

Copyright

by

Subramanian Ramamoorthy

2007

The Dissertation Committee for Subramanian Ramamoorthy  
certifies that this is the approved version of the following dissertation:

**Task Encoding, Motion Planning and Intelligent Control  
using Qualitative Models**

Committee:

---

Benjamin J. Kuipers, Supervisor

---

Aristotle Arapostathis

---

Mircea D. Driga

---

Joydeep Ghosh

---

Edward Marcotte

**Task Encoding, Motion Planning and Intelligent Control  
using Qualitative Models**

by

**Subramanian Ramamoorthy, B.E., M.E.**

**Dissertation**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Doctor of Philosophy**

**The University of Texas at Austin**

August 2007

Dedicated to  
the spirit of free inquiry

# Acknowledgments

A wise man<sup>1</sup> said “ *If you have built castles in the air, your work need not be lost; that is where they should be. Now put the foundations under them.*” At times, graduate school has felt like such a process of securing a lofty goal that was set before fully realizing what it entailed. It is now my pleasure to recognize those who have helped me reach this goal.

As I step forth to receive my doctorate, my first and foremost thanks go to my research advisor, Ben Kuipers. I thank him for technical advice, inspiring conversation and honest criticism. But most of all, I thank him for introducing me to the idea that science is a long, slow conversation, where it is the original and interesting viewpoint that matters.

I thank Peter Stone for introducing me to his pragmatic perspective on research. It was for a term project in his class, unrelated to my then research focus, that I first formulated a key idea in this dissertation. I thank Ari Arapostathis for teaching the stochastic control class from an original point of view. It emboldened me to look outside the box. I thank Rafael de la Llave for being such a welcoming math teacher. Although I entered his classes as an outsider, I left with a sense of ownership. I thank Steve LaValle for encouragement and useful feedback. I thank members of my thesis committee for helpful comments.

I thank Lothar Wenzel for being a friend, mentor, teacher and collaborator. I often learned more from him in hour-long whiteboard discussions than I did in many formal classes on the same topics. I hope to one day emulate his rare ability to simplify practical problems using abstract concepts. I thank Greg Morrow for asking insightful questions at

---

<sup>1</sup>H.D. Thoreau, *Walden*, Yale University Press, 2004.

just the right time. I thank Ram Rajagopal for many frank conversations on research, life and the human condition.

I thank my Robot Lab colleagues - Joseph, Patrick, Aniket, Matt, Jeff, Shilpa, Jonathan, Mohan and other occasional inhabitants - for interesting discussion on all things ranging from AI to politics and the politics of AI. Joseph deserves special mention for many topical discussions on learning and robotics that helped shape my research agenda.

Joseph Hays went well beyond the call of duty to help me pursue my doctoral education while working at National instruments. I thank him for standing by me. I also thank other NI colleagues for their support - Dinesh Nair, Jeannie Falcon, Ravi Marawar and the “analysis family”.

I would not have embarked on the winding roads of graduate school were it not for the unfailing support of my parents. I thank Dad for his guidance and his courage, which he taught by example. Even as he was dying of cancer, he asked us not to mourn but to go forth and do something with our lives. I wish he were here to witness the fruits of his labor. I thank Mom for her patient love. I thank my brother, Harish, for always playing the devil’s advocate - if nothing else, it helped me learn to argue better.

I thank little Syona for taking me along on those all too short walks around the neighborhood, and for stopping to smell the flowers - watching her find sheer joy in the little things has made many a day worthwhile. Lastly, but very importantly, I thank Vivin for love, patience and understanding. I have not said it often enough, but none of this would be possible without her support.

SUBRAMANIAN RAMAMOORTHY

*The University of Texas at Austin*  
*August 2007*

# **Task Encoding, Motion Planning and Intelligent Control using Qualitative Models**

Publication No. \_\_\_\_\_

Subramanian Ramamoorthy, Ph.D.

The University of Texas at Austin, 2007

Supervisor: Benjamin J. Kuipers

This dissertation addresses the problem of trajectory generation for dynamical robots operating in unstructured environments in the absence of detailed models of the dynamics of the environment or of the robot itself. We factor this problem into the subproblem of task variation, and the subproblem of imprecision in models of dynamics.

The problem of task variation is handled by defining task level control strategies in terms of qualitative models that support structurally stable phase space trajectories. Such models define low-dimensional spaces within which it is possible to select trajectories that constitute plans for achieving a desired goal.

The second problem, that of model imprecision, arises when embedding the resulting trajectories in the phase space of the more complex higher-dimensional system that actually performs the task of interest. Trajectories in the high-dimensional phase space that are compatible with the low-dimensional plan are restricted to lie on a manifold. In the absence of analytical models of the high-dimensional dynamics, this manifold may be approximated using observed data generated by a randomized exploration of the state space. Approximations driven by such an imperfect set of observations can lead to spurious trajectories, but this problem is solved by regularizing the approximation using the low-dimensional model.

