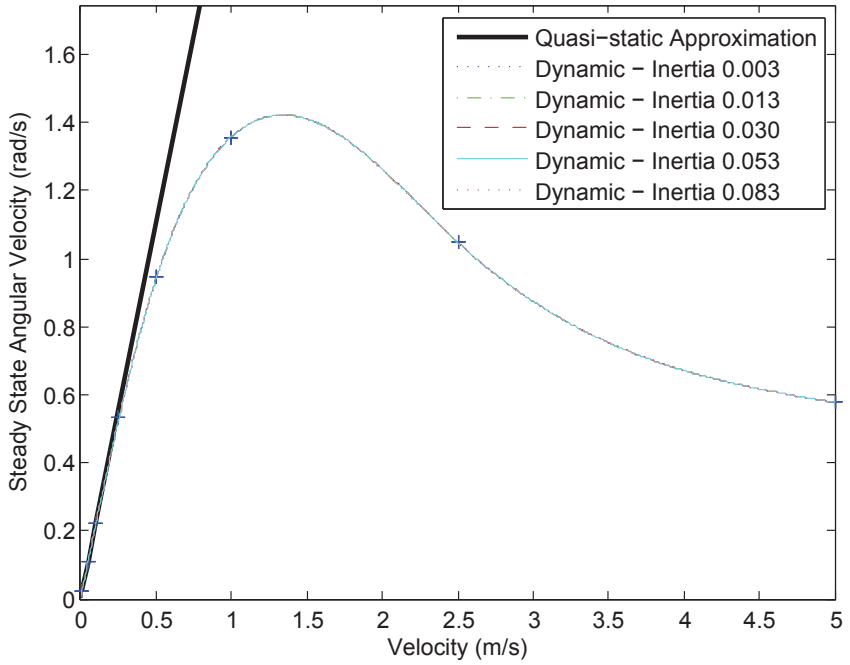


FIGURE 2.24: Comparison of the angular velocity with pushing velocity at various moments of inertia

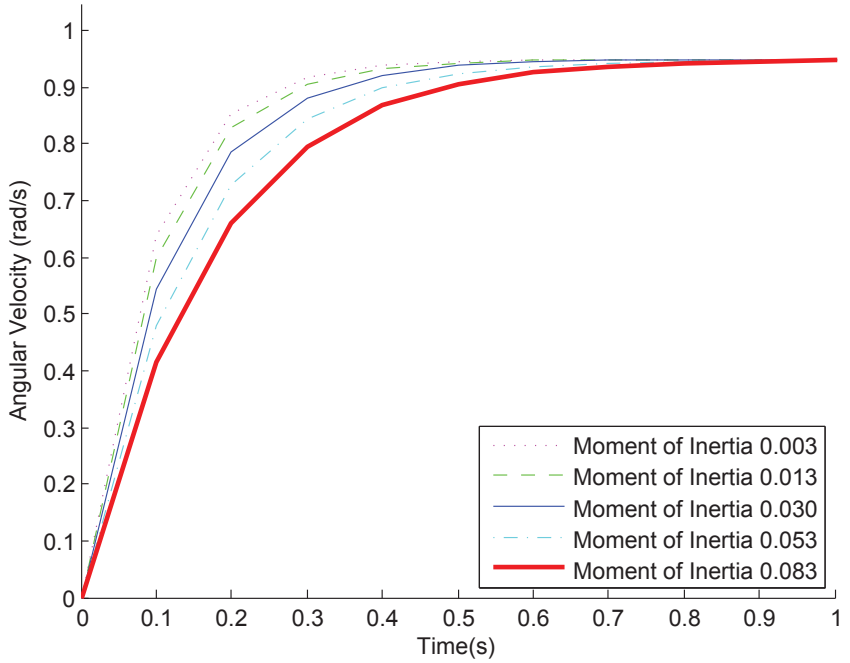


FIGURE 2.25: Transient response of the angular velocity at various moments of inertia

2.4.5 Effect of Support Distribution Variation

This section considers the effect of varying the support contact point locations while keeping all other variables constant. In this example, the support points are arranged in a square pattern centered about the center of mass which is a likely arrangement for legged objects such as a coffee table where contact with the support surface is made at a number of small, well defined regions. The side length of this support pattern is varied from $0.1m$ to $0.5m$ over the course of the simulations. The object's side length, mass and the pushing contact location are maintained at their default values, listed in Table 2.2. Figure 2.26 shows the system response as the side length of the support pattern is varied. Both the quasi-static and dynamic responses exhibit a larger angular velocity for a given pushing velocity as the side length of the support pattern is reduced. It is interesting to note that all of the dynamic responses converge to the same point when pushed at a very high speed. This effect occurs because the dynamic properties of the object are constant for all cases simulated and the static effects become negligible at very high speeds. A final observation is that quasi-static approximation becomes increasingly divergent from the dynamic response as the pushing vector moves away from the support distribution, particularly when the pushing vector does not pass through the convex hull of the support points as a very large angular velocity is required to balance the system resulting in significant inertial effects. As a result, it appears that even at low speeds the quasi-static approximation should not be applied when the pushing vector does not pass through the convex hull of the support points.

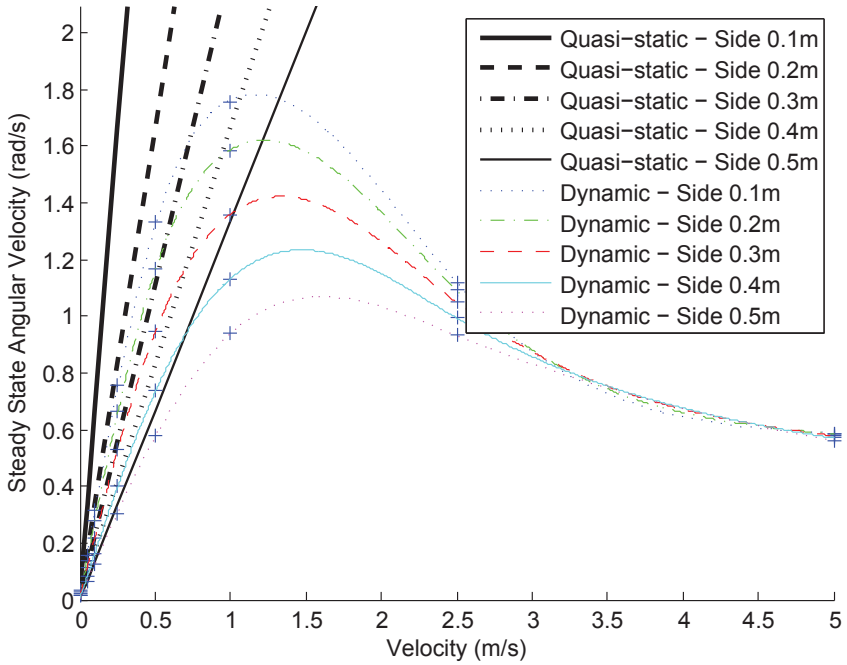


FIGURE 2.26: Comparison of the angular velocity with pushing velocity at various support patterns

2.4.6 Summary

This case study investigates the behavior of an object being pushed at a fixed point on a flat uniform surface at constant velocity. It is shown that when an object is pushed with constant velocity in a direction that does not pass through the center of friction, the object will eventually reach a steady angular velocity and travel in a circular trajectory with the length of the transient period influenced by the dynamic effects of the system. The case study also reveals that the object mass and moment of inertia have no bearing on the steady state angular velocity, although the moment of inertia will influence the length of the transient period. The support friction coefficient plays a role in determining the object behavior and will cause the steady state angular velocity to increase as the support friction increases. The agreement between the quasi-static and dynamic responses is also enhanced as the support friction is increased. The location of the support points also influences the object behavior with widely spaced support patterns giving better agreement between the quasi-static and dynamic responses than closely spaced arrangements.

2.5 Case Study 2: Stable Pushing with a Curved Fence

This section presents the results of a series of computer simulations, deploying a full dynamic model, carried out to explore the existence of stable pushing regions for pushing with a curved fence as discussed in Section 2.3. A cuboid object sliding on a uniform flat surface while being pushed at constant velocity by a cylindrical robot as shown in Figure 2.27 with the default parameters given in Table 2.2 is used in the study. The radius of the pushing robot varies from $125mm$ to $275mm$. As the robot is pushing on one of the flat object sides the contact point will be free to move as the relative rotation between the robot and object changes.

To test the system stability, the robot is simulated to push in a straight line for a distance sufficient that either a stable steady state is reached or control of the object is lost. The stability of the system is tested at regular intervals around the perimeter of the object to determine the location and extent of the regions which exhibit stable pushing behavior. The simulation is repeated for various robot radii and friction center locations.

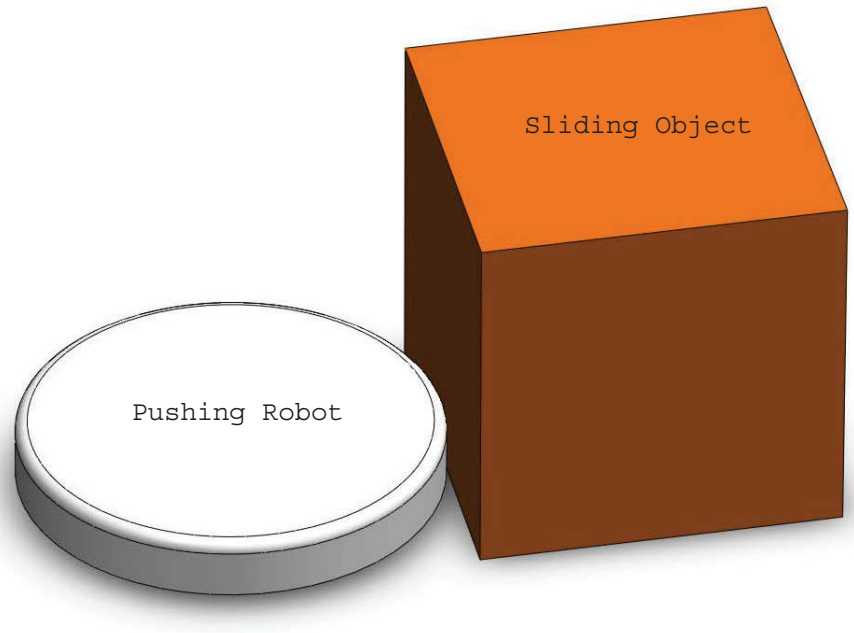
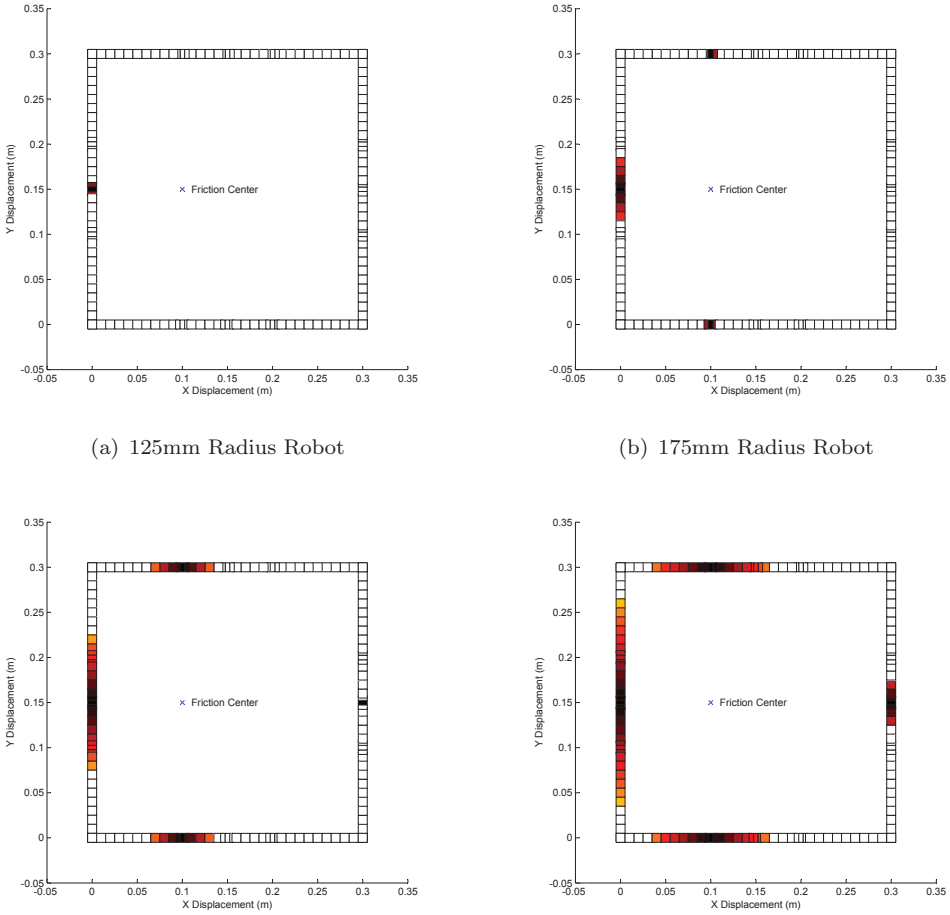


FIGURE 2.27: Round robot pushing a cuboid object on a side resulting in curved surface interaction

Effect of Variation in Robot Radius

The variation of the stable regions, as the robot radius is increased, is shown in Figure 2.28. The sampled pushing locations are represented by the small squares arranged around the object perimeter, colored if a stable equilibrium is achieved or white if a stable equilibrium is not achieved. Pushing locations with a greater tolerance to disturbances in object positions are colored with a darker shade.

As predicted, the stable regions expand as the robot radius increases. Hence, for a $125mm$ robot only one very narrow stable region exists on the side closest to the friction center while for a $275mm$ robot stable regions exist on all four sides of the object and the majority of the side closest to the friction center encompassed by a stable region. Pushing locations which are distant to the edge of the stable region can accommodate a larger disturbance without losing control of the object than pushing locations which are close to, or at the boundary of a stable region.



(a) 125mm Radius Robot

(b) 175mm Radius Robot

(c) 225mm Radius Robot

(d) 275mm Radius Robot

FIGURE 2.28: The stable pushing regions around the perimeter of the object increase as the diameter of the pushing robot increases. Darker squares indicate pushing locations with a greater disturbance tolerance.

Effect of Variation of the Friction Center

The variation of the stable regions as the friction center is moved is shown in Figure 2.29. In this simulation the radius of the pushing robot is kept constant at $175mm$ and the mass distribution of the object is altered to move the friction center. The friction is located towards the upper left, center left, lower left and true center of the object. It can be seen that each stable region is centered about the point where the surface normal passes through the friction center. The width of a particular stable region depends on the perpendicular distance from the contact point to the center of friction with the stable region growing as the distance between the contact point and the friction center reduces.

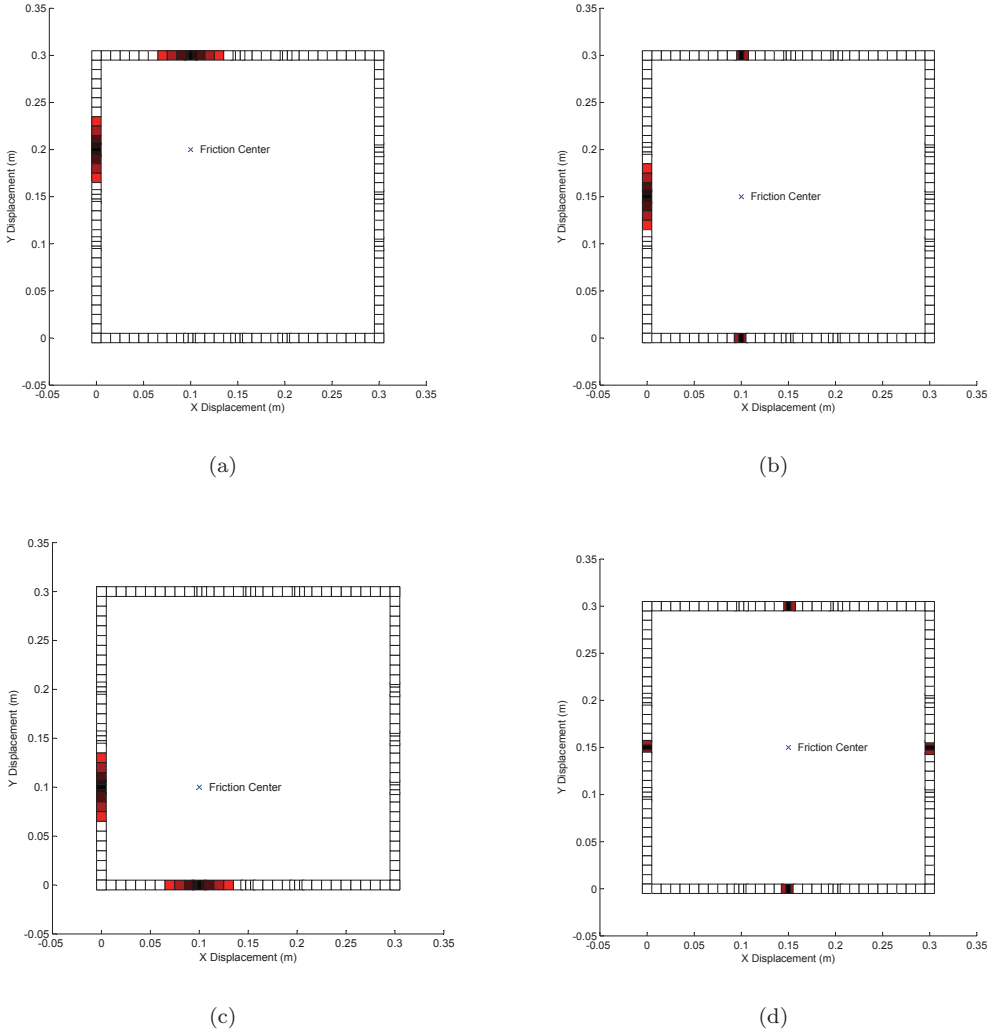


FIGURE 2.29: The stable regions with the friction center in various locations. Darker squares indicate pushing locations with a greater disturbance tolerance.

2.6 Case Study 3: Pushing a Box with a Mobile Robot

This section presents the results of a series of real world experiments carried out to explore the existence of stable pushing regions for pushing with a curved fence as discussed in Section 2.3. The pushed object, shown in Figure 2.30, is a cardboard box fitted with hemispherical nylon feet positioned in a 300mm square pattern and weighed with 2kg of movable weights to allow the friction center to be altered. The object slides on a flat melamine table top and is pushed by a $330mm$ diameter Turtlebot mobile robot equipped with a hokuyo laser range finder, as shown in Figure 2.31. The object contacts with the robot at a single point, which will move as the relative rotation between the robot and object changes, and is constrained to roll without sliding around the perimeter of the robot. The orientation of the object is determined from the laser data using the segmentation algorithm proposed by Kodagoda et.al. [23].

(a) (b)

FIGURE 2.30: Sliding object showing (a) an example of mass distribution for balanced support friction and (b) the location of the support points.