This methodology is developed through a sequence of design problems. First, basic notions regarding control with qualitative models are clarified through the design of a global controller for the inverted pendulum and cart-pole systems. This is followed by the more challenging problem of dynamic bipedal walking on irregular terrain, which is the primary motivating problem for this dissertation. Our solution to the dynamic walking problem advances the state of the art by simultaneously achieving several important properties. Our algorithm generates trajectories to walk on irregular terrain, with only a qualitative model of the dynamics of the robot, and with energy usage comparable with actuated walkers utilizing passive dynamic principles.

Although the definition of tasks in terms of structurally stable orbits and manifolds is very natural when talking about physical systems, this representation yields benefits in more artificial domains as well. This is demonstrated through the example of spatiotemporal control of polygonal shapes, such as in a robot collective.

Parts of the work described in this dissertation have been previously presented in the following articles:

1. B.J. Kuipers, S. Ramamoorthy, Qualitative modeling and heterogeneous control of global system behavior. In C. J. Tomlin and M. R. Greenstreet (Eds.), *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, Volume 2289,



pp 294-307, Springer Verlag, 2002.

2. S. Ramamoorthy, B.J. Kuipers, Qualitative heterogeneous control of higher order systems. In O. Maler and A. Pnueli (Eds.), *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, Volume 2623, pp 417-434, Springer Verlag, 2003.
3. S. Ramamoorthy, B.J. Kuipers, Controller synthesis using qualitative models and constraints. In J. de Kleer and K. Forbus (Eds.), *Proc. 18th International Workshop on Qualitative Reasoning*, pp 41-50, August 2-4, 2004.
4. S. Ramamoorthy, B.J. Kuipers, Qualitative hybrid control of dynamic bipedal walking, In G. S. Sukhatme, S. Schaal, W. Burgard and D. Fox (Eds.), *Robotics : Science and Systems II*, MIT Press, 2007.
5. S. Ramamoorthy, B.J. Kuipers, L. Wenzel, Parametrization and computations in shape spaces with area and boundary invariants, In *Proc. 16th Fall Workshop on Computational and Combinatorial Geometry*, Nov. 10-11, 2006.
6. S. Ramamoorthy, R. Rajagopal, Q. Ruan, L. Wenzel, Low-discrepancy curves and efficient coverage of space. In S. Akella, N. Amato, W. Huang, B. Mishra (Eds.), *Algorithmic Foundations of Robotics VII*, Springer Tracts in Advanced Robotics, Springer Verlag, 2007.

# Contents

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| <b>Acknowledgments</b>                                                         | <b>v</b>   |
| <b>Abstract</b>                                                                | <b>vii</b> |
| <b>Chapter 1 Introduction</b>                                                  | <b>1</b>   |
| 1.1 Statement of the Problem . . . . .                                         | 2          |
| 1.2 Thesis Overview . . . . .                                                  | 2          |
| 1.3 Major Contributions . . . . .                                              | 6          |
| <b>Chapter 2 Background: Planning and Control</b>                              | <b>7</b>   |
| 2.1 State Space Search and Symbolic Planning . . . . .                         | 8          |
| 2.2 Qualitative Reasoning . . . . .                                            | 9          |
| 2.3 Motion Planning and Nonlinear Trajectory Generation . . . . .              | 10         |
| 2.4 Reinforcement Learning . . . . .                                           | 12         |
| 2.5 Learning by Imitation, Lazy Learning and<br>Manifold Learning . . . . .    | 13         |
| 2.6 Control Theory . . . . .                                                   | 14         |
| 2.7 Control in Biology and Biomimetic Robotics . . . . .                       | 16         |
| <b>Chapter 3 Task Encoding Using Qualitative Models</b>                        | <b>18</b>  |
| 3.1 Multiple Model Approaches to Nonlinear Modeling and<br>Control . . . . .   | 20         |
| 3.2 Qualitative Dynamics and a Heterogeneous Control<br>Problem . . . . .      | 21         |
| 3.3 Some Results on the Qualitative Dynamics of Second Order Systems . . . . . | 24         |
| 3.4 A Global Control Strategy for the Pendulum . . . . .                       | 33         |