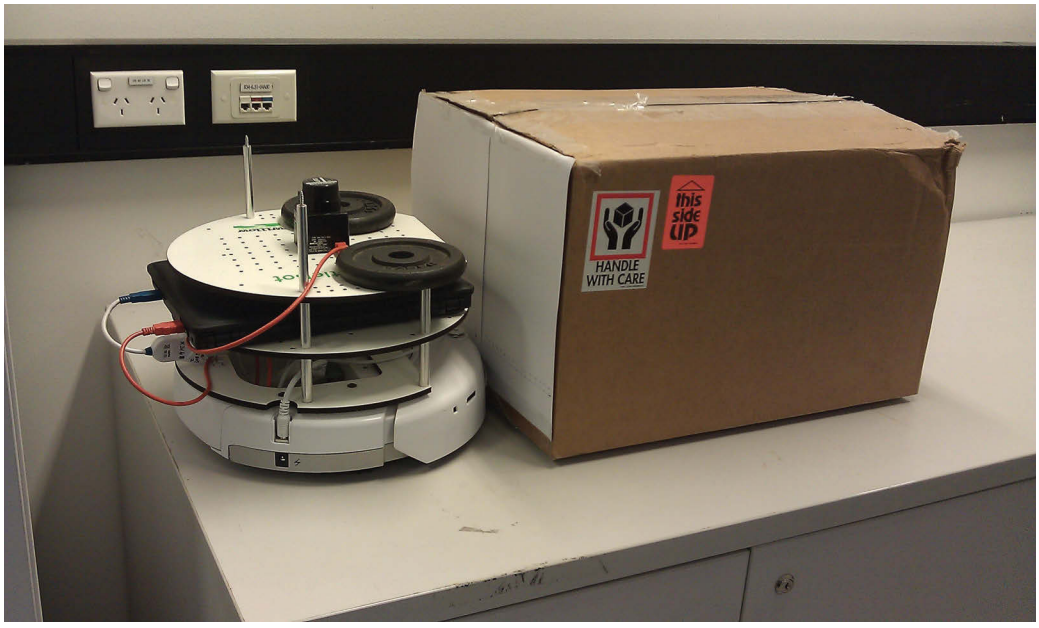


FIGURE 2.31: Robot and object positioned to perform a pushing operation

To test the system stability, the orientation of the object is tracked relative to the robot while the robot is driven forward in a straight line for two meters at $0.1m/s$. At this pushing speed the system will behave in a quasi-static manner. The experiment is commenced with the object either at the neutral position directly in front of the robot or disturbed from the neutral position by either $0.15rad$ or $0.3rad$. Where the test requires an initial disturbance from the steady state position, the object is rolled without sliding from the steady state position to the disturbed position prior to the commencement of the test. The experiment is repeated for various friction center locations, achieved by altering the distribution of weights inside the box.

Effect of Variation of the Friction Center

Figure 2.32 shows the orientation of a sliding object as it is pushed by a mobile robot for various friction center locations. In each case, an orientation of zero indicates that the object is positioned directly in front of the pushing robot which is the steady state position for the object if a stable pushing region exists.

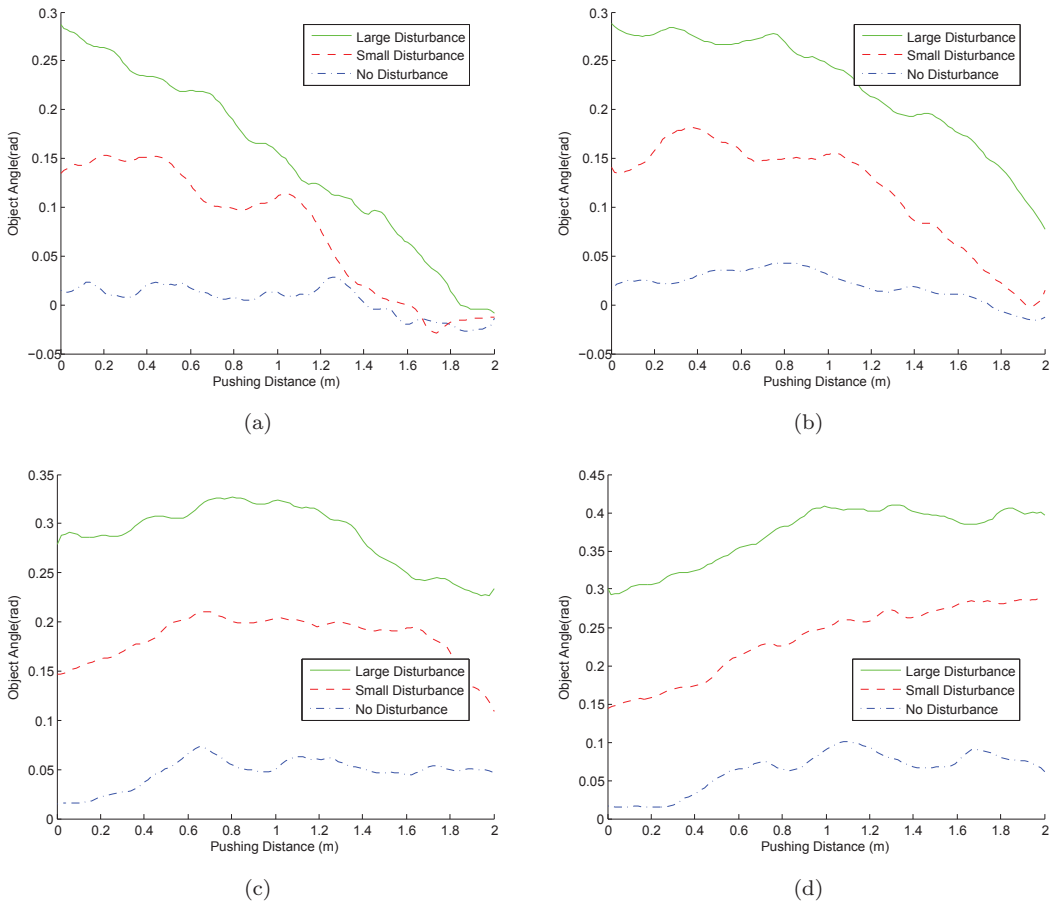


FIGURE 2.32: The rotation of a sliding object pushed from both disturbed and undisturbed initial positions for various friction center locations (a) 50mm (b)100mm (c) 150mm (d) 200mm from the pushing contact point.

In Section 2.3 it was shown that a stable pushing region will exist for a round robot pushing a flat faced object if the perpendicular distance from the pushing contact point to the friction center is less than the robot radius. As the radius of the pushing robot in this experiment is $165mm$, a stable pushing region should exist for the configurations which have a perpendicular distance to the friction center of $50mm$, $100mm$ or $150mm$.

When the friction center is located at $50mm$, as shown in 2.32(a), the object recovers from both small and large disturbances and settles at the neutral position. A similar behavior is demonstrated when the friction center is located at $100mm$, as shown in 2.32(b), although the recovery from disturbance is slower and the system does not have time to reach the neutral position from a large initial disturbance. At $150mm$, the distance to the friction center is quite close to the $165mm$ limit for stable manipulation to occur. As such, fluctuations in the position of the friction center may cause the stable pushing region to disappear for part of the pushing action. Never the less, Figure 2.32(c) shows the object orientation trending towards the neutral position when starting from a disturbed position although the orientation deviates slightly when starting from the neutral position. It is unclear whether the object will be stably pushed by the robot during an extended pushing action.

When the friction center is located at $200mm$, as shown in 2.32(d), no stable pushing region exists. As expected, the orientation of the object tends to diverge from the neutral position under all initial conditions.

2.7 Conclusions

This chapter investigated the behavior of a sliding object when it is pushed on a flat horizontal surface at a single point on the object perimeter. Operation under both quasi-static and dynamic conditions is considered, and revealed that an object will behave like a non-holonomic wheeled vehicle in both cases after a short transient period as the angular velocity is directly proportional to the pushing angle much like the turning rate of a car is directly proportional to the steering angle. For quasi-static conditions this transient period can be safely ignored, as postulated by Kurisu and Yoshikawa [24]. Although traditional non-holonomic path-planning techniques such as those based on Dubins curves [16] cannot be directly applied to systems with non-trivial dynamic effects, they can be used as an excellent starting point for optimization. The investigation also explored how variation of the key system parameters influenced the motion of the sliding object. The support friction distribution was shown to have a significant influence on object motion while other parameters, such as the object mass or moment of inertia, have little or no effect. The existence of stable pushing regions were demonstrated where an object is pushed using a curved fence. The equations governing this behavior were derived and a series of numerical and physical experiments designed to explore the extent of the stable regions under a variety of system configurations were undertaken. It was demonstrated that the location and size of a stable pushing region is predominantly dependent on the geometry of the pushing surfaces and the location of the center of friction.

If attempting stable manipulation, the pushing contact location and initial pushing direction should be chosen to minimize the distance to the friction center. This will assist in maximizing the possibility of a stable push. As it is quite difficult to estimate the support friction in practical scenarios, the friction center may be artificially moved towards a desired pushing contact point by increasing the support pressure near the pushing point. This could be achieved, for example, by placing a weight on the object or pressing downwards using a manipulator.

Subsequent chapters build on the object behavior to develop pushing trajectories which deliver the object to a particular desired position and orientation in the plane. Control strategies are developed to assist in achieving successful manipulation in the presence of uncertainty in the environmental conditions.

Chapter 3

Open Loop Planning for Pushing Actions

For a robotic system to successfully manipulate objects to achieve a desired outcome it must first possess an understanding of how an object in the environment will behave if and when it is acted upon by the robot. This knowledge, if available, enables the robotic system to reason about which actions will result in a desirable object response and thus solve the inverse problem, "what actions must be taken for the object to end up in a particular position?" This chapter builds on the understanding of sliding object motion studied in Chapter 2 and answers that question, combining the dynamic model with a robot path planner to produce near time optimal trajectories to manipulate the object to a desired pose.

If it is assumed that the robot and object are constrained to move on a flat support surface then the robot can directly influence the motion of the contact point with the object in the two perpendicular directions formed by the support surface. The object can, however, also rotate about the contact point and this motion cannot be directly controlled by the robot. Instead, it is determined by the external forces between the object and environment, hence, the system is under-actuated. As seen in the discussion provided in the previous chapter, when the pushing vector passes through the center of friction the object does not rotate allowing the position of an arbitrary point on the object to be directly controlled for motion in one direction. However, if the pushing vector does not pass through the center of friction then a combination of translation and rotation will occur with the exact proportions dictated by the external forces. Hence, it is not possible to directly control the rotation of the object or the translation of the object in any direction other than the pushing direction.

The methods of dealing with this under-actuated system can be broadly divided into two categories. Firstly, the geometry of the system can be defined such that the object becomes constrained with respect to the pushing robot. Traditionally this has been

achieved by pushing with either a flat fence (against a flat sided object) or by pushing with a number of contact points creating a virtual fence between them. The strength of this methodology is that suitable margins of error may be included so that the effect of support uncertainty is not significant in the determination of the object motion. This allows open loop plans which are guaranteed to produce a successful result to be generated. In addition to the classical fence based approaches, it has been demonstrated in Chapter 2 that certain system configurations will yield a similar stable response even though the contact between the pushing robot and sliding object is made only at a single point. The second category are solutions which use specific robot motions, coupled with an understanding of the effect of various environmental forces to act as an additional actuator. While this method allows simpler and more general contacts between the robot and object the solutions rely on an accurate knowledge of the system characterization.

This chapter begins with a review of the literature related to the planning of manipulator actions to move a sliding object to a goal location. Following this, the path planning problem is posed as a constrained optimization problem and solved for a variety of system configurations. Finally, the sensitivity of the produced trajectories to variations in the system parameters is investigated to determine which parameters are most likely to cause errors in the object motion.

3.1 Literature Review

Early work generating manipulation trajectories for pushed sliding objects focused heavily on applications where the geometry and system mechanics cause the object to align with and become attached to the pushing robot [30, 31, 36]. Often this is achieved by pushing a polygonal object using a flat fence, however, pushing an object at several points can create a virtual fence and achieve the same result. Regardless of how the "stable push" is achieved, the fact that the object maintains a constant position and orientation relative to the robot greatly simplifies the task of generating motion plans. As long as the pushing constraints are honored, a successful manipulation trajectory can be produced by stringing together stable pushes until the robot, and hence the object being manipulated, arrive at the desired location.

Significant research into manipulation by pushing was undertaken in the 90's and early 00's resulting in a large body of work. Remaining issues, such as those addressed in this thesis have proven to be challenging and the work in this area came to somewhat of a standstill. It is only recently that researchers are re-visiting the problems proposed ten years ago. Much of the contemporary work is focused on managing the uncertain nature of the support friction to successfully achieve object manipulation. Nozawa et.al. [44] estimate the support friction force of a heavy sliding object by measuring the pushing forces applied by a two armed humanoid robot. While this work is not directly related

to this thesis, it is noted here as a significant challenge which needs to be addressed for practical implementations.

Manipulation of objects using a single point of contact has also been investigated by several researchers. Walker and Salisbury [51] store a response map indicating how a particular object will respond to perpendicular linear push at locations distributed around the perimeter of the object. The planning algorithm chooses the perimeter location at which a fixed length linear push moves the object closest to the desired goal location. As there is no need to explicitly store a model of the support friction under the object this system is relatively easy to implement. Unfortunately, however, the manipulation map used for a particular object cannot be generalized for other objects, not even larger versions of the same object, so a new map must be generated for each object that must be manipulated. While the current implementation only uses a planning horizon of a single push, the authors indicate that it is possible to adapt the system to look ahead for a planning horizon of multiple pushes to improve system performance.

Agarwal et.al. investigate the task of manipulating uniform unit disks in the presence of polygonal static obstacles [1]. The pushing contact is made at a single point which is assumed to have perfect roughness [35] ensuring that the contact does not slip along the perimeter of the disk. The manipulation trajectory is composed of a sequence of linear pushes where the pushing location, direction and duration are chosen by the planning algorithm. The motion for each step is predicted by treating the robot and object as an articulated two body non-holonomic robot and applying a variation of the tree based search presented in [5] to determine a trajectory with the minimum number of discrete pushing actions. This method is limited to cases where the dynamic forces are negligible, hence, it is only applicable when quasi-static conditions may be assumed.

Kurisu and Yoshikawa [24] approach the trajectory generation problem by considering the kinematics of a particular pushing system and implementing an optimizer based search to determine an appropriate control action. They assume that the manipulation takes place under quasi-static conditions, thus allowing the inertial effects to be discounted and the system treated as a purely kinematic system. A contact point on the perimeter of the object is selected and maintained unchanged throughout the entire manipulation action. A trajectory specifying the relative pushing angle between the robot and the object at this contact point over the course of the manipulation action is used to specify the actions necessary to control the object behavior. Work presented in this chapter uses the full dynamic equations for the system in place of a kinematic approximation. It builds on optimal planning techniques that have been successfully applied to a number of problems outside the task of manipulation by pushing [12, 27, 28] to generate near minimum time open-loop trajectories for positioning an object using pushing actions.

3.2 General equations

Consider the motion of a dynamic system consisting of a sliding workpiece pushed by a mobile robot as described in Chapter 2 and shown in Figure 3.1. The system consists of a sliding object, pushed at a single contact point by a mobile robot. As many robotic systems are equipped with on-board velocity controllers, the control action is selected to consist of a constant forward velocity coupled with a variable pushing angle.

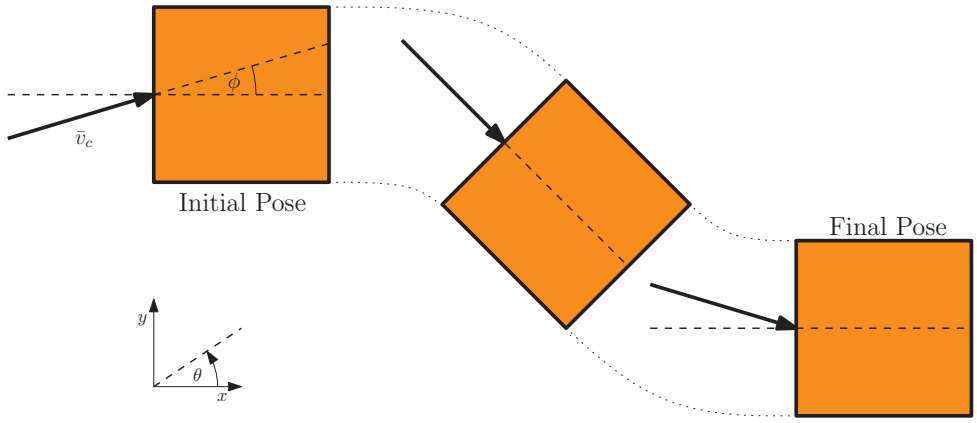


FIGURE 3.1: Movement and pushing input from an initial to final object position

The system state may be described as

$$\begin{pmatrix} s_{(t)} \\ \dot{s}_{(t)} \end{pmatrix} \tag{3.1}$$

where

$$s_{(t)} = \begin{pmatrix} x_{(t)} \\ y_{(t)} \\ \theta_{(t)} \end{pmatrix} \tag{3.2}$$

$$\dot{s}_{(t)} = \begin{pmatrix} \dot{x}_{(t)} \\ \dot{y}_{(t)} \\ \dot{\theta}_{(t)} \end{pmatrix} \tag{3.3}$$

The trajectory planning task of moving the object from an initial location, at time zero, to a goal location, at time t_f, in minimum time can be stated as: determine the trajectory s such that

$$\underset{(t_f, \phi_{(t)(0 \leq t \leq t_f)})}{min} t_f \tag{3.4}$$

where t_f is the terminal time to be minimized.

The trajectory produced must satisfy the following terminal conditions

$$s_{(0)} = \begin{pmatrix} x_{(0)} \\ y_{(0)} \\ \theta_{(0)} \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ \theta_0 \end{pmatrix} \tag{3.5}$$

$$s_{(t_f)} = \begin{pmatrix} x_{(t_f)} \\ y_{(t_f)} \\ \theta_{(t_f)} \end{pmatrix} = \begin{pmatrix} x_f \\ y_f \\ \theta_f \end{pmatrix} \tag{3.6}$$

To ensure that a continuous contact between the object and the pusher is maintained, the following inequality constraints must be satisfied at all times

$$R_n \geq 0 \tag{3.7}$$

$$\|R_t\| \leq \|\mu_c R_n\| \tag{3.8}$$

where R_n is the normal pushing reaction force, R_t is the tangential pushing reaction force and μ_c is the coefficient of friction between the robot and the object.

The free terminal time optimal control problem as stated, is not tractable, except for a number of relatively simple scenarios. This problem can, however, be approximated using an appropriate parametrization to represent the control input $\phi(t)$. Dissanayake et. al. [12] used a piecewise constant approximation in the form shown in Figure 3.2 to describe the control input as a set of unknown constants, to obtain near time optimal motion plans for robotic manipulators. This strategy makes it possible to transform the optimal control problem to a constrained optimization problem whose dimension is related to the number of partitions used in the piece-wise constant approximation. An approximate solution to the original free terminal time optimal control problem can now be obtained using conventional Sequential Quadratic Programming [8] techniques such as the Nelder-Mead Simplex Algorithm [26].

For a given set of input angles ϕ and the terminal time, the final pose of the object can be computed. Maintaining continuous contact as defined in Equations 3.7 and 3.8 require enforcing continuous state constraints. While this type of constraint can

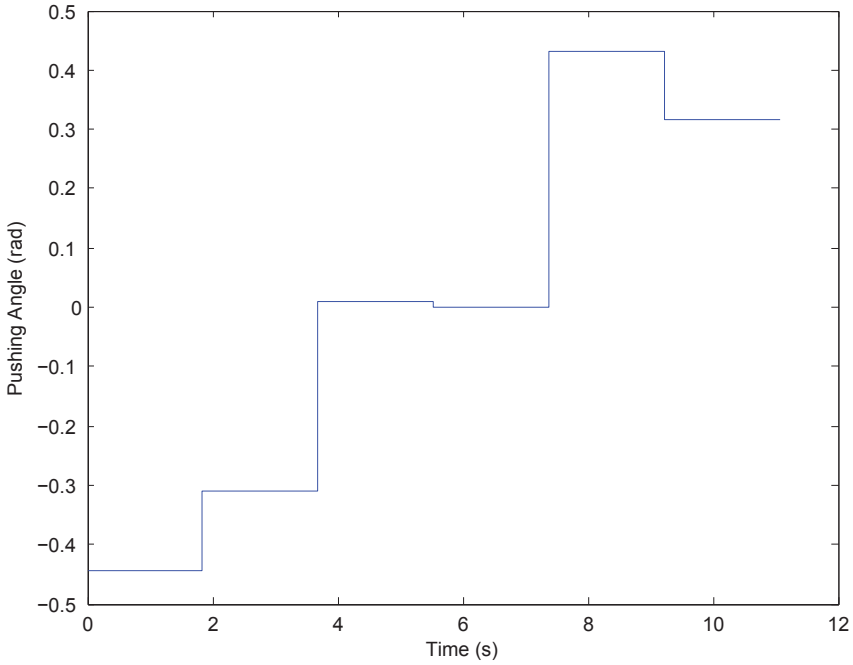


FIGURE 3.2: Piecewise constant approximation of the pushing angle

be handled, as demonstrated in [12], it is computationally expensive. As the robot is constantly advancing at a fixed speed, the simplifying assumption that contact is always maintained is made. In order to enforce the constraint that the tangential reaction force remain within acceptable bounds, the control input ϕ is constrained to be within appropriate bounds.

$$-\arctan\mu_c \leq \phi \leq \arctan\mu_c \tag{3.9}$$

where μ_c is the coefficient of friction at the pushing contact point and $\arctan\mu_c$ gives the maximum pushing angle before sliding occurs.