|                  |                                                                                      |            |
|------------------|--------------------------------------------------------------------------------------|------------|
| 3.4.1            | Stabilizing the Pendulum: Balance Mode . . . . .                                     | 34         |
| 3.4.2            | Escaping the Elliptic Fixed Point: Pump Mode . . . . .                               | 36         |
| 3.4.3            | Handling High Energy Rotations: Spin Mode . . . . .                                  | 36         |
| 3.4.4            | Ensuring Correct Mode Transitions: Separatrix as a Sliding Mode . . . . .            | 37         |
| 3.5              | A Global Control Strategy for the Cart-Pole System . . . . .                         | 41         |
| 3.5.1            | Using Time-Scale Separation to Deal with Underactuation . . . . .                    | 42         |
| 3.5.2            | Handling Nonsmooth Nonlinearities . . . . .                                          | 49         |
| 3.5.3            | Experiments with a Physical Cart-Pole System . . . . .                               | 51         |
| 3.6              | Synthesizing Nonlinear Controllers in a Qualitatively<br>Constrained Space . . . . . | 52         |
| 3.7              | Discussion: What is Hard about Task Encoding? . . . . .                              | 55         |
| <b>Chapter 4</b> | <b>Bipedal Walking: Motion Planning for a Template Model</b>                         | <b>59</b>  |
| 4.1              | Introduction to the Dynamic Bipedal Walking Problem . . . . .                        | 59         |
| 4.2              | Some Important Ideas in Dynamic Bipedal Walking . . . . .                            | 60         |
| 4.3              | Observations Regarding Biological and Biomimetic Robotic Walking . . . . .           | 65         |
| 4.4              | Outline of a Strategy for Dynamic Bipedal Walking . . . . .                          | 69         |
| 4.5              | Motion Planning with the Template Model . . . . .                                    | 74         |
| 4.5.1            | Template Model for Dynamic Bipedal Walking . . . . .                                 | 74         |
| 4.5.2            | A ballistic parametrization for irregular terrain gaits . . . . .                    | 77         |
| 4.5.3            | An algorithm for intermittently actuated passive walking . . . . .                   | 81         |
| 4.5.4            | Energy control within the template model . . . . .                                   | 86         |
| 4.5.5            | Simulation results . . . . .                                                         | 86         |
| <b>Chapter 5</b> | <b>Bipedal Walking: Learning the Template-Anchor Embedding</b>                       | <b>92</b>  |
| 5.1              | Relationship between Template and Anchor Models . . . . .                            | 92         |
| 5.2              | Algorithm for Data-driven Approximation of Task<br>Manifolds . . . . .               | 95         |
| 5.3              | Experiments with a Simulated Multi-link Legged Robot . . . . .                       | 100        |
| <b>Chapter 6</b> | <b>Task Encoding in Abstract Spaces</b>                                              | <b>110</b> |
| 6.1              | Introduction . . . . .                                                               | 110        |
| 6.1.1            | Abstractions and the process of functional biomimesis . . . . .                      | 110        |
| 6.1.2            | Some useful notions from differential geometry . . . . .                             | 113        |
| 6.1.3            | Why are dynamical tasks manifold structured? . . . . .                               | 121        |

|                                                           |                                                                   |            |
|-----------------------------------------------------------|-------------------------------------------------------------------|------------|
| 6.2                                                       | Encoding the Constrained Evolution of a Polygonal Shape . . . . . | 124        |
| 6.3                                                       | Approximation of Geodesics on Task Manifolds . . . . .            | 129        |
| 6.3.1                                                     | Well-distributed point sets . . . . .                             | 130        |
| 6.3.2                                                     | Algorithm for manifold sampling-based approximation of geodesics  | 132        |
| 6.4                                                       | Well-distributed Curves on Riemannian Manifolds . . . . .         | 135        |
| 6.4.1                                                     | Low-discrepancy curves in the unit square . . . . .               | 135        |
| 6.4.2                                                     | Fair mappings between Euclidean and Riemannian spaces . . . . .   | 141        |
| 6.4.3                                                     | Examples of the coverage of various surfaces . . . . .            | 144        |
| 6.4.4                                                     | Discussion: On the definition of fair diffeomorphisms . . . . .   | 148        |
| <b>Chapter 7 Conclusions and Prospects for the Future</b> |                                                                   | <b>150</b> |
| 7.1                                                       | Key Ideas . . . . .                                               | 150        |
| 7.2                                                       | Future Work and Open Questions . . . . .                          | 151        |
| 7.2.1                                                     | Scaling up to physical bipeds . . . . .                           | 151        |
| 7.2.2                                                     | Understanding the proper role of template models . . . . .        | 152        |
| 7.2.3                                                     | Towards scientific discovery . . . . .                            | 153        |
| <b>Bibliography</b>                                       |                                                                   | <b>155</b> |
| <b>Vita</b>                                               |                                                                   | <b>166</b> |

# Chapter 1

## Introduction

The promise of robotics and autonomous agents derives from the hope that one day we will be able to construct machines that work alongside humans to assist with tedious, or even dangerous, tasks - ranging from rescue to interplanetary exploration. Engineers are still far from being able to fully realize this promise. There are numerous reasons for this, covering the gamut of engineering technologies - from energy storage to mechanical design, but there is one that stands out as being particularly challenging and unique to the area of robotics. This is our inability to efficiently specify and reliably execute movement tasks outside of highly scripted and structured environments, in the face of significant imprecision in models of dynamics and the environment [1, 2].

Quite apart from the needs of robotics, the problem of movement generation is of great scientific interest in many communities, ranging from neuroscience to kinesiology. As observed by Wolpert et al. [3],

*From the motor chauvinist's point of view the entire purpose of the human brain is to produce movement. Movement is the only way we have of interacting with the world. All communication, including speech, sign language, gestures and writing, is mediated via the motor system. All sensory and cognitive processes may be viewed as inputs that determine future motor outputs.*

Animals handle such an incredible variety of tasks, seemingly effortlessly, through the use of a few common planning and control strategies [4, 5]. Can we learn from and replicate this spectacular success? Can we create a consistent framework of representations and algorithms that would enable us to build similarly efficient and robust robots?

## 1.1 Statement of the Problem

Concretely, the problem of movement generation is one of *selecting suitable trajectories for each of the degrees of freedom of a robot, a high-dimensional nonlinear system, so as to reliably and predictably achieve spatial and temporal goals in an external world.*

The key technical challenge is to find ways to *encode and achieve global behaviors* in such complex systems subject to the following constraints:

- *Imprecise knowledge of the environment*, e.g., in a locomotion task, a robot can only expect to have reliable information within a small neighborhood in time and space. The movement generation task involves appropriately reacting to the environment as new information is acquired and corresponding goals are formulated. This necessitates algorithms that can act on-line in order to handle not just small disturbances such as due to sensor noise but also much more severe large deviations that require a global approach to recovery and task level control.
- *Imprecise knowledge of dynamics and constraints*, models of which must be acquired and improved upon on-line, from noisy and imperfect experience. This constraint is driven by the observation that the paradigm of carefully modeling a system and restricting the structure of the task and the environment, that is so common in control engineering practice, is unrealistic for an autonomous agent that must operate on its own in novel environments. In particular, many robotic tasks are the result of a coupled interaction between a high-dimensional nonlinear system and an unstructured environment. It is difficult to model such nonlinear interactions and task-specific constraints well enough to be able to utilize typical model based design techniques.

The goal we set for ourselves is the development of a consistent framework of representations and algorithms that can enable both the achievement of such complex movement tasks and the analysis of important dynamical properties.

## 1.2 Thesis Overview

At the highest level, there are two related issues that characterize our problem: complexity and imprecise knowledge.

The problem of complexity is addressed first, decomposing the movement generation problem into one of encoding the desired task in terms of simple abstractions and

then of lifting these tasks to the much more complex system that is really of interest. The simple abstraction will serve as an efficient way to encode reaction to environmental and task variations while imprecision about knowledge of the high-dimensional dynamics will be handled by data-driven learning.

This approach is inspired by biological theories [4, 6] that suggest that nature handles complexity in a similarly hierarchical manner. Moreover, in many cases [4], the simpler abstractions seem to be based on clever use of the *natural dynamics* of the system, with minimal active modifications and forces.

The efficiency in animal locomotion is achieved through a combination of morphological adaptations and fine-tuned control strategies, both resulting from the very complex satisficing process of evolution. For the purposes of the current dissertation, the emphasis is on replicating the planning and control strategies, *given* a typical robotic mechanism. This is the approach of *functional* biomimesis.

In this vein, this dissertation is a combination of empirical and theoretical arguments. The work is empirical to the extent that the thesis is developed through a sequence of specific design problems. However, the solution to each design problem involves principled analysis and sound algorithms.

Chapter 3 explores the question of what it means to utilize *qualitative dynamics* to encode tasks? One of the major benefits of thinking in terms of qualitative models that support structurally stable dynamical behaviors is the resulting elegant formulation of *global* control tasks in a hybrid systems or multiple model [7] setting. It is well known that the behavior of a nonlinear system can be understood in terms of topologically defined families of phase space trajectories. If one were to specify the global *task* as a suitable composition of these local trajectories then it becomes possible to obtain a corresponding family of global behaviors. Given a family of such behaviors, a new goal in a changing environment is handled by appropriate selection of a suitable member of the family, with the assurance that there exists a behavior capable of achieving the task.

The application of this general idea is demonstrated using a benchmark problem that is famous in the nonlinear control and machine learning communities - the inverted pendulum or cart-pole problem. Although the simple local control formulation of stabilizing the inverted pendulum is considered a solved problem, the problem of ensuring global stability in the face of arbitrary large disturbances and spatial constraints is nontrivial and is of continuing interest in the research literature. In chapter 3, this problem is solved and results from numerical simulation and physical experiments are presented.