Enforcing the constraint ensuring that the input variables are within bounds during an optimization is relatively straightforward.

The optimization problem is formulated using the set of ϕ, that approximates the control input, and the terminal time as the variables to be optimized. The terminal condition, in response to the input variables, was obtained using a dynamic simulation generated with the Simulink SimMechanics toolbox instead of a traditional mathematical model of the system. Terminal state constraints and the bounds on the control input were implemented with a penalty function, making it possible to use the Nelder-Mead Simplex Algorithm for solving the optimization problem. The implementation of this algorithm in the Matlab function fminsearch was used to generate the results shown in the following sections.

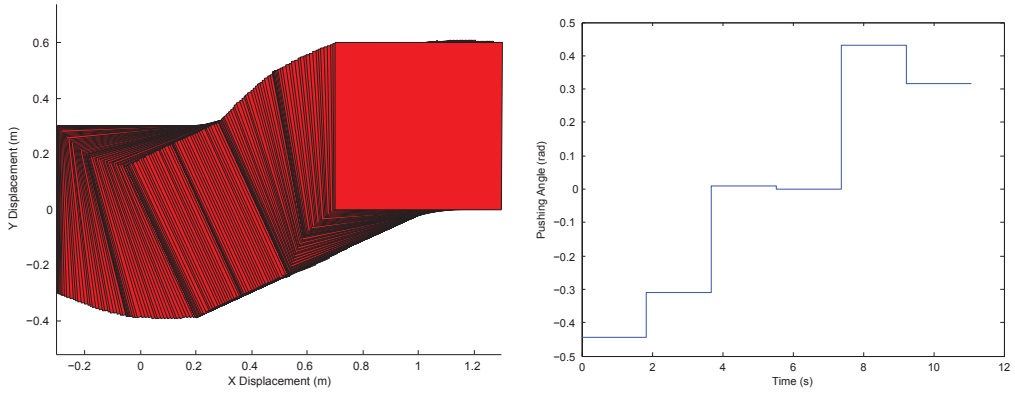
3.3 Selected Numerical Case Studies

The optimization problem is solved for a number of different system configurations, namely, an object pushed by a point robot and an object pushed by a cylindrical robot of both relatively small radius and relatively large radius. In all cases the object is of cuboid shape with uniformly distributed mass and physical properties as stated in Table 2.2. The friction distribution is chosen to be uniform and concentrated at the four corners of the support surface.

3.3.1 Trajectory Generation for Manipulation using a Point Contact

The first configuration investigated is a system where the robot pushes the object at a single point in the center of one of the object faces with a constant forward velocity. The initial pose is set to the origin [0 0 0] while the desired final pose is set to [1 0.3 0]. The pushing velocity is maintained at $0.1m/s$ and the pushing angle limited to $\pm 0.45rad$. The number of partitions in the piecewise constant approximation is selected to be six. Other system parameters are as specified in Table 2.2.

Figure 3.3(a) shows the path followed and Figure 3.3(b) the associated input vector generated by the optimizer. The optimization was started from multiple initial guesses for the input vector to confirm the result to be a true minimum solution.



(a) (b)

FIGURE 3.3: Manipulation by a point robot (a) Object Trajectory (b) Input

Impact of Number of Partitions

As the number of piecewise linear control input partitions approaches infinity the resulting trajectory will always approach the true minimum time solution as the input partition edges align with the switching points of the true minimum time solution. The computation burden also increases as the number of partitions increases, hence, in practical implementations a compromise must be reached where both the accuracy and

computation time are acceptable. Figure 3.4 compares the input vectors and trajectories generated using different numbers of partitions. It can be seen that the form of the input vectors are substantially similar, each exhibiting a period of significant negative pushing angle followed by a period of near zero pushing angle and finally a period of significant positive pushing angle. However, trajectories with more partitions are able to better account for the dynamic nature of the system and arrangements where the optimal switching points are irregularly spaced in time. Table 3.1 lists the computation time for various numbers of partitions, normalized about the computation time for six partitions, for a symmetrical trajectory with a goal at [1 0.3 0] and an asymmetrical trajectory with a goal at [1 0.1 -0.75] as illustrated in Figure 3.5. As expected, a steady increase in the computation time is observed as the number of partitions is increased. In the asymmetrical case the final orientation of the goal location is different to the initial orientation of the object, thus the pushing trajectory will require a greater rotational effort in one direction than the other. As a result, the pushing trajectory will become asymmetrical with the period of positive pushing angle being dominant in this example. Again, the computation time exhibits a steady increase as the number of partitions is increased. However, unlike the previous example where the symmetrical nature of the trajectory allowed the successful generation of trajectories with a number of partitions down to two, for the current goal location, the unsymmetrical nature of the trajectory requires a minimum of four partitions before a successful trajectory may be generated. Furthermore, in the four partition case the partition boundaries and switching points are particularly poorly aligned requiring a significant amount of processing time to obtain a solution. For the remaining case studies presented in this chapter, the number of partitions is selected to be six. This provides sufficient accuracy for both symmetrical and asymmetrical trajectories and a reasonable computation time.

Number of Segments	Symmetrical Trajectory Computation Time (%)	Asymmetrical Trajectory Computation Time (%)
2	31	-
3	43	-
4	76	148
5	81	69
6	100	100
7	139	240
12	444	503
18	611	689

TABLE 3.1: Computation times for typical symmetrical and asymmetrical trajectories with an increasing number of segments. Computation time for six segments used as a benchmark

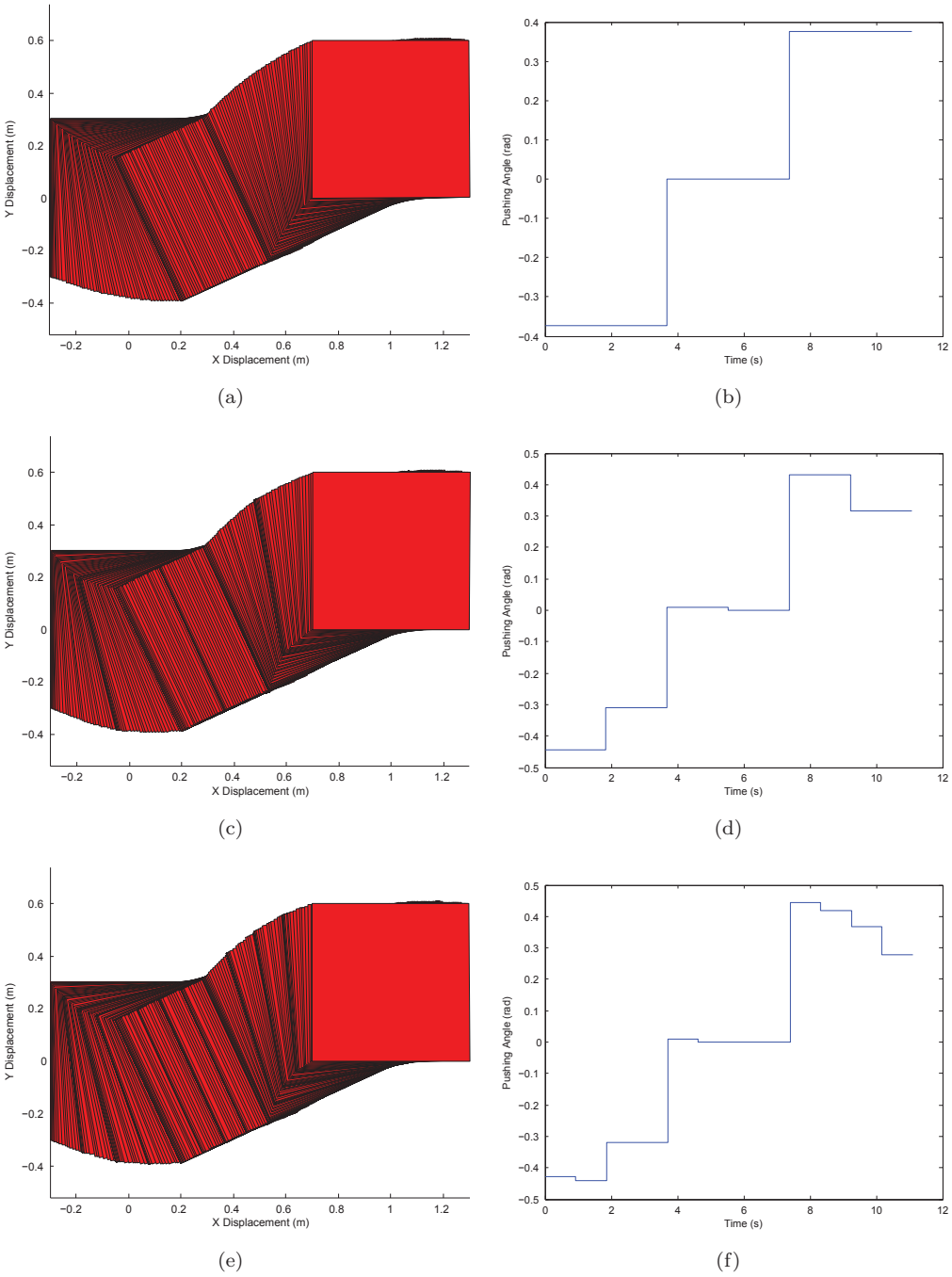
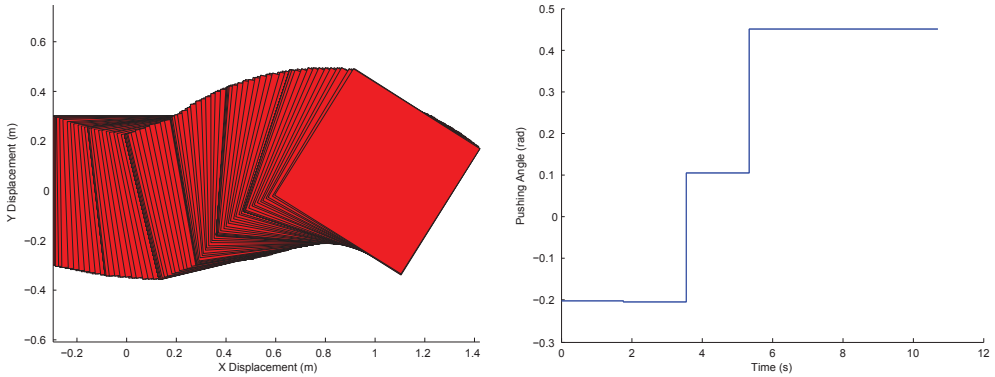


FIGURE 3.4: Trajectories and inputs for selected numbers of segments.



(a) (b)

FIGURE 3.5: Manipulation by a point robot to [1 0.1 -0.75] (a) Object Trajectory (b) Input

Effect of Increased Pushing Velocity

If the object is pushed at different velocities, the variation in the dynamic forces of the system will influence the pushing trajectory generated by the optimizer. Figure 3.6 illustrates this variation for pushing velocities ranging from $0.05m/s$ to $0.5m/s$. In this example, the result generated for a pushing velocity of $0.05m/s$ is near identical, in both object motion and input vector, to the response for a pushing velocity of $0.1m/s$ shown in Figure 3.3. As the pushing velocity is increased to $0.25m/s$ there is a small change in the input trajectory with the pushing angle increasing for several segments. When the pushing velocity is increased to $0.5m/s$ the pushing angle is increased dramatically with the period of low pushing angle in the center of the trajectory and the resulting pure translation being eliminated. This result demonstrates that as dynamic forces become increasingly significant, more dramatic input actions are required to achieve a comparable response. It also suggests that some manipulation actions will be possible using slow manipulation but impossible using rapid manipulation.

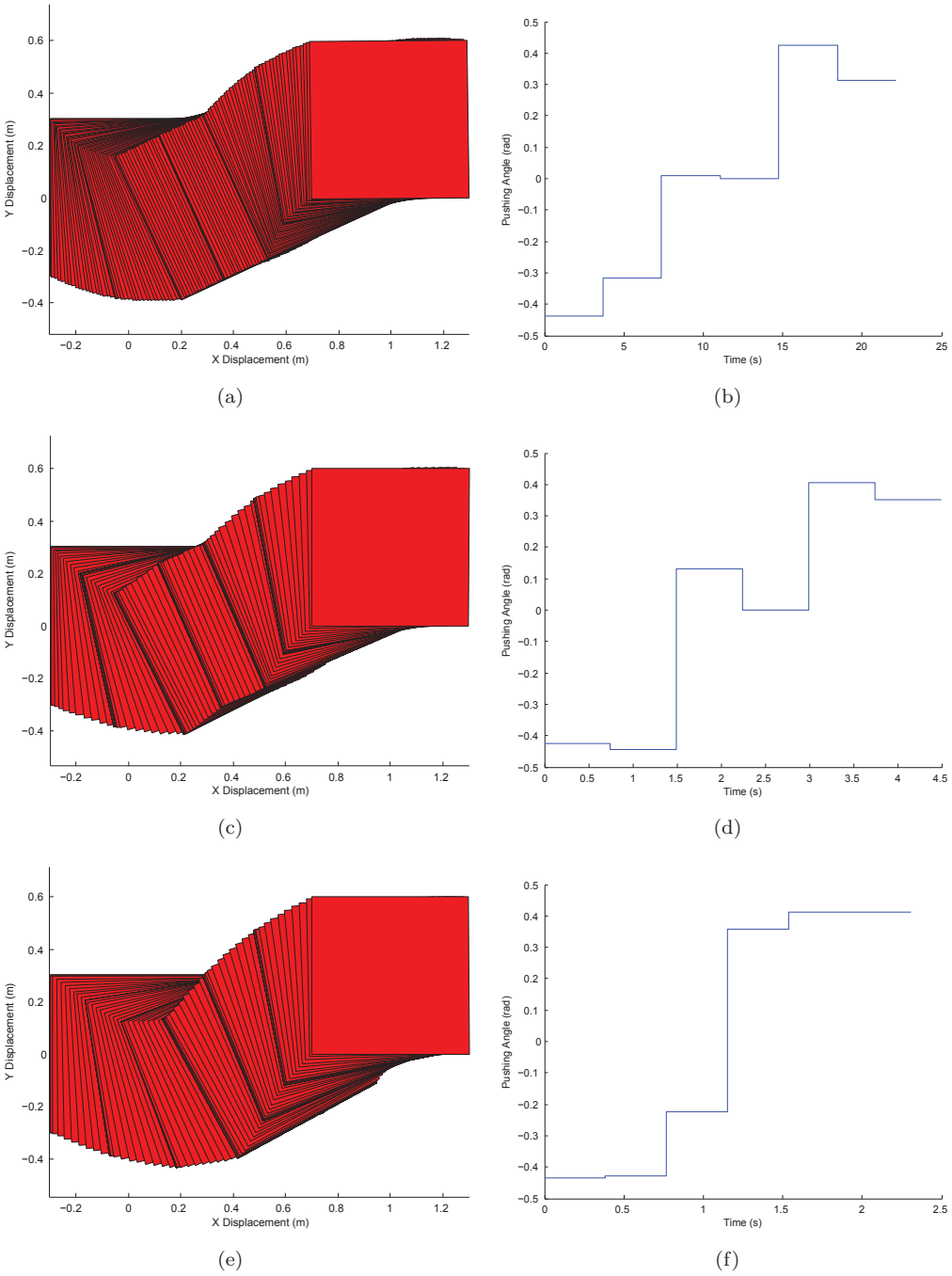


FIGURE 3.6: Trajectories and inputs for input velocities of 0.05, 0.25 and 0.5 m/s

Effect of Contact Coefficient of Friction

If a system is pushed at a single point, the maximum angular velocity which an object may achieve, for a given pushing velocity, is defined by the maximum allowable pushing angle. If a non-sliding contact is desired, the pushing force must fall within the friction cone, as shown in Figure 3.7. A pushing force acting outside the friction cone will result in the sliding of the contact point along the object's surface and a potential loss of control over the object motion. Hence, the maximum allowable pushing angle is defined by the pushing contact coefficient of friction. Figure 3.8 shows the resulting trajectory when a perfectly rough object is manipulated, such as a situation where the contact is a hinge or an enclosing grip around a table leg. While the maximum pushing angle increases from approximately $0.4rad$ to approximately $0.7rad$, it is interesting to note that the pushing angle does not increase to the maximum allowed pushing angle of $\frac{\pi}{2}rad$ as may be suggested by traditional Dubins [16] or Reeds - Shepp curves [49]. Instead the pushing angle reaches an intermediate level where the resulting translation and rotation of the sliding object maneuver it to the goal more rapidly than a trajectory which emphasizes either linear or rotational motions.

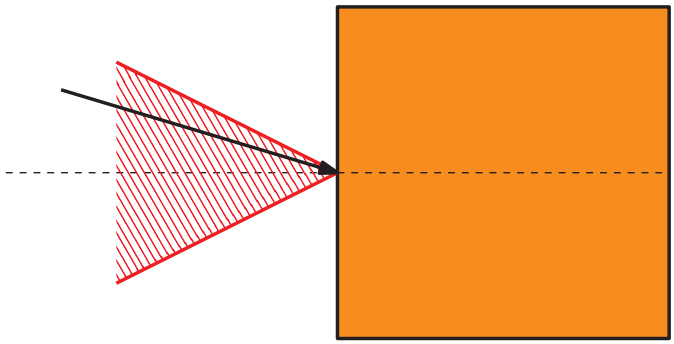
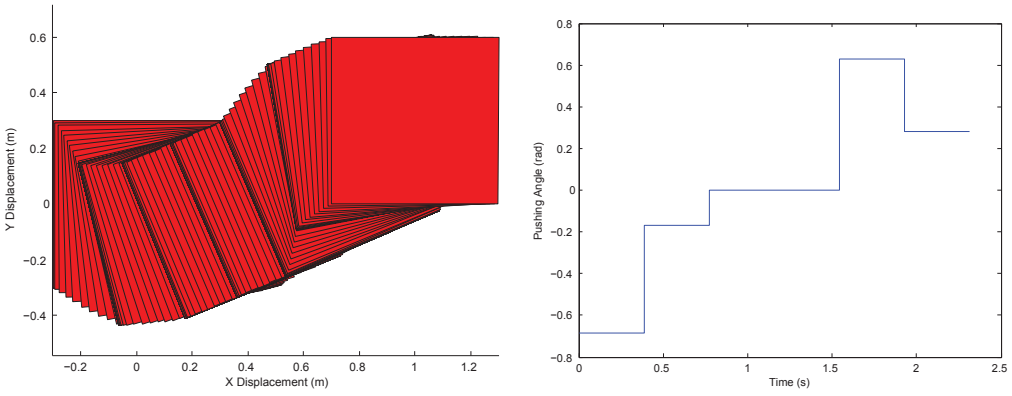


FIGURE 3.7: Pushing vector within the friction cone resulting in sticking contact

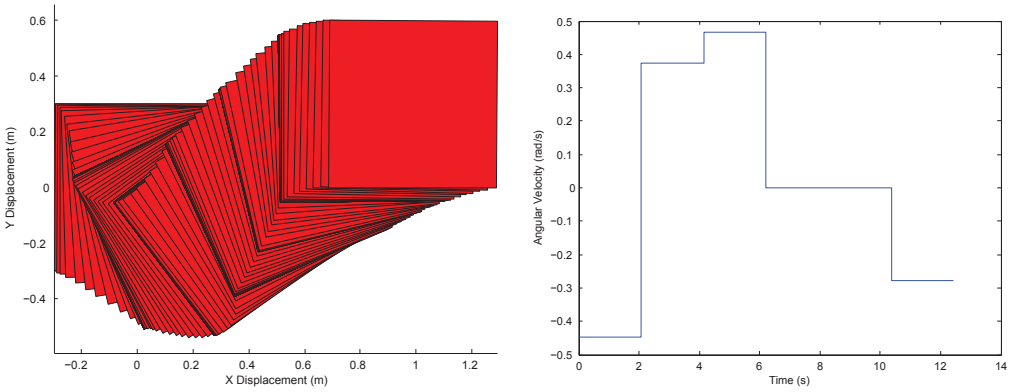


(a) (b)

FIGURE 3.8: Trajectories and inputs showing the effect of increasing velocity if there is no limitation on the pushing angle.