This sets the stage for a much more challenging design problem - that of legged locomotion on irregular terrain. There is a very long history of research on planning and control strategies for legged robots. Literally, every major planning and control technique has been applied to the problem of legged locomotion. Even so, there are some design goals that have remained elusive.

An important open problem is to combine two formerly dichotomous approaches to the design of bipedal robot gaits - ballistic walking and actively controlled walking. It has been shown [8, 9] that robots can be designed to walk down simple and uniform slopes with absolutely no actuation or run-time computation, powered only by gravity and regulated by mechanical design. This mirrors observations of similar passive regimes in human gait [10], [11]. However, a passive robot is not capable of handling irregular terrain or reacting to a changing environment - which requires active planning and control. Even in the situations where passive walking works, the gait is often fragile. For this reason, and due to the general difficulty of constructing reliable passive robots of the multi-link legged form, there is great interest in understanding how the passive and active paradigms can be combined [12].

This problem is solved in two stages. Firstly, chapter 4 considers the problem of intermittently actuating the simplest two-legged robot, similar to the original McGeer walker [9] and embodying the principle of compass gait pendular walking. An algorithm for intermittently actuated passive walking is presented - one that allows the robot to move ballistically, unactuated, for much of the walking cycle and requiring only brief pulses of actuation to influence the dynamics towards planned footfall goals. Moreover, it is shown that the properties of such a gait can be analyzed in a principled way and its limits of applicability can be quantitatively established.

This intermittent-actuation scheme can be implemented on a suitable physical instantiation to augment passive walking. However, its real significance is that it constitutes an excellent template in the spirit of Full and Koditschek [6]. So, the next step is to investigate how we can lift this behavior to a multi-link legged robot. As stated in the previous section, this needs to be achieved in the absence of precise models of high-dimensional dynamics. In chapter 5, this is achieved through machine learning. Much of the template gait is entirely ballistic, mirroring similar dynamics in human gait. What this really means is that in between points of discontinuities at footfall and subsequent launch into the next step, the dynamics is akin to that of a conservative system evolving along suitable energy level curves. This suggests that the ideal encoding of the task is in terms of a manifold. This is a natural extension of the approach presented in chapter 3, where the separatrix - a 1-dim



manifold, is used in the organization of a global control strategy. Then, the learning problem becomes one of non-parametrically approximating the task manifold given traces of observations of the robot's high dimensional dynamics and knowledge of the template gait from chapter 4. This problem is solved through the use of a novel manifold approximation algorithm.

Much of the above argument rests on the identification and utilization of qualitative structures in phase space. In particular, the reader may have discerned the primary role played by pendulum systems in the above arguments. However, the proposed methodology is not restricted to pendular systems, or even just to physical systems.

The notion of utilizing the natural dynamics and organizing global behaviors in terms of manifold structures has a sound basis in the elegant mathematical machinery of the geometric approach to dynamical systems. For physical systems, this connection has been fully developed [13, 14]. However, there are numerous other domains that also utilize this representation - including problems involving shape spaces [15], the space of probability distributions [16], etc.

However, in *artificial* or abstract spaces, not having easy recourse to well known conservation laws and physical principles that yield qualitative models, how does one reason about *dynamics*? In fact, where do the dynamics come from? A natural way to encode tasks in such spaces is to represent the space as a manifold and feasible trajectories as geodesics. This leads to constructive algorithms that can work with limited information such as the specification of the configuration manifold and task constraints.

Chapter 6 explores this direction, using a simple but nontrivial problem involving the controlled evolution of shape. Shape plays a crucial role in robotics, e.g., formation control. Problems involving shape have also been the focus of a successful collaboration between roboticists and computational biologists, e.g., the study of protein behaviors [17, 18].

An important problem involving shape is that of motion planning for robot collectives. As before, decomposing the problem using template and anchor models leads to more tractable solutions. The template model is a shape manifold that encodes the constraints of the task, such as constant enclosed area or constant perimeter. Then, through the use of efficient algorithms for sampling on such manifolds, and subsequent approximation of geodesics, it becomes possible to plan and compute with this artificial dynamical model.

### 1.3 Major Contributions

1. Hybrid control: A novel approach to the principled design of global behaviors for a nonlinear system, which is used to solve the global version of a famous benchmark problem involving the inverted pendulum and cart-pole systems.
2. Robotics (legged locomotion): An encoding of the task of walking on irregular terrain, for a compass-gait template model, that is used to develop a novel algorithm for intermittently actuated passive dynamic walking.
3. Robotics (legged locomotion): Definition of a scheme by which the gait synthesized in step 2 can be lifted to a robot with more complex morphology, i.e., multi-link legs.
4. Machine learning: Algorithm for non-parametrically approximating task manifolds, using data gathered from experience and a template based low-dimensional definition of tasks.
5. Motion planning in abstract spaces: An encoding of the task of constrained shape evolution in terms of a Riemannian manifold. This, in conjunction with efficient algorithms for incremental sampling on nonlinear manifolds, leads to an efficient solution to the on-line motion planning problem.