3.3.2 Trajectory Generation for Manipulation using a Curved Fence

The second configuration investigated is a system where the robot pushes an object with a curved fence forming a rolling contact with the object surface. As in Section 3.3.1, the initial pose is set to the origin while the desired final pose is set to [1 0.3 0]. The pushing velocity is maintained at $0.1m/s$ and the number of partitions in the piecewise constant approximation is selected to be six. Fig 3.9 shows the path taken and the input vector generated by the optimizer for the selected goal pose. When pushing with a single point of contact, a number of simplifications are afforded by adopting a strategy where the pushing is defined as a piecewise linear sequence of pushing angles. When pushing with a curved fence, the benefits gained in the point pushing arrangement are no longer applicable as the contact point will move as the pushing angle is varied. Instead, it becomes preferable to directly specify the linear velocities and angular velocities which the robot should traverse in order to manipulate the object to the desired goal. This approach has the potential to produce complex trajectories as, on occasion, it requires the robot to initially move away from the goal location so as to orient the object towards the goal for the remainder of the pushing sequence. For this simulation, the piece-wise constant approximation of the control input specifies the angular velocity of the robot instead of the pushing angle. The radius of the pushing robot, or the pushing implement, is $50mm$ and the other system parameters as specified in Table 2.2. The optimization was started from multiple initial guesses to confirm the result to be a true minimum solution.



(a) (b)

FIGURE 3.9: Manipulation by $50mm$ radius robot (a) Object Trajectory (b) Input

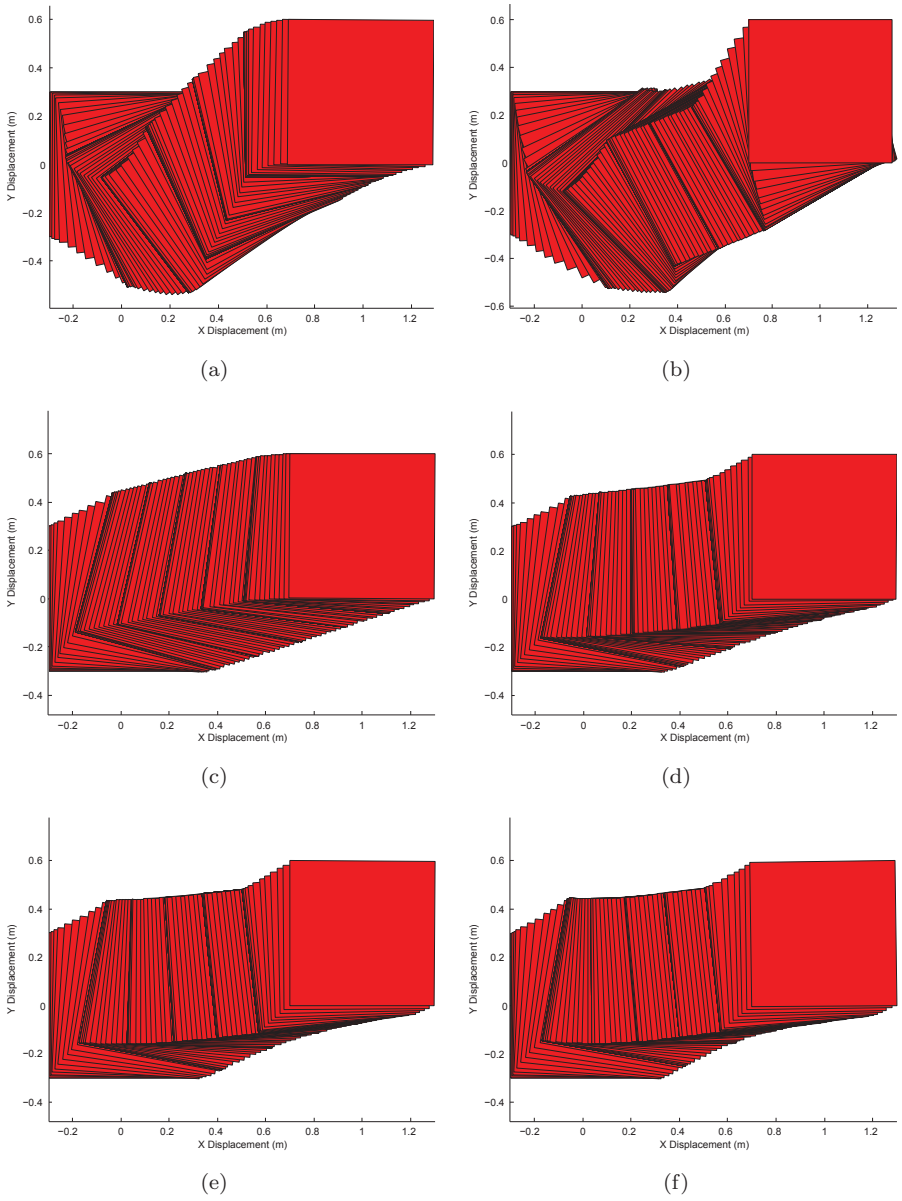


FIGURE 3.10: Trajectories for robot radii of (a) 50, (b) 100, (c) 200, (d) 300, (e) 400 and (f) 500mm

Figures 3.10 and 3.11 show the trajectories and input vectors for a variety of steadily increasing robot radii. It is interesting to note that once the robot radius exceeds 200mm the form of the shortest path and the associated input dramatically changes, with the robot no longer moving away from the goal at the beginning of the motion. Instead the robot begins by rotating and traversing towards the goal location, relying on the stabilizing effect of the curved fence to rotate the object towards the goal. While a robot diameter of 400mm or 500mm cannot be considered truly infinite as relative movement between the robot and the object is still evident, further increases in robot radius are likely to have little effect on the system behavior.

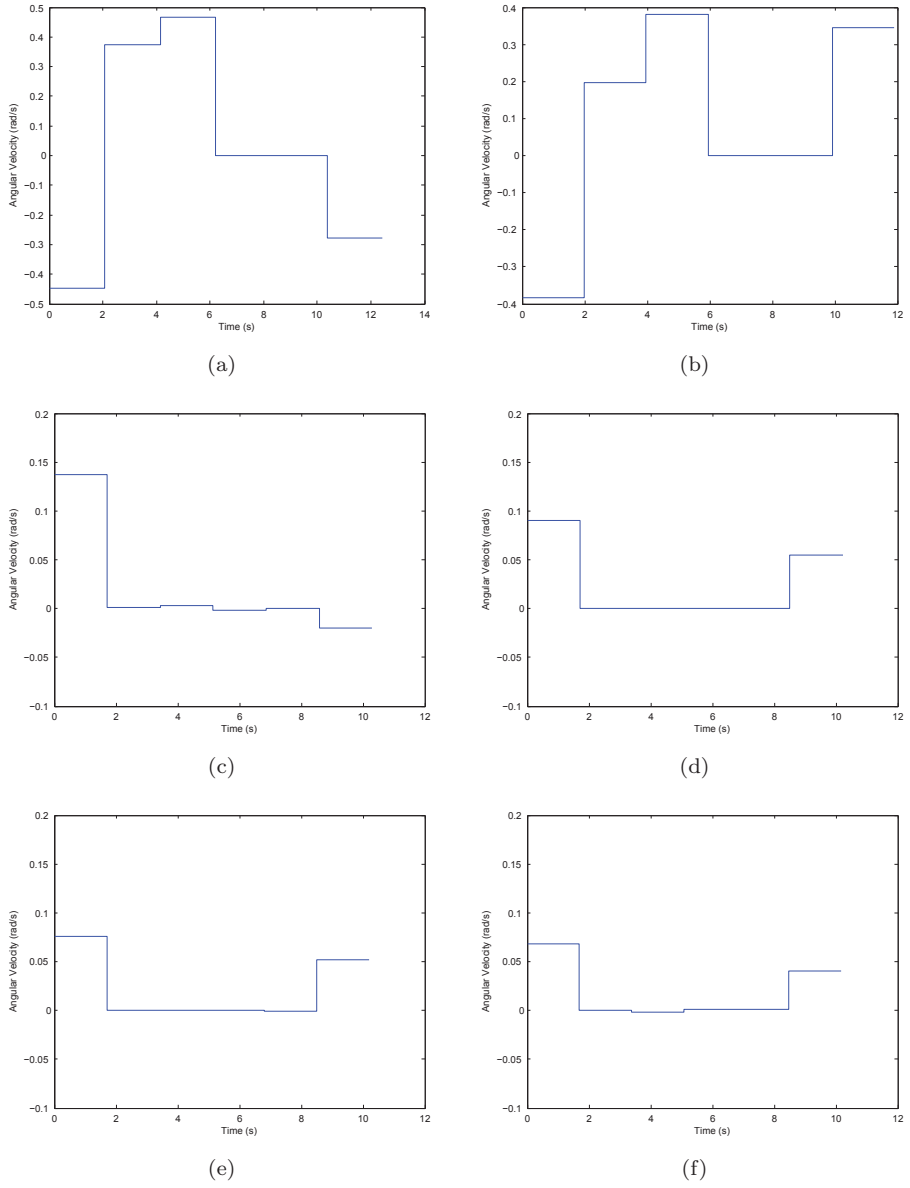


FIGURE 3.11: Inputs for robot radii of (a) 50, (b) 100, (c) 200, (d) 300, (e) 400 and (f) 500mm

3.3.3 Sensitivity Analysis

This section investigates the effect of variations in the pushing velocity, friction coefficient, object mass, object inertia, support point location, and friction center location on the accuracy of the final object location resulting from an application of open loop control actions presented is Section 3.2. Unless otherwise stated, the system parameters are as stated in Table 2.2.

Effect of Pushing Velocity Variation

To explore the effect of varying the pushing velocity, a trajectory is generated to manipulate an object, using a pushing speed of $0.1m/s$, to a goal at [1 0.3 0]. Following this, the input trajectory is executed with a variety of different pushing velocities ranging from $0.05m/s$ to $1m/s$ to observe the effect of a pushing velocity, different to that which was planned, on the final object position. Considering the results presented in Chapter 2, it can be hypothesized that for slow pushing velocities, where the system may be assumed to be operating under quasi-static conditions, variations in the pushing velocity will have little effect on the system response. Conversely, variations in rapid pushing velocities will cause a significant variation from the expected object trajectory due to the dynamic forces of the system. Table 3.2 demonstrates this effect, with significant variations in the final object position evident for pushing velocities of $0.5m/s$ or greater. This effect may also be seen in Figure 3.12. This behavior can be attributed to the influence of dynamic effects on the object angular velocity when pushed rapidly. As a result, the object will not rotate as much as expected during manipulation, resulting in a final position error.

Pushing Velocity (m/s)	Linear Error (m)	Rotational Error (rad)
0.05	-0.003	0.010
0.10	-0.010	0.012
0.25	-0.026	0.006
0.50	-0.042	0.030
0.75	-0.232	0.001
1.00	-0.225	0.015

TABLE 3.2: Final position error due to variation in pushing velocity

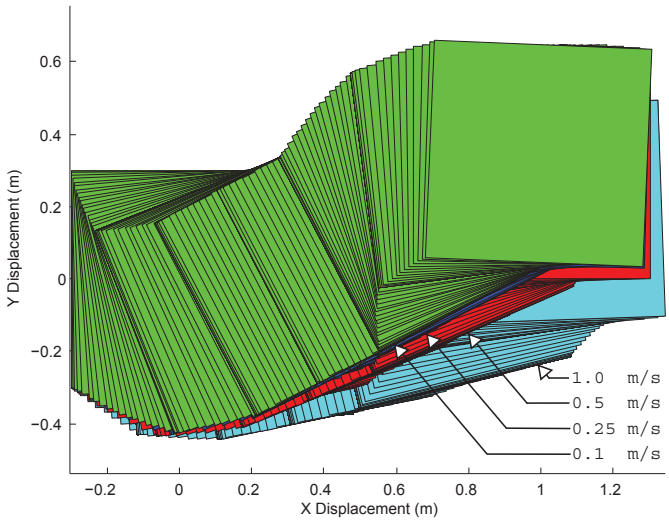


FIGURE 3.12: Comparing the response to an equivalent input at various velocities: 0.1, 0.25, 0.5 and $1.0m/s$

Effect of Support Friction Coefficient Variation

To explore the effect of varying the coefficient of the support friction, a trajectory is generated to manipulate an object, using a support friction coefficient of 0.3, to a goal at [1 0.3 0]. Following this, the input trajectory is executed with a variety of different support friction coefficients ranging from 0.1 to 0.7 to observe the effect of a support friction coefficient, different to that which was planned, on the final object position. Reducing the friction coefficient, as detailed in Table 3.3, will have a similar effect as increasing the pushing velocity since it reduces the support friction forces, thereby increasing the significance of the dynamic forces acting on the system. However, unless the system experiences extremely low support coefficients of friction, such that are rarely experienced in practice, the variation of the response is small.

Friction Coefficient	Linear Error (m)	Rotational Error (rad)
0.1	-0.018	0.011
0.2	-0.012	0.011
0.3	-0.010	0.012
0.4	-0.009	0.012
0.5	-0.009	0.012
0.6	-0.007	0.015
0.7	-0.001	0.009

TABLE 3.3: Final position error due to variation in friction coefficient

Effect of Object Mass or Moment of Inertia Variation

To explore the effect of varying the object mass or moment of inertia, a trajectory is generated to manipulate an object, using a mass of $2kg$ and a moment of inertia of $0.03kg/m^2$, to a goal at [1 0.3 0]. Following this, the input trajectory is executed with a variety of object masses ranging from $0.5kg$ to $5kg$ and moments of inertia ranging from $0.003kg/m^2$ to $0.083kg/m^2$ to observe the effect of a mass or moment of inertia, different to that which was planned, on the final object position. Section 2.4.2, predicts that varying the object mass by changing the object density will have no effect on the object motion as the object inertia and the support friction forces will both vary in proportion, and as such, the balance of forces in the system will remain unchanged. Again, this effect is only evident under constant velocity pushing conditions where the required pushing force is applied to maintain an appropriate pushing velocity. Table 3.4 clearly exhibits this result. A similar result is obtained when varying the object moment of inertia while keeping the mass constant as the system response is only affected during the transient period and the steady state response of the system is unchanged. Since the system settles into this steady state quite rapidly during execution it is understandable that the overall system response shows little variation, as seen in Table 3.5.

Mass (kg)	Linear Error (m)	Rotational Error (rad)
0.5	-0.010	0.012
1	-0.010	0.012
2	-0.010	0.012
3	-0.010	0.012
4	-0.010	0.012
5	-0.010	0.012

TABLE 3.4: Final position error due to variation in mass

Moment of Inertia (kg/m^2)	Linear Error (m)	Rotational Error (rad)
0.003	-0.010	0.012
0.013	-0.010	0.012
0.030	-0.010	0.012
0.053	-0.010	0.012
0.083	-0.006	0.009

TABLE 3.5: Final position error due to variation in moment of inertia

Effect of Support Friction Variation

The support friction can vary in two ways. The location of an individual support point can move relative to the object frame of reference. Alternatively, the support friction coefficient of an individual support point may vary, usually resulting in a shift in the center of friction. The effect of such variations is explored by generating a trajectory to manipulate an object with the support locations arranged in a square of $0.3m$ side length and a support friction coefficient of 0.3. Following this, the input trajectory is executed with a support point side length varying from $0.1m$ to $0.5m$ to observe the effect of a support pattern, different to that which was planned, on the final object position. This same input trajectory is then executed with the support friction coefficient of each support individually raised and lowered to observe the resulting variation in the final object position.

Table 3.6 shows the effect of varying the size of the support pattern. The support pattern is increased incrementally from a square with a side length of $0.1m$ to a side length of $0.5m$. The results reveal that support patterns which are clustered towards the center of the object allow the object to rotate more easily resulting in an undershoot in the tested scenario. Conversely, moving the support locations further away from the center makes it more difficult for the object to rotate which creates an overshoot in the tested scenario. Thus, even if the symmetry of the friction distribution is maintained so that the friction center remains at the same location, the ease with which an object will rotate in response to a push may change and result in a motion variation.

Support Pattern (m)	Linear Error (m)	Rotational Error (rad)
0.1	-0.100	0.014
0.2	-0.056	0.042
0.3	-0.010	0.012
0.4	0.076	0.077
0.5	0.125	0.248

TABLE 3.6: Final position error due to variation in support location

Table 3.7 shows the effect of altering the friction coefficient at individual support points, thus, causing the friction center to relocate. The support friction coefficient for each individual support point, labeled in Figure 3.13, is altered sequentially while the remaining support points maintain an unchanged friction coefficient of 0.3. In the tested scenario, the imbalanced friction center produces a system which can rotate more easily in one direction than the other. As a result, variations which cause the friction center to rise above the centerline result in an overshoot in the final displacement, where as, situations where the friction center is located below the centerline experience an undershoot in the final position.

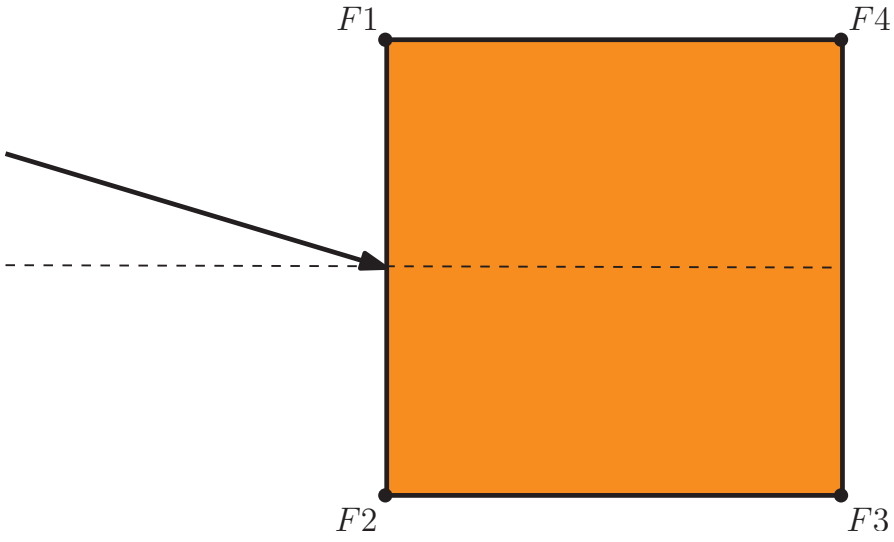


FIGURE 3.13: Location of friction support points referred to by Table 3.7

Support Point	Friction Coefficient	Linear Error (m)	Rotational Error (rad)
F1	0.4	0.039	0.089
F2	0.4	-0.087	0.002
F3	0.4	-0.068	0.004
F4	0.4	0.032	0.075
All	0.3	-0.010	0.012
F1	0.2	-0.089	0.005
F2	0.2	0.036	0.084
F3	0.2	0.066	0.087
F4	0.2	-0.118	0.024

TABLE 3.7: Final position error due to variation in friction center location due to imbalanced friction co-efficients at support points

3.4 Conclusions

This chapter builds on the knowledge of the object behavior to develop pushing trajectories which deliver an object to a desired goal position and orientation. This task is cast as a constrained minimization problem and solved to produce near time optimal pushing trajectories. These trajectories utilize the environmental interaction to resolve the limitations imposed by the under-actuated nature of the system, with solutions generated for both point pushing and curved fence pushing arrangements. Investigation of the system sensitivity to variations in the environmental parameters confirms that pushing rapidly will cause a reduction in system performance if the pushing velocity is different to that which is expected. A similar result is experienced if the support friction coefficient is lowered, however, for any effect to be noticeable the friction must be reduced below levels which are commonly encountered in practice.