## Chapter 2

# Background: Planning and Control

Many problems in robotics are commonly addressed by hierarchically composing modules that handle deliberative reasoning, task level control and lower level feedback stabilization or trajectory following. For instance, a humanoid robot that is out on a rescue mission must first deliberate on high level goals, such as whether it needs to return to its base to recharge batteries or whether it should cross the river and look for victims who need help. Having decided to cross the river, it must identify reliable footholds, e.g., a sequence of stepping stones on the river. Next, it must synthesize a concrete plan, in terms of its degrees of freedom, that will enable it to physically achieve these foothold targets. This is the trajectory generation problem, the interface between the higher level logical deliberations and the continuous physical world. The lowest level task is then to execute this planned trajectory through feedback stabilization and trajectory following control laws. This hierarchical scheme is depicted in figure 2.1.

In practice, there are many good reasons for striving towards a separation of concerns between levels in this hierarchy. This leads to more efficient system design [19]. From the algorithmic point of view, what this hierarchy implies is the need for a consistent framework in the sense that the choice to a technique to solve one of the constituent problems should not adversely affect or unduly constrain one's ability to efficiently solve the remaining problems.

The focus of this dissertation is primarily on the middle level, the task-level control or trajectory generation problem. With this in mind, we seek an understanding of how best to abstract continuous dynamics so that there is a suitably small set of symbolic states useful for higher-level deliberative reasoning, and also how the complexities of global dynamic

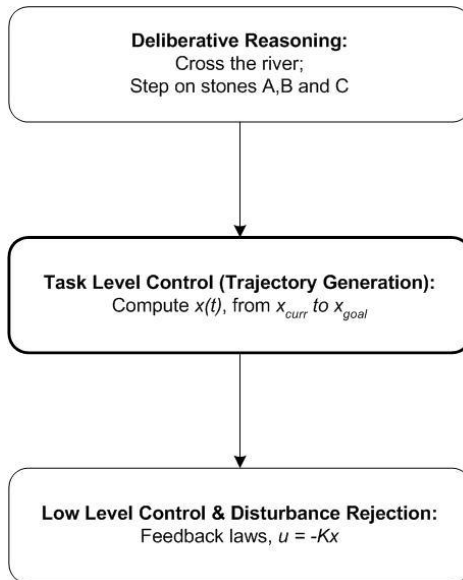


Figure 2.1: Canonical hierarchical structure of planning and control problems. The primary focus of this dissertation is on the middle block - task level control.

behavior can be handled within the process of trajectory generation - enabling the low level controllers to be locally defined and simple to implement.

This chapter is a brief survey of relevant concepts from the planning and control literature, covering each of the major steps in the above hierarchy but emphasizing recent advances that impact the task-level control problem.

## 2.1 State Space Search and Symbolic Planning

Much of the early work in artificial intelligence was focussed on the problem of synthesizing sequences of actions in order to achieve specific goals. A canonical approach to this problem [20] involves encoding the problem in terms of states and operators or primitive actions. Each operator is precisely specified in terms of pre- and post-conditions so that the process of going from an initial to a final state involves reasoning about sequencing such operators. A feasible sequence would lead from the initial to the final states, satisfying the constraints of every operator along the way. The focus of research in this area is on efficient representations for state spaces and on algorithmic techniques for eliminating

spurious plans, often aided by specialized heuristics for each problem. This approach has resulted in many memorable successes, including grandmaster-level chess and backgammon programs running on typical desktop computers. However, direct application of such search techniques to continuous systems has been, by and large, unsuccessful.

## 2.2 Qualitative Reasoning

The problem of reconciling the search and constraint satisfaction approach to AI with the demands of a continuous physical world led to research into qualitative simulation and reasoning [21]. One of the major contributions of this research was the definition of a language for abstracting continuous dynamics in terms of a finite collection of landmarks, qualitative states and directions. With this carefully defined finite collection, it becomes possible to invoke the existing machinery of state space search and constraint satisfaction in order to automatically reason about the physical world [22, 23, 24, 25, 21]. In fact, this procedure can be followed in a formal provably correct manner, as shown by Shults [26].

A particularly interesting trend in this research was the use of ideas from nonlinear dynamics, specifically the use of families of phase space trajectories, as the basis for abstracting the continuous world. This approach was used successfully by Yip in the design of the KAM program [27] that could autonomously search for interesting dynamic behaviors by appropriately invoking numeric simulations and reasoning about qualitative aspects of the results. Bradley, Zhao and others applied similar ideas in the domain of nonlinear control [28], [30] and nonlinear system identification [29].