The object mass and moment of inertia has very little to no influence on system performance, largely due to the constant velocity nature of the pushing input. However, variation in the support point location or the friction center location produced significant performance errors. As support friction is constantly changing in practice and difficult to measure or estimate, developing strategies for compensating for support friction variation is an important objective. The following chapter investigates techniques for selecting control actions in a way which is both amenable to real time operation and robust to variations in the environmental conditions.

Chapter 4

Real Time Control for Pushing Operations

The focus of Chapter 2 is on understanding the manipulation of an object by pushing with a mobile robot where contact between the robot and object is made at a single point. One consequence of this manipulation mode is that the system formed is both under-actuated and unstable, except in a limited set of scenarios where pushing is performed with a curved fence. The open-loop, near time optimal pushing trajectories for moving an object between arbitrary start and goal locations derived in Chapter 3 demonstrate the fact that both problems of under-actuation and stability can be overcome by exploiting the dynamic equations of motion of the system, given a full knowledge of the physical parameters governing the system behavior. Although the trajectories derived are shown to be robust to changes in some of the physical parameters, they are clearly not suitable for implementation in a real world system due to issues of computational cost and robustness to changes in the interaction between the object and its environment. This chapter investigates the possibility of using the trajectories derived in Chapter 3 as the basis for a feedback control algorithm to overcome these limitations.

4.1 Literature Review

As pushing with a single point forms an unstable system it is essential that some form of control is implemented to stabilize the system. Okawa and Yokoyama approach this problem by treating the robot and object as a multi-body system coupled by a rotary joint with its axis perpendicular to the support plane [46]. A feedback control law is presented to push the object towards the goal. In this case the error was calculated based on the angle between the robot, object and goal and was minimized to ensure that the object was steadily pushed towards the goal. As there was no attempt made to control the rotation of the object or to ensure that it arrived in a particular orientation there

was no requirement for path planning to be performed. Instead it was sufficient for the robot to calculate the instantaneous error and appropriate control action. This control strategy is applied to a robot manipulating a box in a corridor-like environment [45]. Later an improved control strategy was proposed by Kurisu and Yoshikawa [25] which allowed both the position and orientation to be controlled while the object is pushed to a goal location. The control is achieved by locating a point on the object termed the pseudo-center at which the motion of the sliding object can be approximated by the motion of a non-holonomic wheeled robot. Once the pseudo-center is located, a traditional controller for a non-holonomic wheeled robot can be applied to track the provided trajectory to the goal location. This result was one of the first to successfully control both the position and orientation of a sliding object, although it relies on the assumption that the system is quasi-static.

The control and stabilization of general under-actuated robotic systems has been studied [6, 10, 11]. In [10], De Luca et. al. investigate the control and stabilization of an under-actuated planar robot consisting of two rotary joints, where only the joint located in the base is actuated and the other is a frictionless passive joint. Their approach uses a two phase method consisting of an "alignment" phase, where the active joints are moved to their desired positions, followed by a "transition" phase where the active joints are cyclically driven such that the passive joints converge to their desired position. The transition phase is accomplished using an iterative state steering approach based on an open loop control which steers the system closer to the desired equilibrium state during each iteration. While this methodology is demonstrated, through simulation, to successfully bring both the passive and active joints of the robot arm to a desired orientation if the motion is restricted to the first quadrant, the result is not directly applicable to pushing scenarios as the oscillating nature of the transition phase will violate the pushing constraints resulting in a loss of contact with the pushed object.

Goh et.al. [19] developed an optimal feedback controller capable of driving a non-linear system from an arbitrary initial state to a fixed final state in minimum time. The controller was implemented with a feed forward multilayer neural network trained using open-loop optimal control data generated off-line. The controller was shown, through simulation, to be effective at bringing a two link robot arm with driven joints to a desired final state in the presence of both sensor/actuator noise and parameter variation. One of the strengths of this approach is that the non-linear control space is mapped in advance allowing the appropriate control action to be calculated very rapidly during operation. A similar approach may be applied to the problem of pushing an object to a desired pose allowing the appropriate control action for any object pose to be rapidly selected from an existing map of the optimal feedback control law.

Recently Igarshi et.al. [21] presented a simple method for pushing an object to a desired goal location, although the final orientation of the object is not controlled. Knowing the global position of both the object and goal, a set of paths based on a modified

dipole force field are generated for the robot to follow. The nature of these paths is such that the robot will tend to push the object towards the goal location regardless of the relative size of the robot and object. While this methodology does not require knowledge of the size, mass or friction properties of the pushed object, it is only applicable to slow motions where the quasi-static assumption is valid. In addition, the methodology does not account for the object orientation when generating the control action, and as such, the final orientation of the object at the goal position is arbitrary. The authors indicate that while this algorithm has the potential to be improved to be applicable in situations where rapid manipulation is required as well as situations where the object is pushed by a non-holonomic robot, further work is required to achieve this.

4.2 Feedback Control Law for Single Point Pushing

To successfully execute real world manipulation tasks, the robot must be able to select control actions quickly to enable the system to operate in real time. The system must also detect and compensate for errors in the object motion caused by uncertainty and variation within the system parameters. An approach similar to that of Gow et.al. [19], where the non-linear near time-optimal control problem is solved for manipulation from an arbitrary starting pose to a fixed terminal pose, is applicable to the manipulation problem under consideration. This section provides a technique for deriving a state based feedback law which exploits information from the open-loop trajectories derived in Chapter 3 to map the current object pose to the control action which will act to move the object closer to the desired position and orientation.

A state feedback controller needs to generate a control signal as a function of the current state of the system. In the present example, the system state is defined by the pose and the velocities of the object being pushed. The control action is the pushing angle. There is no error feedback signal, in the traditional sense, as an appropriate control action for an "erroneous" object pose will be calculated to move the object towards the goal on the next iteration. During operation the feedback controller is sampled as required by the robot control system. The resulting control action is not dependent on the sampling frequency. Without any loss of generality, the assumption is made that the desired final state is the origin, as this can be achieved for any given desired final state using a straightforward coordinate transform. Given that optimal trajectories aiming to reach identical goal positions do not intersect in the state space, there is a unique control action that corresponds to any given system state. A data base relating the optimum control actions to state space can be generated by solving the near minimum time trajectory planning problem from a variety of initial poses with the final state as the origin. When the data available is of sufficient density, a straightforward interpolation technique can be used to compute the control action for a given state, thus resulting in a feedback control law.

In the following, the feedback control law is generated as follows. The off-line algorithm presented in Chapter 3 is used to generate 20 optimal trajectories for each of 15 different starting orientations equally spaced from $0rad$ to $\frac{\pi}{2}rad$, resulting in a total of 300 trajectories. For each starting orientation, the 20 goal locations are equally spaced $0.1m$ apart, vertically at $x = 1m$. Symmetry in the sliding object behavior allows a mirror of these trajectories to be used for starting orientations from $0rad$ to $-\frac{\pi}{2}rad$ for a total of 600 trajectories. Each optimization is restarted multiple times with different initial guesses to ensure that a true minimum trajectory has been found. The data set for the interpolation is formed by the points along each trajectory and the current control action at each point. Interpolation is based on Delaunay triangulation as implemented by the triscatteredinterp function provided with the Matlab software package is used to generate the control action corresponding to a given state, which can then be directly acted upon.

The important question is whether this feedback control law is robust to changes in the system parameters as the data used in the derivation is generated for a system with a given set of physical properties. Evaluating this question will be the focus of the numerical case studies presented in the following section.

4.3 Selected Numerical Case Studies

Figures 4.1 and 4.2 compare the input trajectories generated by the optimizer presented in Chapter 3 and the feedback control law presented here for initial object poses of [-0.9 0.3 0] and [-0.9 -0.7 0.5] respectively. While the trajectories generated by the optimizer are piecewise constant and the trajectories generated by the feedback control law are continuous in nature, for a given starting pose, both trajectories follow the same general form. The resulting object motion, shown in Figures 4.3 and 4.4, exhibits a comparable response for both methodologies.

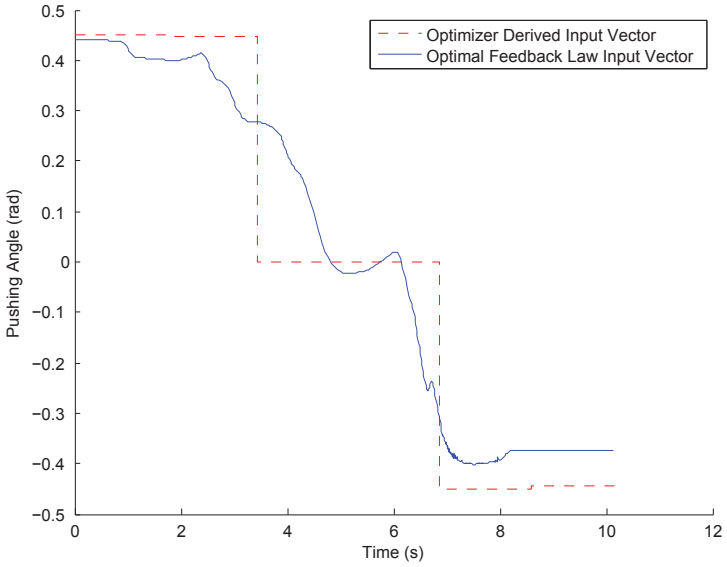


FIGURE 4.1: Control action for object manipulation generated by optimizer and feedback control law

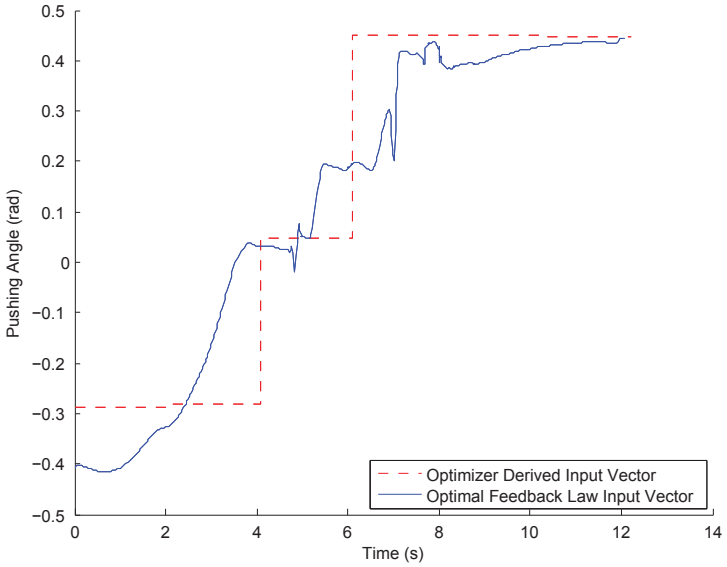
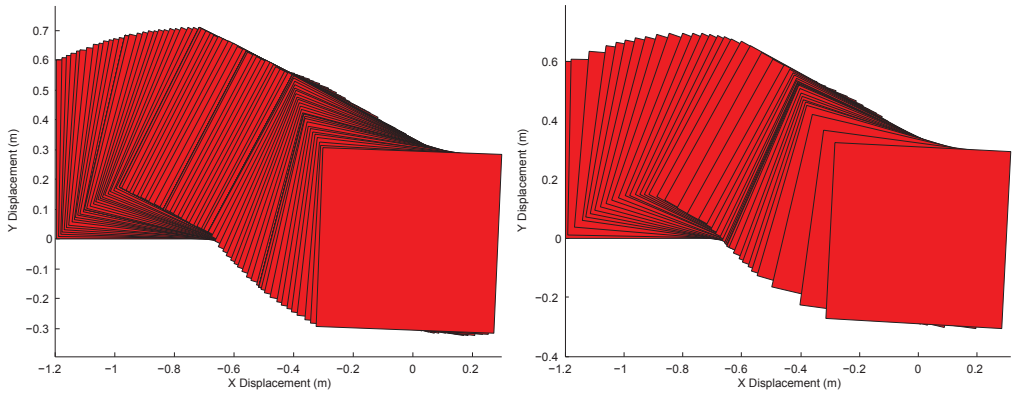
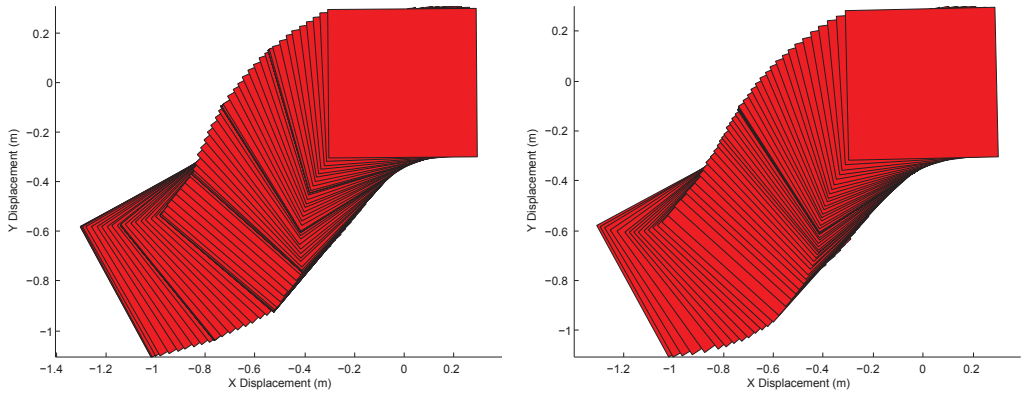


FIGURE 4.2: Control action for object manipulation generated by optimizer and feedback control law



(a) (b)

FIGURE 4.3: Object trajectory for (a) optimizer and (b) feedback control law



(a) (b)

FIGURE 4.4: Object trajectory for (a) optimizer and (b) feedback control law

Effect of Support Friction Variation

In practice, it is unlikely that the support friction will remain perfectly constant for the duration of the manipulation. Variation in the support friction is modeled as Gaussian noise similar to that shown in Figure 4.5 for the purpose of examining such a change. For a starting pose of [-0.9 -0.7 0.5], the input trajectory produced by the feedback control law in the presence of Gaussian noise, shown in Figure 4.6, is similar to the baseline response shown in Figure 4.2. The resulting object motion generated by both the optimizer and feedback control law based trajectories, shown in Figure 4.7, are similar and both successfully able to reach the goal indicated by the black square.

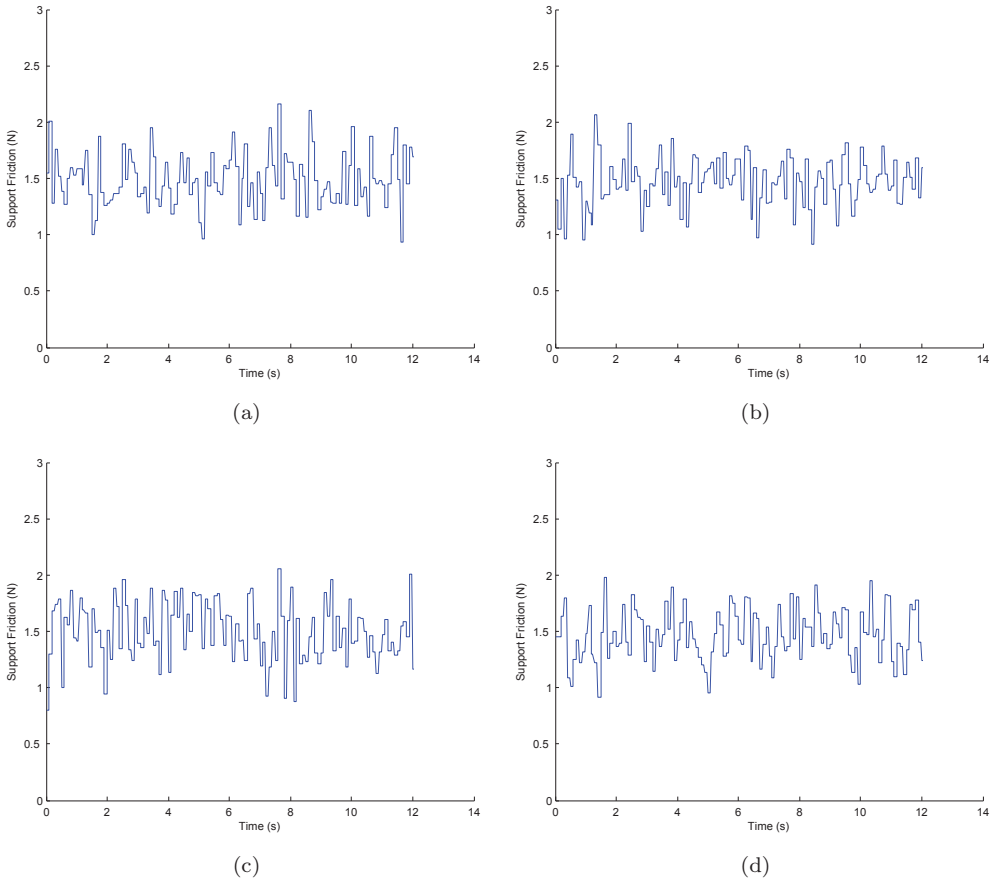


FIGURE 4.5: Variation of support friction coefficients over time for a Gaussian disturbance. Support points located (a) Top Left (b) Top Right (c) Bottom Left and (d) Bottom Right

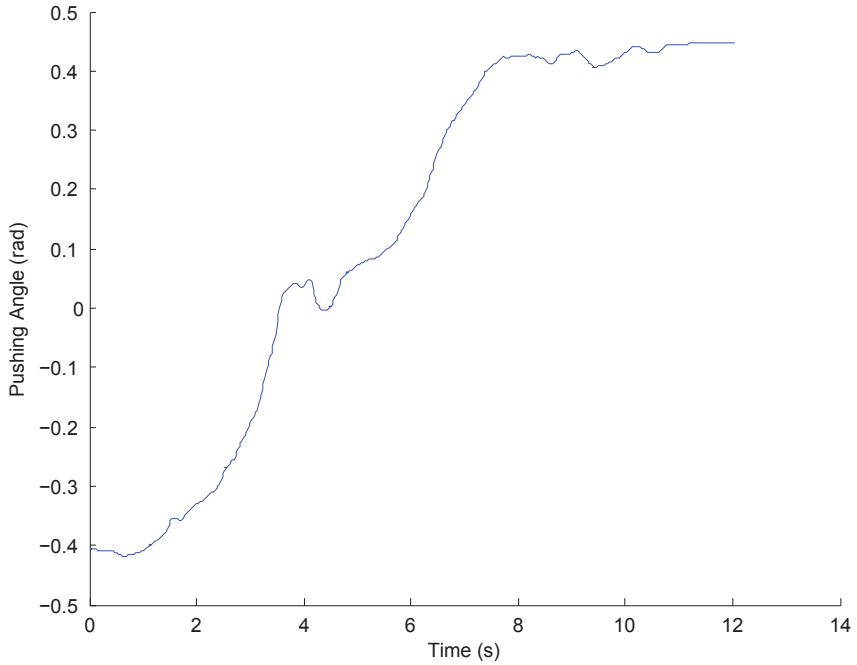
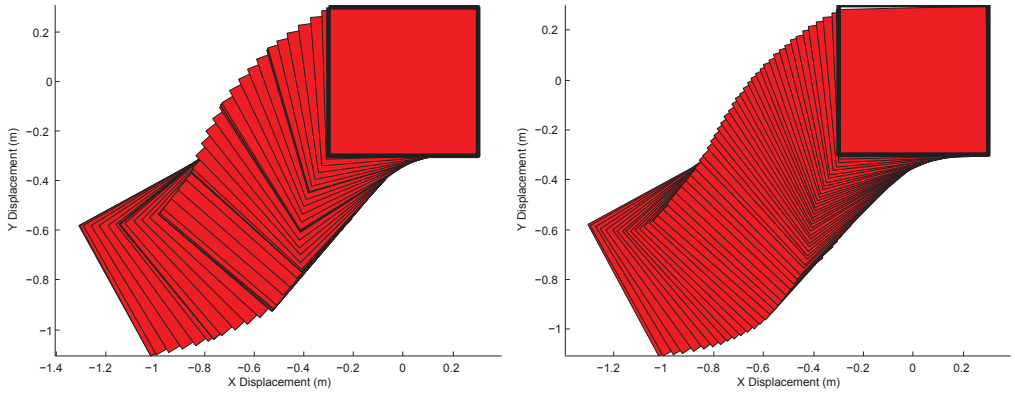


FIGURE 4.6: Input trajectory for object manipulation generated by the feedback control law in the presence of a Gaussian disturbance



(a) (b)

FIGURE 4.7: Object trajectory for (a) optimizer and (b) feedback control law in the presence of Gaussian disturbance

Effect of a Temporary Disturbance

The effect of a temporary disturbance is investigated by considering a manipulation action, commencing from [-0.9 -0.7 0.5], where the support friction on one side of the object increases for a short period of time, as shown in Figure 4.8. This situation can be experienced in practice if, for example, part of the object briefly passes over a sticky area on the support surface. The magnitude of the disturbance is varied between zero and 100%, that is, a support friction of twice the normal level.

As the input trajectory generated by the optimizer is open-loop in nature, there is no mechanism to recover from the disturbance and the final object pose will not be at the goal location, as shown in Figure 4.10(a). Figure 4.9(a) shows the motion of the object center as the magnitude of the disturbance increases. The object motion becomes increasingly erroneous as the pushing action proceeds and as the magnitude of the disturbance increases. In comparison, the feedback control law provides a mechanism to recover as errors become apparent in the object response. In effect, the feedback control law is always calculating the instantaneous control action required for manipulation from the current object pose to the goal. Hence, if the object motion is disturbed, a control action to bring the object from the disturbed pose to the goal will be calculated enabling the manipulation task to be successfully completed, as shown in Figure 4.10(b). Figure 4.9(b) shows the motion of the object center as the magnitude of the disturbance increases. The object motion becomes increasingly erroneous for the period of the disturbance as the magnitude of the disturbance increases. However, in all cases, the object motion converges to the desired goal pose once the disturbance is removed. The input trajectory generated by the feedback control law in this case, shown in Figure 4.11, exhibits a similar general form to the undisturbed trajectory, however, the period of negative pushing angle at the beginning of the manipulation is reduced as the system attempts to compensate for the support friction disturbance. The spikes in the middle of the input trajectory are caused by local roughness in the interpolant.

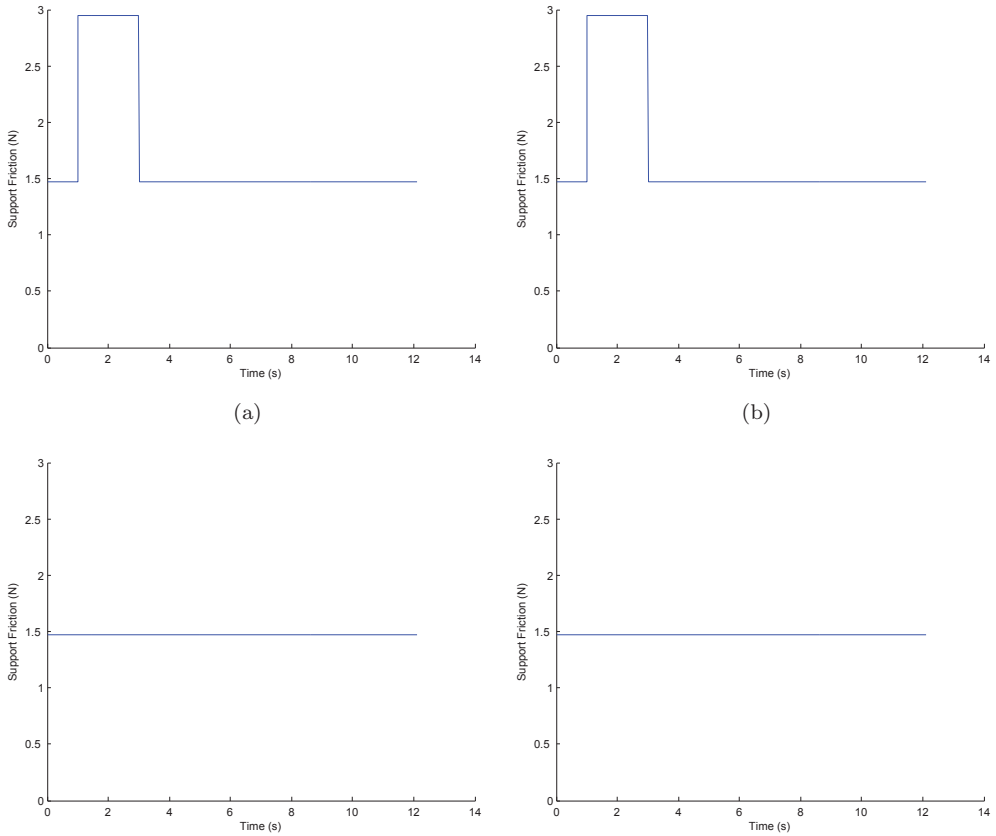


FIGURE 4.8: Variation of support friction coefficients over time for a temporary disturbance. Support points located (a) Top Left (b) Top Right (c) Bottom Left and (d) Bottom Right

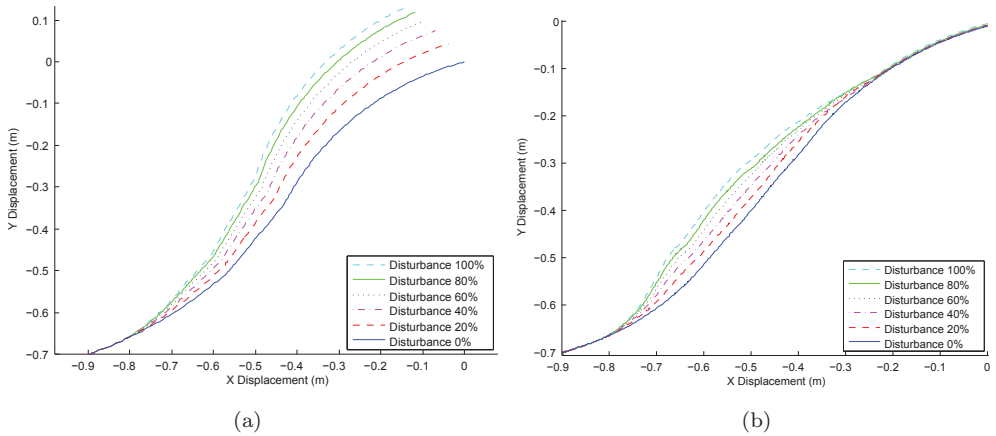
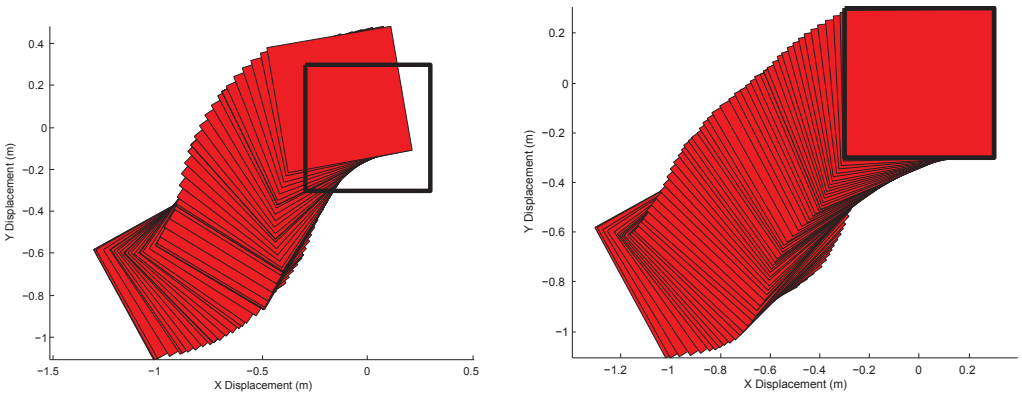


FIGURE 4.9: Object center trajectory for (a) optimizer and (b) feedback control law for a temporary disturbance



(a) (b)

FIGURE 4.10: Object trajectory for (a) optimizer and (b) feedback control law for a temporary disturbance

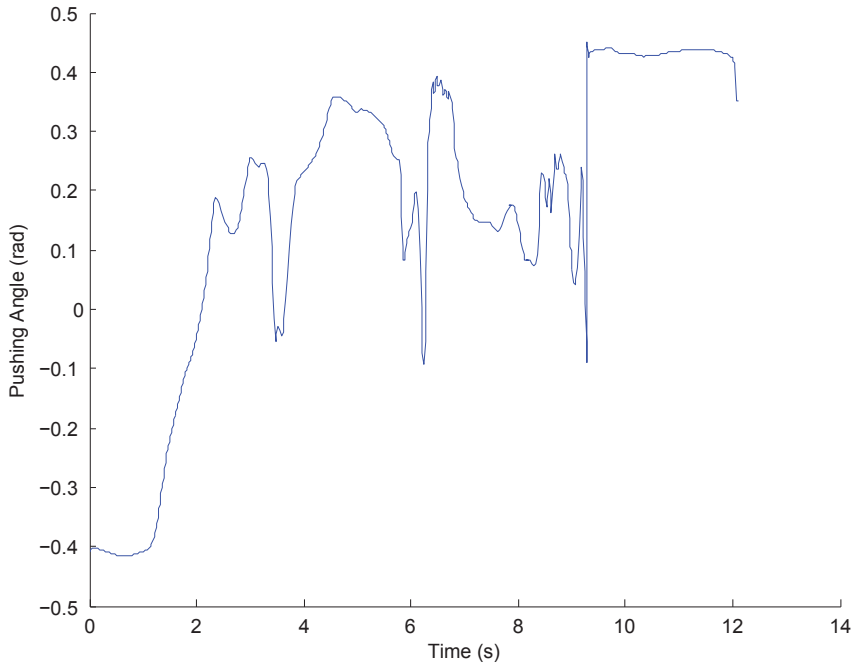


FIGURE 4.11: Input trajectory for object manipulation generated by the feedback control law in the presence of a temporary disturbance

Effect of a Sustained Disturbance

The effect of a sustained disturbance is investigated by considering a manipulation action, commencing from [-0.9 -0.7 0.5], where the support friction on one side of the object doubles for an extended period of time, as shown in Figure 4.12. This situation can be experienced in practice if, for example, an additional mass is placed on one side of the object increasing the support friction on that side of the object. The magnitude of the disturbance is varied between zero and 50%, that is, a support friction of one and a half times the normal level.

Again, as the input trajectory generated by the optimizer is open-loop in nature, there is no mechanism to recover from the disturbance and the final object pose will not be at the goal location, as shown in Figure 4.14(a). Figure 4.13(a) shows the motion of the object center as the magnitude of the disturbance increases. Again, the object motion becomes increasingly erroneous as the pushing action proceeds and as the magnitude of the disturbance increases. However, as the disturbance is applied for the majority of the manipulation action, the deviation experienced is greater than for a temporary disturbance of comparable magnitude. While the feedback control law does provide a mechanism to feed back information as errors become apparent in the object response, the model used to select the instantaneous control actions assumes different friction properties. Hence, the control actions will always be less than what is required to bring the object to the goal from the current location. As a result, the feedback control law will manipulate the object closer to the desired goal pose than the optimizer based approach but it will ultimately be unsuccessful at compensating for a sustained disturbance, as shown in Figures 4.14(b) and 4.10(b). The input trajectory generated by the feedback control law in this case, shown in Figure 4.15, again exhibits a similar general form to the undisturbed trajectory, however, the control action saturates at a maximum positive pushing angle early in the manipulation as the system attempts to compensate for the sustained support friction disturbance.

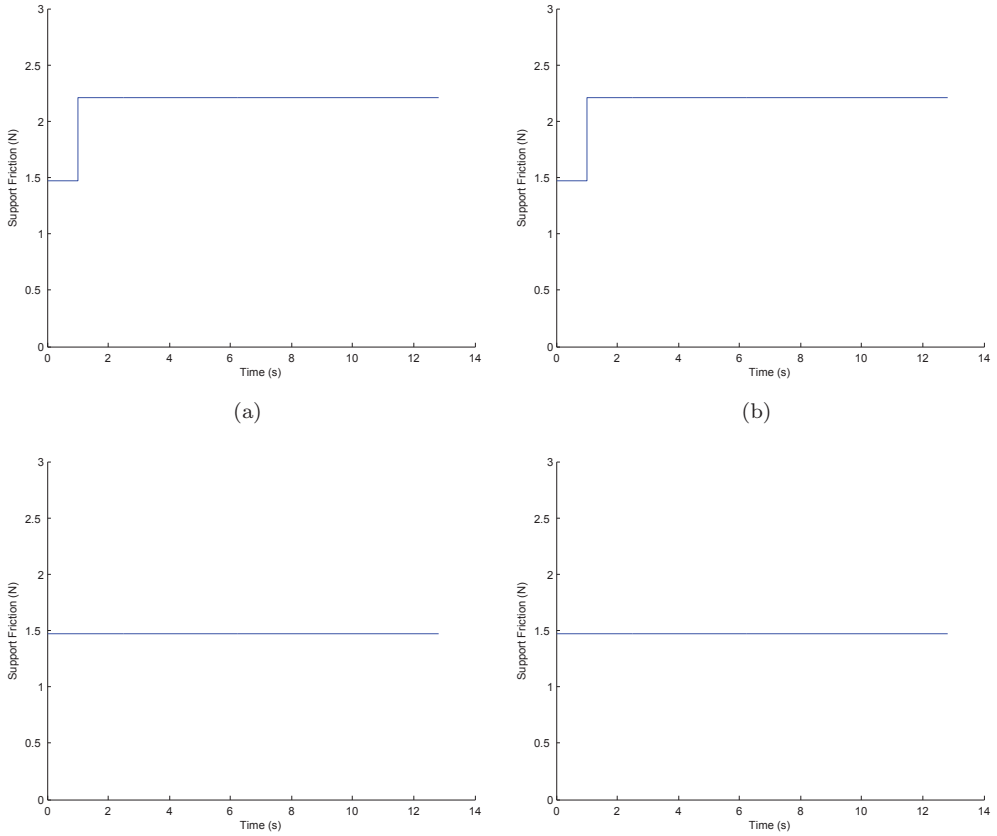


FIGURE 4.12: Variation of support friction coefficients over time for a sustained disturbance. Support points located (a) Top Left (b) Top Right (c) Bottom Left and (d) Bottom Right

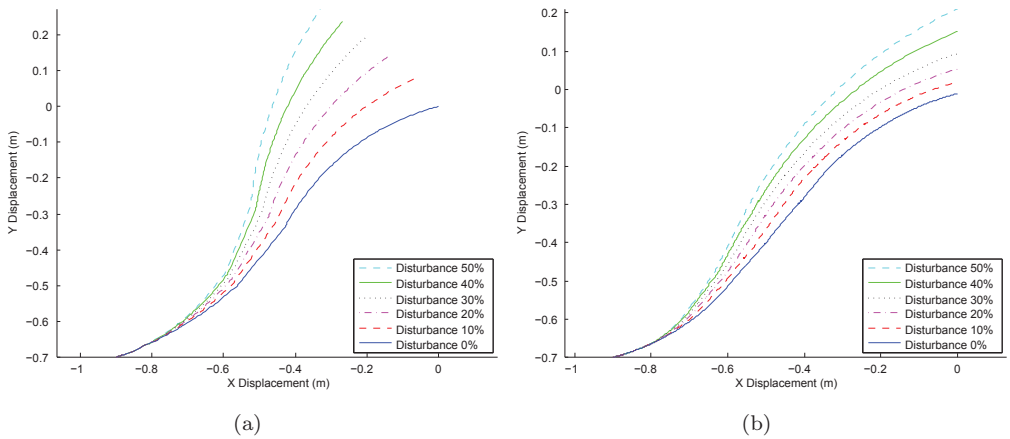
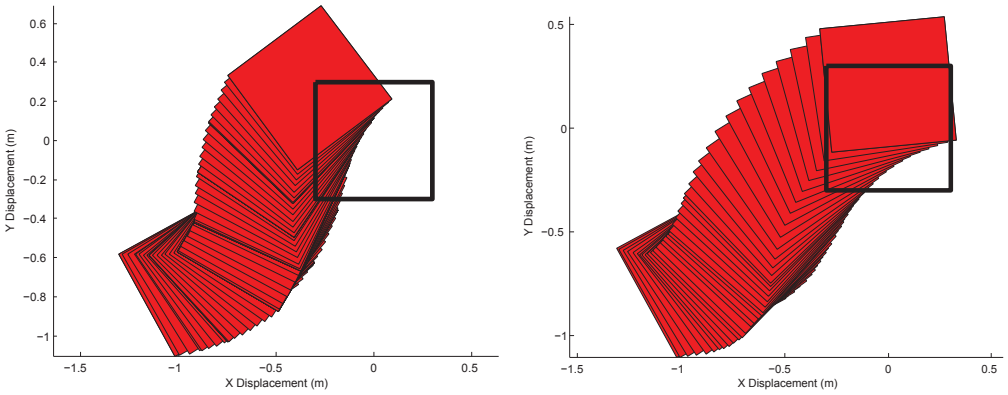


FIGURE 4.13: Object center trajectory for (a) optimizer and (b) feedback control law for a sustained disturbance



(a) (b)

FIGURE 4.14: Object trajectory for (a) optimizer and (b) feedback control law for a sustained disturbance

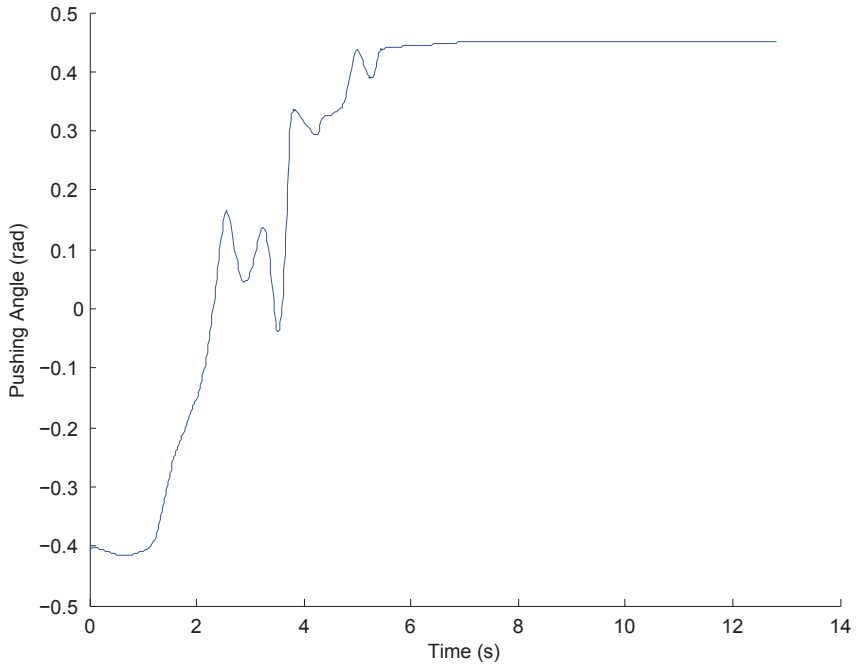


FIGURE 4.15: Input trajectory for object manipulation generated by the feedback control law in the presence of a sustained disturbance

4.4 Conclusions

This chapter builds on the optimization based trajectory planner presented in Chapter 3 to develop a feedback control law which facilitates the rapid generation of control actions required during real-time operation and provides some error recovery capability, enabling an object to be manipulated to a desired goal in the presence of some disturbance to the support friction distribution. Investigation of the system response confirms that the trajectories generated by the feedback control law are equivalent to those produced by the optimizer. Also, the feedback control law was demonstrated to be effective in recovering from a temporary disturbance in the support friction distribution and able to deliver the object to the desired goal regardless. In the presence of sustained variations in the support friction distribution however, although the feedback control law was able to deliver the object closer to the desired goal, neither the optimizer, nor the feedback control law were able to deliver the object to the desired goal.