Despite these successes, the qualitative reasoning method has not been developed to the point where it can be used to solve more demanding, higher-dimensional, problems in robotics. Part of the problem is with the fact that the dynamics become categorically more complex when we move into higher dimensions [31], so that the alphabet explodes and traditional search and constraint satisfaction methods become too cumbersome to use, if not intractable. In this context, it is interesting to note the success of Hofmann [32], who used Bradley and Zhao's flow tubes idea to regularize an optimization problem for humanoid robot motion planning. What made this possible was the use of conventional model-based nonlinear control design tools to first transform the high-dimensional nonlinear system into decoupled linear systems, which can be controlled by multiple proportional-derivative controllers - such that high level planning and qualitative reasoning becomes tractable. The question remains as to whether it is possible to achieve similar results without reliance on

linearization and decoupling in the nonlinear control design sense, requiring precise models of dynamics and interaction effects.

This dissertation is based on similar insights as Yip, Bradley, Zhao and others but the problem is posed in a different way and qualitative modeling is combined with modern tools from machine learning to deal with the complexity of the dynamics. The template-anchor decomposition of robotic tasks naturally sets the stage for such an approach - where the template (which is still a nonlinear model, of sufficient complexity to fully specify the task) is amenable to qualitative reasoning and the embedding of the resulting strategy in the anchor is handled by machine learning.

Chapters 5 and 6 present one approach to making the transition from simple abstractions to the more complex systems of interest, through the use of appropriate phase-space geometric representations and machine learning algorithms. Recent advances in related areas, such as manifold learning [33, 34], computational geometry [35] and topology [36, 37] represent a repertoire of powerful tools that could be used to carry this program of research much further.

## **2.3 Motion Planning and Nonlinear Trajectory Generation**

As stated earlier, the primary focus of this dissertation is on the trajectory generation problem.

There is a long history of investigations into this general area, dating back to the writings of 17th and 18th century mathematicians involving optimality of paths, etc. Many deep concepts at the foundation of mechanics and dynamics, particularly the work of Lagrange, Jacobi, Hamilton, Bellman and others, are statements about the nature and behavior of optimal trajectories subject to various conditions and constraints [38].

From the point of view of applications, the major turning point was the development of techniques for numerical optimization that made it possible for these abstract ideas to be computed and implemented. Every standard textbook in robotics, graphics and animation includes an exposition on methods for computing parameterized trajectories via numerical optimization. Mombaur [39] presents an excellent recent example of the application of such methods for gait synthesis. Milam [40] presents another approach to nonlinear trajectory generation, demonstrating how it can be made to work with lower level nonlinear controllers.

Despite such successes, direct application of numerical optimization can be difficult

in the face of nontrivial dynamic constraints and conflicting requirements. These issues complicate the *optimization landscape* and result in sensitive dependence of the generated solution on the choice of initial condition, making the overall design process fragile.

In view of this, many modern motion planning algorithms take a different approach - that of (quasi)randomly sampling the free configuration space and approximating feasible paths using efficient data structures and algorithms. Techniques such as rapidly exploring random trees and probabilistic roadmap methods [41] are excellent examples of this trend. The RRT is a data structure that is constructed incrementally such that it quickly reduces the expected distance of a randomly chosen point in free space to the tree. An RRT can be intuitively considered as a Monte-Carlo way of biasing search into largest Voronoi regions and it has been used a way to generate open-loop trajectories for nonlinear systems with state constraints. Similarly, a probabilistic roadmap samples the free configuration space and constructs a graph that represents all feasible paths for the system, subject to obstacles.

Both the numerical optimization approach and the sampling based planning approach require the ability to evaluate the dynamics of the system at arbitrary points in phase space, which is only possible given a good analytical model or a suitable simulation of the complete state and action space. If one were to begin with only imprecise knowledge of the high-dimensional dynamics and the task constraints, this requirement can turn out to be problematic. For instance, a preferred approach to learning from experience is based on traces gathered from imperfect experience. This is difficult to achieve with the above methods.

In this connection, the work of Frazzoli [42], Dever [43] and others on defining global trajectories in terms of *parameterized maneuvers*, that then form the basis of optimization and sampling based planning is very appealing. One way to use these algorithms is to observe an expert who performs a task, such as acrobatic aerial maneuvering, and to organize these observations in terms of parameterized families that share underlying Lie group symmetries or other dynamical invariants. These works do not address all of the requirements stated in section 1.1 in that they are dependent on good models of dynamics and restrict themselves to interpolation of observed behaviors (which can cause problems when all observed behaviors are noisy or if a particular behavior of interest was never observed in the exactly desired form). Nonetheless, they clearly point in the desired direction and the qualitative models used in this dissertation will have many similarities to this concept of parameterized maneuvers.