When the dynamic forces in the system are significant, a feedback control law generated for a particular pushing velocity will not be suitable for manipulation using a pushing velocity which is significantly different. This is due to the discrepancy between the expected object behavior and actual object behavior in this situation, thus, as for the sustained variation in friction distribution, the object will not be manipulated to the desired goal. Future work may investigate online estimation techniques to enable the feedback control law to be updated to properly represent the expected object motion subsequent to a sustained variation in the system parameters.

Chapter 5

Conclusions

This thesis presented an investigation into the motion of an object sliding on a flat support surface in the presence of friction and in response to a pushing action imparted by a mobile robot. The contact between the robot and the object was made at a single point, either a fixed location such as a perimeter vertex or a variable location such as the contact point between two curved surfaces free to roll along each other. The dynamic equations of motion were derived and implemented in simulation to explore the system behavior and to determine which system parameters have a strong influence over the system response. The potential for stable pushing conditions to be encountered while pushing with a curved fence was explored. A trajectory planning approach was presented which uses an optimization coupled with the dynamic equations of the system to produce near time optimal pushing trajectories to manipulate an object to a desired goal. Finally, the trajectory planner was used to produce a feedback control law to enable the robot real time operation to push an object to a goal and to recover from disturbances encountered.

5.1 Summary of Contributions

5.1.1 Investigation of the Behavior of a Sliding Object

The motion of an object sliding on a flat support surface in response to a constant velocity push at a single contact point was investigated with a view to enabling non-prehensile manipulation by a mobile robot. Both arrangements where the pushing contact is made at a single sharp point and arrangements where the pushing contact is made between two curved surfaces free to roll along one another were explored and the full dynamic equations of motion were derived. The equations of motion were implemented in simulation and used to explore the system response to a range of parameter variations including pushing velocity, object mass, moment of inertia, support friction coefficient

and support point location. The results showed that some parameters, such as pushing velocity and support friction, are closely linked to the response of the system where as others, such as object mass, have very little if any effect on the system behavior. It was also noted that an object pushed at a fixed location will behave in a manner similar to a non-holonomic wheeled vehicle when the pushing angle is maintained constant. While the non-linear nature of the dynamic system makes it infeasible to describe the motion variables mathematically, the experimental results empirically demonstrate that the comparison made by Kurisu and Yoshikawa [24, 25] for the quasi-static case may be extended to systems with non-trivial dynamic effects.

5.1.2 Discovery and Investigation of Stable Pushing Regions with a Curved Fence

While investigating the behavior of systems where the pushing contact is made at a point between two curved surfaces, it was discovered that certain system topologies will display self stabilizing behavior. This behavior has been known to exist in systems where the pushing contact is made between two flat surfaces or at multiple points of contact, however, this is the first time that this behavior has been investigated for the single contact point case. It was found that the stability of any particular system is largely defined by the location of the friction center and the moment which it generates about the pushing contact point. For a stable configuration to exist, a restoring moment must be generated about the contact point if the system experiences a small disturbance. The location and size of the stable regions were computed based on the location of the friction center, pushing contact point, robot and object radius. The extent of the stable regions were tested through dynamic simulation and found to correspond to the computed stable regions.

5.1.3 Synthesis of Near Time Optimal Pushing Trajectories

A methodology has been presented to derive near time-optimal trajectories that overcome the limitations imposed by the under-actuated nature of the system and enable the delivery of sliding objects to a desired goal. The planning task was posed as a constrained optimization problem and solved using the full dynamic system model. The optimization output is a piecewise constant function in time for the angle at which the robot must push the sliding object. Selected numerical case studies explored the impact of the number of partitions used in the piecewise constant approximation of the input vector, the effect of variation in pushing velocity and the effect of variation in the support friction distribution when planning near time optimal trajectories. The effect of the pushing robot radius when planning trajectories for systems with contact formed between two curved surfaces was also explored. A sensitivity study analyzes the effect

of variation in key system parameters on the performance of the open loop trajectories generated by the optimization.

5.1.4 Control for Real Time Execution

A methodology for the rapid selection of control actions in a manner suitable for real time applications was presented. This methodology also allows the system to compensate for variation experienced in environmental parameters such as friction distribution in practical applications. A wide range of trajectories were generated for a variety of starting locations and used to obtain a control law that interpolates between these trajectories to generate an instantaneous control action for the current object pose. The system was implemented in simulation and was found to be able to effectively reject non-persistent perturbations. Persistent perturbations where found to be detrimental to the system performance although the control methodology partly mitigated the effect of the perturbation when compared to the open loop control case.

5.2 Discussion of Limitations and Future Work

Autonomous non-prehensile manipulation is one of the challenges for robotic research. This thesis has addressed the challenging issue of generating pushing trajectories for the manipulation of objects where contact is made at a single point and the dynamics of the system may be significant. However, fully realizing an autonomous system which operates in an environment with unknown physical parameters is still a challenging development exercise.

In the proposed approach, the robot is assumed to be sufficiently powerful so as to precisely execute the requested constant velocity control actions. As such, the research does not investigate the impact of the object parameters on the robot performance. In addition, the pose of both the robot and the object are assumed to be known with absolute certainty throughout the manipulation action, therefore, no problems associated with robot or object localization have been covered by this research.

The operating environment for the system is assumed to be a flat horizontal plane, free of obstacles and the manipulation of the object is assumed to be achieved with a single manipulation action. As such, the investigation was limited to manipulations where the robot was not required to change between multiple pushing locations to deliver an object to the goal. Future work to expand the planner capabilities to accommodate relocations in the pushing point will greatly enhance the maneuverability of the system, especially in the presence of obstacles. Such a system will need to determine the optimal locations at which to push the object during each phase of the manipulation, as well as, determine how long each phase of the overall manipulation action must persist.

The time currently taken for the computation of the near time optimal open loop trajectories precludes them from being directly applied in a real time robotic system. One potential direction for future work is the reduction of the computation time to a practical duration by implementing the dynamic equations of the system in a more efficient programming language such as C++. The optimization itself may be further hastened by investigating and implementing more sophisticated optimization methods.

As the support friction is very difficult to measure and is likely to vary during operation, no attempt is made to estimate the support friction distribution throughout this research. The feedback control law is currently able to reject temporary perturbations in the support friction distribution but suffers if a sustained perturbation is experienced. The performance of the system may therefore be improved if these sustained perturbations can be detected and the the feedback control law adjusted to suit the new set of friction parameters. This will greatly improve the performance of a system where slow or infrequent variations in the support distribution are apparent. If a stable pushing approach using a curved fence is implemented, the success of the manipulation action largely depends on the location of the friction center. If the robot takes actions to influence the friction center to take a more favorable position, the manipulation action will be more likely to succeed. A potential method of moving the friction center towards the contact point is to place a weight on the object, close to the pushing contact point, or to press down on the object using a robotic manipulator.

Appendix A

Nomenclature

$\boldsymbol{p}_c^u$	position of the contact point in the global frame
$\boldsymbol{p}_o^u$	origin of the object coordinate frame in the global frame
$\boldsymbol{p}_c^o$	position of the contact point in the object frame
$\dot{\boldsymbol{p}}_c^o$	velocity of the contact point in the object frame
$\ddot{\boldsymbol{p}}_c^o$	acceleration of the contact point in the object frame
$\boldsymbol{p}_i^o$	position of the ith support point in the object frame
θ	rotation angle of the object coordinate frame in the global frame
ϕ	direction of the pushing velocity in the object frame
v_c	magnitude of the pushing velocity
$\boldsymbol{f}_c^u$	pushing force vector in the global frame
$\boldsymbol{v}_i^u$	velocity vector at ith support point in the global frame
$\boldsymbol{f}_i^u$	friction force vector at ith support point in the global frame
$\boldsymbol{f}_f^u$	unified support friction vector in the global frame
n	number of support points
N_i	normal reaction force at ith support point
μ_i	coefficient of friction at ith support point
m	mass of the sliding object
I_c	moment of inertia about the pushing contact point
ω	angular velocity of the sliding object

$\dot{\omega}$	angular acceleration of the sliding object
$\boldsymbol{n}_{\phi}^{o}$	direction vector of the pushing velocity
U_{xy}	global inertial reference frame
O_{xy}	object coordinate frame
R_{xy}	robot coordinate frame
R_{o}^{u}	object frame to global frame rotation matrix
R_{r}^{u}	robot frame to global frame rotation matrix
R_{o}^{r}	object frame to robot frame rotation matrix
$\dot{\boldsymbol{p}}_{c}^{u}$	velocity of the contact point in the global frame
$\dot{\boldsymbol{p}}_{o}^{u}$	velocity of the object coordinate frame in the global frame
$\ddot{\boldsymbol{p}}_{o}^{u}$	acceleration of the object coordinate frame in the global frame
$\psi(\phi)$	function of steady state object angular velocity under quasi-static conditions
$\dot{\theta}$	object angular velocity
$\ddot{\theta}$	object angular acceleration
$\boldsymbol{r}_{pc}$	radius of the pushing robot
$\boldsymbol{r}_{sc}$	radius of the object
$\boldsymbol{r}_{fc}$	radius to the friction center in the object frame
β	angle to $\boldsymbol{r}_{fc}$ from the positive x axis in the object frame
$\boldsymbol{d}_{fc}$	vector from the contact point to the friction center
α	angle to $\boldsymbol{d}_{fc}$ from the positive x axis in the robot frame
ϵ	angle to the contact point in the object frame
M_c	moment about the contact point
$\boldsymbol{t}_{pc}$	vector from the initial contact point to the current contact point in the object frame
$\boldsymbol{t}_{fc}$	vector from the initial contact point to the normal with the friction center
$\boldsymbol{n}_{fc}$	normal distance from the object boundary to the friction center

R_n	normal reaction force due to pushing contact friction
R_t	tangential reaction force due to pushing contact friction
μ_c	coefficient of friction between the robot and object
$\boldsymbol{v}_c$	pushing velocity vector
$\boldsymbol{d}_{pc}$	contact point displacement along a flat fence
$\boldsymbol{p}_r^u$	origin of the robot coordinate frame in the global frame
$\dot{\boldsymbol{p}}_r^u$	velocity of the robot in the global frame
$\ddot{\boldsymbol{p}}_r^u$	acceleration of the robot in the global frame
$\boldsymbol{p}_c^r$	location of the contact point in the robot frame
$\dot{\boldsymbol{p}}_c^r$	velocity of the contact point in the robot frame
$\ddot{\boldsymbol{p}}_c^r$	velocity of the contact point in the robot frame
x	displacement of the object in the global x direction
y	displacement of the object in the global y direction
$\dot{x}$	velocity of the object in the global x direction
$\dot{y}$	velocity of the object in the global y direction
$\ddot{x}$	acceleration of the object in the global x direction
$\ddot{y}$	acceleration of the object in the global y direction
η	object rotation in the robot frame
γ	robot rotation in the global frame
ρ	rotation of the pushing contact point in the robot frame
x_c^o	displacement of the pushing contact point in the x direction in the object frame
$\dot{x}_c^o$	velocity of the pushing contact point in the x direction in the object frame
$\ddot{x}_c^o$	acceleration of the pushing contact point in the x direction in the object frame
y_c^o	displacement of the pushing contact point in the y direction in the object frame

$\dot{y}_c^o$ velocity of the pushing contact point in the y direction in the object frame

$\ddot{y}_c^o$ acceleration of the pushing contact point in the y direction in the object frame

x_c^r displacement of the pushing contact point in the x direction in the robot frame

$\dot{x}_c^r$ velocity of the pushing contact point in the x direction in the robot frame

$\ddot{x}_c^r$ acceleration of the pushing contact point in the x direction in the robot frame

y_c^r displacement of the pushing contact point in the y direction in the robot frame

$\dot{y}_c^r$ velocity of the pushing contact point in the y direction in the robot frame

$\ddot{y}_c^r$ acceleration of the pushing contact point in the y direction in the robot frame

x_r^u displacement of the robot in the global x direction

$\dot{x}_r^u$ velocity of the robot in the global x direction

$\ddot{x}_r^u$ acceleration of the robot in the global x direction

y_r^u displacement of the robot in the global y direction

$\dot{y}_r^u$ velocity of the robot in the global y direction

$\ddot{y}_r^u$ acceleration of the robot in the global y direction

Appendix B

Glossary of Terms

Environment	A flat, horizontal surface on which a robot and object are positioned for manipulation.
Robot	A mobile robot capable of maneuvering in the environment and applying forces to the object.
Object	Some object to be manipulated. Assumed to slide freely in the environment.
Pose	The position and orientation of the object in the environment.
Support Plane	See Environment.
Support Point	A point of contact between the object and the support surface.
Contact Point	The point of contact between the robot and the object.
Support Friction	Friction experienced between the object and the support surface.
Contact Friction	Friction experienced between the robot and the object.
Non-prehensile	Without grasping.
Stable Push	An action where the system mechanics drive the object to a steady state relative to the robot.
Manipulation	The act of influencing the pose of an object.
Under-actuated	A system which encompasses more degrees of freedom than can be directly influenced.
Trajectory	A sequence of object positions as it moves between poses.
Fence	A rigid, smooth and vertical surface, either flat of curved, used to push an object.

Quasi-static	A system where the dynamic forces are negligible and may be ignored.
Interpolant	A surface created by interpolation.
End-effector	A robotic tool, usually mounted at the end of a robotic arm.

Appendix C

Derivation of Equations of Motion for Pushing with a Curved Fence

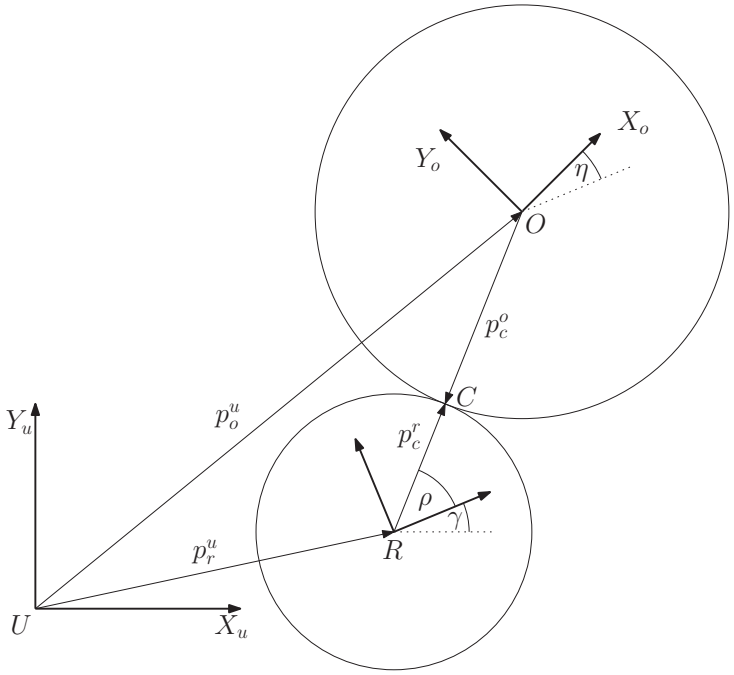


FIGURE C.1: A sliding object subject to an input velocity between two curved faces

Again U_{xy} is a global inertial reference frame, R_{xy} is a robot coordinate frame, fixed at the robot center and rotating with the robot, and O_{xy} is an object coordinate frame, fixed at the center of mass and rotating with the object. The position of O_{xy} and hence the position of the object center of mass relative to U_{xy} is given by

$$\boldsymbol{p}_o^u = \boldsymbol{p}_r^u + R_r^u \boldsymbol{p}_c^r - R_o^r \boldsymbol{p}_c^o \tag{C.1}$$

where R_r^u is the ground to robot rotation matrix given by

$$R_r^u = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \tag{C.2}$$

and R_o^r is the robot to object rotation matrix given by

$$R_o^r = \begin{bmatrix} \cos\eta & -\sin\eta \\ \sin\eta & \cos\eta \end{bmatrix} \tag{C.3}$$

Differentiating (C.1) we get

$$\dot{\boldsymbol{p}}_o^u = \dot{\boldsymbol{p}}_r^u + \dot{R}_r^u \boldsymbol{p}_c^r + R_r^u \dot{\boldsymbol{p}}_c^r - (\dot{R}_o^r \boldsymbol{p}_c^o + R_o^r \dot{\boldsymbol{p}}_c^o) \tag{C.4}$$

since

$$\dot{R}_r^u = \dot{\gamma} R_r^u D \tag{C.5}$$

and

$$\dot{R}_o^r = \dot{\eta} R_o^r D \tag{C.6}$$

simplify

$$\dot{\boldsymbol{p}}_o^u = \dot{\boldsymbol{p}}_r^u + \dot{\gamma} R_r^u D \boldsymbol{p}_c^r + R_r^u \dot{\boldsymbol{p}}_c^r - \dot{\eta} R_o^r D \boldsymbol{p}_c^o - R_o^r \dot{\boldsymbol{p}}_c^o \tag{C.7}$$

with

$$D = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{C.8}$$

Differentiating (C.7) we obtain the acceleration of the center of mass

$$\begin{aligned} \ddot{\boldsymbol{p}}_o^u \;=\; & \ddot{\boldsymbol{p}}_r^u + \ddot{\gamma} R_r^u D \boldsymbol{p}_c^r + \dot{\gamma} \dot{R}_r^u D \boldsymbol{p}_c^r + \dot{\gamma} R_r^u \dot{D} \boldsymbol{p}_c^r + \dot{\gamma} R_r^u D \dot{\boldsymbol{p}}_c^r + \dot{R}_r^u \dot{\boldsymbol{p}}_c^r + R_r^u \ddot{\boldsymbol{p}}_c^r \\ & - (\ddot{\eta} R_o^r D \boldsymbol{p}_c^o + \dot{\eta} \dot{R}_o^r D \boldsymbol{p}_c^o + \dot{\eta} R_o^r \dot{D} \boldsymbol{p}_c^o + \dot{\eta} R_o^r D \dot{\boldsymbol{p}}_c^o) - (\dot{R}_o^r \dot{\boldsymbol{p}}_c^o + R_o^r \ddot{\boldsymbol{p}}_c^o) \end{aligned} \tag{C.9}$$

simplify

$$\begin{aligned} \ddot{\boldsymbol{p}}_o^u \;=\; & \ddot{\boldsymbol{p}}_r^u + \ddot{\gamma} R_r^u D \boldsymbol{p}_c^r + \dot{\gamma}^2 R_r^u D^2 \boldsymbol{p}_c^r + \dot{\gamma} R_r^u D \dot{\boldsymbol{p}}_c^r + \dot{\gamma} R_r^u D \dot{\boldsymbol{p}}_c^r + R_r^u \ddot{\boldsymbol{p}}_c^r \\ & - \ddot{\eta} R_o^r D \boldsymbol{p}_c^o - \dot{\eta}^2 R_o^r D^2 \boldsymbol{p}_c^o - \dot{\eta} R_o^r D \dot{\boldsymbol{p}}_c^o - \dot{\eta} R_o^r D \dot{\boldsymbol{p}}_c^o - R_o^r \ddot{\boldsymbol{p}}_c^o \end{aligned} \tag{C.10}$$

simplify further

$$\ddot{\boldsymbol{p}}_o^u = \ddot{\boldsymbol{p}}_r^u + \ddot{\gamma} R_r^u D\boldsymbol{p}_c^r + \dot{\gamma}^2 R_r^u D^2 \boldsymbol{p}_c^r + 2\dot{\gamma} R_r^u D\dot{\boldsymbol{p}}_c^r + R_r^u \ddot{\boldsymbol{p}}_c^r - \ddot{\eta} R_o^r D\boldsymbol{p}_c^o - \dot{\eta}^2 R_o^r D^2 \boldsymbol{p}_c^o - 2\dot{\eta} R_o^r D\dot{\boldsymbol{p}}_c^o - R_o^r \ddot{\boldsymbol{p}}_c^o \tag{C.11}$$

If the robot is translating at constant velocity, $\dot{\gamma} = \ddot{\gamma} = \ddot{\boldsymbol{p}}_r^u = 0$

then equation (C.7) and (C.11) simplify to

$$\dot{\boldsymbol{p}}_o^u = \dot{\boldsymbol{p}}_r^u + R_r^u \dot{\boldsymbol{p}}_c^r - \dot{\eta} R_o^r D\boldsymbol{p}_c^o - R_o^r \dot{\boldsymbol{p}}_c^o \tag{C.12}$$

and

$$\ddot{\boldsymbol{p}}_o^u = R_r^u \ddot{\boldsymbol{p}}_c^r - \ddot{\eta} R_o^r D\boldsymbol{p}_c^o - \dot{\eta}^2 R_o^r D^2 \boldsymbol{p}_c^o - 2\dot{\eta} R_o^r D\dot{\boldsymbol{p}}_c^o - R_o^r \ddot{\boldsymbol{p}}_c^o \tag{C.13}$$

respectively

If $\boldsymbol{p}_o^u = [x, y]^T$, $\dot{\boldsymbol{p}}_o^u = [\dot{x}, \dot{y}]^T$, $\ddot{\boldsymbol{p}}_o^u = [\ddot{x}, \ddot{y}]^T$, $\boldsymbol{p}_r^u = [x_r^u, y_r^u]^T$, $\dot{\boldsymbol{p}}_r^u = [\dot{x}_r^u, \dot{y}_r^u]^T$, $\ddot{\boldsymbol{p}}_r^u = [\ddot{x}_r^u, \ddot{y}_r^u]^T$, $\boldsymbol{p}_c^r = [x_c^r, y_c^r]^T$, $\dot{\boldsymbol{p}}_c^r = [\dot{x}_c^r, \dot{y}_c^r]^T$, $\ddot{\boldsymbol{p}}_c^r = [\ddot{x}_c^r, \ddot{y}_c^r]^T$, $\boldsymbol{p}_c^o = [x_c^o, y_c^o]^T$, $\dot{\boldsymbol{p}}_c^o = [\dot{x}_c^o, \dot{y}_c^o]^T$, $\ddot{\boldsymbol{p}}_c^o = [\ddot{x}_c^o, \ddot{y}_c^o]^T$,

Velocity

$$\dot{x} = \dot{x}_r^u + \cos(\gamma)\dot{x}_c^r - \sin(\gamma)\dot{y}_c^r + \dot{\eta}\sin(\eta)x_c^o + \dot{\eta}\cos(\eta)y_c^o - \cos(\eta)\dot{x}_c^o + \sin(\eta)\dot{y}_c^o \tag{C.14}$$

$$\dot{y} = \dot{y}_r^u + \sin(\gamma)\dot{x}_c^r + \cos(\gamma)\dot{y}_c^r - \dot{\eta}\cos(\eta)x_c^o + \dot{\eta}\sin(\eta)y_c^o - \sin(\eta)\dot{x}_c^o - \cos(\eta)\dot{y}_c^o \tag{C.15}$$

If $\gamma = 0$ then $\cos(\gamma) = 1$, $\sin(\gamma) = 0$

$$\dot{x} = \dot{x}_r^u + \dot{x}_c^r + \dot{\eta}(x_c^o \sin(\eta) + y_c^o \cos(\eta)) - \dot{x}_c^o \cos(\eta) + \dot{y}_c^o \sin(\eta) \tag{C.16}$$

$$\dot{y} = \dot{y}_r^u + \dot{y}_c^r + \dot{\eta}(y_c^o \sin(\eta) - x_c^o \cos(\eta)) - \dot{x}_c^o \sin(\eta) - \dot{y}_c^o \cos(\eta) \tag{C.17}$$

If η is small, $\sin(\eta) \approx \eta$ and $\cos(\eta) \approx 0$

$$\dot{x} = \dot{x}_r^u + \dot{x}_c^r + \eta(\dot{\eta}x_c^o + \dot{y}_c^o) \tag{C.18}$$

$$\dot{y} = \dot{y}_r^u + \dot{y}_c^r + \eta(\dot{\eta} y_c^o - \dot{x}_c^o) \tag{C.19}$$

Acceleration

$$\begin{aligned} \ddot{x} &= \cos(\gamma)\ddot{x}_c^r - \sin(\gamma)\ddot{y}_c^r + \ddot{\eta}(x_c^o \sin(\eta) + y_c^o \cos(\eta)) + \dot{\eta}^2(x_c^o \cos(\eta) - y_c^o \sin(\eta)) \\ &\quad + 2\dot{\eta}(\dot{x}_c^o \sin(\eta) + \dot{y}_c^o \cos(\eta)) - \ddot{x}_c^o \cos(\eta) + \ddot{y}_c^o \sin(\eta) \end{aligned} \tag{C.20}$$

$$\begin{aligned} \ddot{y} &= \sin(\gamma)\ddot{x}_c^r + \cos(\gamma)\ddot{y}_c^r + \ddot{\eta}(y_c^o \sin(\eta) - x_c^o \cos(\eta)) + \dot{\eta}^2(x_c^o \sin(\eta) + y_c^o \cos(\eta)) \\ &\quad + 2\dot{\eta}(\dot{y}_c^o \sin(\eta) - \dot{x}_c^o \cos(\eta)) - \ddot{x}_c^o \sin(\eta) + \ddot{y}_c^o \cos(\eta) \end{aligned} \tag{C.21}$$

If $\gamma = 0$

$$\begin{aligned} \ddot{x} &= \ddot{x}_c^r + \ddot{\eta}(x_c^o \sin(\eta) + y_c^o \cos(\eta)) + \dot{\eta}^2(x_c^o \cos(\eta) - y_c^o \sin(\eta)) \\ &\quad + 2\dot{\eta}(\dot{x}_c^o \sin(\eta) + \dot{y}_c^o \cos(\eta)) - \ddot{x}_c^o \cos(\eta) + \ddot{y}_c^o \sin(\eta) \end{aligned} \tag{C.22}$$

$$\begin{aligned} \ddot{y} &= \ddot{y}_c^r + \ddot{\eta}(y_c^o \sin(\eta) - x_c^o \cos(\eta)) + \dot{\eta}^2(x_c^o \sin(\eta) + y_c^o \cos(\eta)) \\ &\quad + 2\dot{\eta}(\dot{y}_c^o \sin(\eta) - \dot{x}_c^o \cos(\eta)) - \ddot{x}_c^o \sin(\eta) + \ddot{y}_c^o \cos(\eta) \end{aligned} \tag{C.23}$$

If η is small, $\sin(\eta) \approx \eta$ and $\cos(\eta) \approx 0$

$$\ddot{x} = \ddot{x}_c^r + \ddot{\eta} x_c^o \eta - \dot{\eta}^2 y_c^o \eta + 2\dot{\eta}\dot{x}_c^o \eta + \ddot{y}_c^o \eta \tag{C.24}$$

$$\ddot{x} = \ddot{x}_c^r + \eta(\ddot{\eta} x_c^o - \dot{\eta}^2 y_c^o + 2\dot{\eta}\dot{x}_c^o + \ddot{y}_c^o) \tag{C.25}$$

$$\ddot{y} = \ddot{y}_c^r + \ddot{\eta} y_c^o \eta + \dot{\eta}^2 x_c^o \eta + 2\dot{\eta}\dot{y}_c^o \eta - \ddot{x}_c^o \eta \tag{C.26}$$

$$\ddot{y} = \ddot{y}_c^r + \eta(\ddot{\eta} y_c^o + \dot{\eta}^2 x_c^o + 2\dot{\eta}\dot{y}_c^o - \ddot{x}_c^o) \tag{C.27}$$

Finding η

$$\tan(\rho) = \frac{y_c^r}{x_c^r} \tag{C.28}$$

If ρ is small then $\tan(\rho) \approx \rho$

$$\rho = \frac{y_c^r}{x_c^r} \tag{C.29}$$

$$\eta = \gamma + \rho \frac{\|\boldsymbol{p}_c^r\|}{\|\boldsymbol{p}_c^o\|} \tag{C.30}$$

$$\eta = \gamma + \frac{y_c^r}{x_c^r} \sqrt{\frac{x_c^{r2} + y_c^{r2}}{x_c^{o2} + y_c^{o2}}} \tag{C.31}$$

If $\gamma = 0$

$$\eta = \frac{y_c^r}{x_c^r} \sqrt{\frac{x_c^{r2} + y_c^{r2}}{x_c^{o2} + y_c^{o2}}} \tag{C.32}$$

If $\sqrt{\frac{x_c^{r2} + y_c^{r2}}{x_c^{o2} + y_c^{o2}}} = G$ or "gear ratio"

$$\eta = \frac{y_c^r}{x_c^r} G \tag{C.33}$$

Angular Velocity

$$\dot{\eta} = \frac{x_c^r \dot{y}_c^r - \dot{x}_c^r y_c^r}{x_c^{r2}} G \tag{C.34}$$

Angular Acceleration

$$\ddot{\eta} = \frac{\ddot{y}_c^r x_c^{r2} - 2\dot{y}_c^r x_c^r \dot{x}_c^r + y_c^r (2\dot{x}_c^{r2} - x_c^r \ddot{x}_c^r)}{x_c^{r3}} G \tag{C.35}$$

Hence, equations of motion

$$\begin{array}{rcl} \dot{x} & = & \dot{x}_r^u + \dot{x}_c^r + \eta(\dot{\eta} x_c^o + \dot{y}_c^o) \\ \dot{y} & = & \dot{y}_r^u + \dot{y}_c^r + \eta(\dot{\eta} y_c^o - \dot{x}_c^o) \\ \dot{\eta} & = & \frac{x_c^r \dot{y}_c^r - \dot{x}_c^r y_c^r}{x_c^{r2}} G \\ \ddot{x} & = & \ddot{x}_c^r + \eta(\ddot{\eta} x_c^o - \dot{\eta}^2 y_c^o + 2\dot{\eta}\dot{x}_c^o + \ddot{y}_c^o) \\ \ddot{y} & = & \ddot{y}_c^r + \eta(\ddot{\eta} y_c^o + \dot{\eta}^2 x_c^o + 2\dot{\eta}\dot{y}_c^o - \ddot{x}_c^o) \\ \ddot{\eta} & = & \frac{\ddot{y}_c^r x_c^{r2} - 2\dot{y}_c^r x_c^r \dot{x}_c^r + y_c^r (2\dot{x}_c^{r2} - x_c^r \ddot{x}_c^r)}{x_c^{r3}} G \end{array} \tag{C.36}$$

Bibliography

[1] P. Agarwal, J.-C. Latombe, R. Motwani, and P. Raghavan. Nonholonomic path planning for pushing a disk among obstacles. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 4, pages 3124 –3129 vol.4, apr 1997.

[2] S. Akella. *Robotic manipulation for parts transfer and orienting: mechanics, planning, and shape uncertainty*. PhD thesis, Pittsburgh, PA, USA, 1996. Adviser-Mason, Matthew T.

[3] S. Akella and M. Mason. Parts orienting by push-aligning. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 1, pages 414 –420 vol.1, 21-27 1995.

[4] S. Akella and M. T. Mason. Posing polygonal objects in the plane by pushing. *The International Journal of Robotics Research*, 17(1):70–88, 1998.

[5] J. Barraquand and J. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10:121–155, 1993. 10.1007/BF01891837.

[6] R. e. a. Brockett. *Asymptotic stability and feedback stabilization*. Defense Technical Information Center, 1983.

[7] M. Brokowski, M. Peshkin, and K. Goldberg. Curved fences for part alignment. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 467 –473 vol.3, may 1993.

[8] T. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.

[9] A. Cosgun, T. Hermans, V. Emeli, and M. Stilman. Push planning for object placement on cluttered table surfaces. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4627 –4632, sept. 2011.

[10] A. De Luca, R. Mattone, and G. Oriolo. Control of underactuated mechanical systems: application to the planar 2r robot. In *Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 2, pages 1455 –1460 vol.2, dec 1996. top result

in google scholar for "Robotic Manipulation by Pushing using a Single Point at Constant Velocity: Modelling and Techniques".

[11] A. De Luca and G. Oriolo. Nonholonomy in redundant robots under kinematic inversion. In *Proc. 4th IFAC Symp. on Robot Control*, pages 179–184, 1994.

[12] M. W. M. G. Dissanayake, C. J. Goh, and N. Phan-Thien. Time-optimal trajectories for robot manipulators. *Robotica*, 9(02):131–138, 1991.

[13] M. Dogar and S. Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2123 –2130, oct. 2010.

[14] M. Dogar and S. Srinivasa. A framework for push-grasping in clutter. In *in Robotics: Science and Systems, Los Angeles, CA, USA, June 2011*, 2011.

[15] M. Dogar and S. Srinivasa. A planning framework for non-prehensile manipulation under clutter and uncertainty. *Autonomous Robots*, 33:217–236, 2012. 10.1007/s10514-012-9306-z.

[16] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):pp. 497–516, 1957.

[17] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *IEEE ROBOTICS AND AUTOMATION MAGAZINE*, 13(2):99–108, 2006.

[18] J. Esposito. Distributed grasp synthesis for swarm manipulation with applications to autonomous tugboats. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1489 –1494, may 2008.

[19] C. J. Goh, N. J. Edwards, and A. Y. Zomaya. Feedback control of minimum-time optimal control problems using neural networks. *Optimal Control Applications and Methods*, 14(1):1–16, 1993.

[20] K. Harada, J. Nishiyama, Y. Murakami, and M. Kaneko. Pushing multiple objects using equivalent friction center. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3, pages 2485 –2491, 2002.

[21] T. Igarashi, Y. Kamiyama, and M. Inami. A dipole field for object delivery by pushing on a flat surface. In *2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA, May 2010.

[22] J. Kenney, T. Buckley, and O. Brock. Interactive segmentation for manipulation in unstructured environments. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1377 –1382, may 2009.

[23] K. R. S. Kodagoda, W. Wijesoma, and A. Balasuriya. Road curb and intersection detection using a 2d lms. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 19–24 vol.1, 2002.

[24] M. Kurisu and T. Yoshikawa. Trajectory planning for an object in pushing operation. In *Japan-U.S.A. Symposium on Flexible Automation*, pages 1009–1016, Kobe, Japan, 1994.

[25] M. Kurisu and T. Yoshikawa. Tracking control for an object in pushing operation. *Journal of Robotic Systems*, 14(10):729–739, 1997.

[26] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the NelderMead simplex method in low dimensions. *SIAM Journal of Optimization*, 9:112–147, 1998.

[27] K. Lee, M. Dissanayake, J. Gal, and G. Fang. Gait synthesis for a hybrid legged-wheeled mobile vehicle. In *PROCEEDINGS OF THE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ROBOTS*, volume 26, pages 417–422. INTERNATIONAL FEDERATION OF ROBOTICS, & ROBOTIC INDUSTRIES, 1995.

[28] K. Y. Lee and M. W. M. G. Dissanayake. Near minimum-time trajectory for two coordinated manipulators. *Robotica*, 13(02):177–184, 1995.

[29] Q. Li. *Coordinated nonprehensile manipulation for parts transfer: Mechanics, control, and planning.* PhD thesis, School of Engineering Science, Simon Fraser University, 2006.

[30] K. Lynch. The mechanics of fine manipulation by pushing. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 2269 –2276 vol.3, may 1992.

[31] K. Lynch. Locally controllable manipulation by stable pushing. *Robotics and Automation, IEEE Transactions on*, 15(2):318 –327, apr 1999.

[32] K. Lynch and M. T. Mason. Controllability of pushing. *IEEE International Conference on Robotics and Automation*, 1:112–119, 1995.

[33] K. M. Lynch. *Nonprehensile robotic manipulation: controllability and planning.* PhD thesis, Pittsburgh, PA, USA, 1996.

[34] K. M. Lynch. Issues in nonprehensile manipulation. In *Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective: the algorithmic perspective*, WAFR '98, pages 237–250, Natick, MA, USA, 1998. A. K. Peters, Ltd.

[35] K. M. Lynch and M. T. Mason. Pulling by pushing, slip with infinite friction, and perfectly rough surfaces. *The International Journal of Robotics Research*, 14(2):174–183, 1995.

[36] K. M. Lynch and M. T. Mason. Stable pushing: mechanics, controllability, and planning. *Int. J. Rob. Res.*, 15(6):533–556, 1996.

[37] K. M. Lynch and M. T. Mason. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *The International Journal of Robotics Research*, 18(1):64–92, 1999.

[38] K. M. Lynch and T. D. Murphey. Control of nonprehensile manipulation. In *Control Problems in Robotics*, volume 4 of *Springer Tracts in Advanced Robotics*, pages 39–58. Springer Berlin / Heidelberg, 2003.

[39] M. T. Mason. Mechanics and planning of manipulator pushing operations. *Int. J. Rob. Res.*, 5(3):53–71, 1986.

[40] M. T. Mason. Progress in nonprehensile manipulation. *The International Journal of Robotics Research*, 18(11):1129–1141, 1999.

[41] M. T. Mason. *Mechanics of robotic manipulation*. MIT Press, Cambridge, MA, USA, 2001.

[42] M. Mataric, M. Nilsson, and K. Simsarin. Cooperative multi-robot box-pushing. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 556 –561 vol.3, aug 1995.

[43] K. Matheus and A. M. Dollar. Benchmarking grasping and manipulation: Properties of the objects of daily living. In *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, pages 5020–5027, 2010.

[44] S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. Controlling the planar motion of a heavy object by pushing with a humanoid robot using dual-arm force control. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1428–1435, 2012.

[45] Y. Okawa, K. Taguchi, and K. Yokoyama. Control of mobile robots for the push-a-box operation. In *Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON '93., International Conference on*, pages 1472 –1477 vol.3, nov 1993.

[46] Y. Okawa and K. Yokoyama. Control of a mobile robot for the push-a-box operation. In *Robotics and Automation, 1992. Proceedings of the 1992 IEEE International Conference on*, 1992.

[47] M. Peshkin and A. Sanderson. The motion of a pushed, sliding workpiece. *Robotics and Automation, IEEE Journal of*, 4(6):569–598, 1988.

[48] L. Qingguo and S. Payandeh. Manipulation of convex objects via two-agent point-contact push. *The International Journal of Robotics Research*, 26(4):377–403, 2007.

[49] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[50] A. Schmid, R. Yechangunja, S. Thalhammer, and M. Srinivasan. Human-operated 3d micro-manipulator with haptic feedback. In *Haptics Symposium (HAPTICS), 2012 IEEE*, pages 517 –522, march 2012.

[51] S. Walker and J. Salisbury. Pushing using learned manipulation maps. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3808 –3813, may 2008.

[52] Y. Wang and C. de Silva. Multi-robot box-pushing: Single-agent q-learning vs. team q-learning. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3694 –3699, oct. 2006.

[53] T. Yoshikawa and M. Kurisu. Indentification of the center of friction from pushing an object by a mobile robot. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, pages 449–454 vol.2, 1991.