## 2.4 Reinforcement Learning

Reinforcement learning [44] is an approach to solving sequential decision making tasks that does not require models of dynamics or the environment, that is suited to the paradigm of acquiring knowledge from experience and can accommodate uncertainties in state and action spaces. The basic idea is to discretize the state and action spaces, assign probabilities to transitions between states, thus setting up a (Partially Observable) Markov Decision Process which can then be updated and solved by various versions of dynamic programming algorithms.

Directly applying these algorithms to high-dimensional nonlinear systems results in poor convergence and intractability. In fact, as noted by Rosenstein and Barto [45],

*Reinforcement learning methods provide a means for solving optimal control problems when accurate models are unavailable. For many such problems, however, RL alone is impractical and the associated learning problem must be structured somehow to take advantage of prior knowledge.*

Recently, many researchers have explored ways to incorporate such additional knowledge [45, 46, 47]. In general, the approach has been to use classical stability arguments from control theory, e.g., Lyapunov functions or other stability criteria from linear systems theory, to filter or shape the outputs of the learning algorithm.

In the robotics literature, Tedrake [48] presents a very nice approach to robot design where the “prior knowledge” is not another controller but, in fact, a clever mechanical design that implements passive walking and simplifies the learning problem to one of performance optimization.

These are all highly desirable and instructive directions. However, the dependence on a fully functional pre-existing solution leaves room for improvements and new ideas. Can we eliminate this dependence or at least mitigate it to the point where it becomes possible to learn from crude, imperfect experience? This dissertation will show that the idea of encoding tasks in terms of manifolds and approximating these geometric structures from data will yield efficient learning, while allowing for more powerful encoding of imprecise “prior knowledge”.



## 2.5 Learning by Imitation, Lazy Learning and Manifold Learning

The main drift of the previous few sections is that there is *no free lunch* [49] in the sense that efficient solutions to many problems need to exploit the structure of the problem. It has been proposed by Schaal [50] that the efficient way to exploit problem structure, in tasks such as humanoid locomotion and manipulation, is to look at an existing solution and to imitate it. There have been many other attempts within the AI community to acquire skill in this way, see e.g., [51].

Atkeson et al. have proposed lazy learning algorithms [52] as the way to solve the imitation learning problem for a fairly broad class of systems and control tasks. The basic idea is to store all experience in a database and to locally, statistically, approximate control actions. The fact that all actions are based on direct experience makes it possible to capture some of the subtleties of the task fairly efficiently. However, as acknowledged by these authors, one of the major problems with direct application of this approach is that lazy learning depends crucially on having good representations already selected. So, while it is a very good idea - and one that will be adopted in this dissertation, its success depends on first solving the representation engineering problem.

In high dimensions, one of the reasons this approach works at all is that [52],

*... it is actually quite difficult to think of useful tasks that require the system to have an accurate model over the entire input space. Indeed for a robot of more than, say, eight degrees of freedom it will not be possible for it to get into every significantly different configuration even once in its entire lifetime. Many tasks require high accuracy only in low-dimensional manifolds of input space or thin slices around those manifolds.*

The recent development of manifold learning techniques in the machine learning community has enabled a new and direct attack on the lazy learning problem, by utilizing these manifold structures more explicitly. Much of this research has focussed on discovering the low-dimensional manifolds from observations such as human motion capture [53, 54]. The underlying machinery for solving these problems exploits the fact that computations involving nonlinear manifolds can be carried out in locally linear patches. So, for purposes of computation, the data can be laid out in a Euclidean space using extensions to the multidimensional scaling approach [33]. Recent work [34] has shown that most

manifold learning algorithms can be reformulated in a unified framework - as kernel learning algorithms. The point of dimensionality reduction algorithms for movement generation problems is that they make it possible to define control laws [55, 56] in a lower-dimensional subspace, reducing some of the computational burden of controller tuning and adaptation.

However, there are two problems with this program:

1. In the absence of external information about properties of the desired task manifold, manifold learning algorithms can easily make mistakes with imperfect and noisy observations. For instance, watching a robot “almost walk” for a period of time will not, by itself, yield enough information to induce the successful strategy.
2. As stated in [34], a combination of high curvature in the manifold and the presence of many manifolds (e.g., the case of parameterized maneuvers) can drive the data requirements for successful learning to be impracticably high.

The above problems are intrinsic to this local approach to learning, so there are no easy ways to alleviate them. However, we can modify the overall program such that, rather than depending on acquiring the strategy entirely from observations, we can utilize a template model to constrain the trajectory. This makes the data-driven manifold learning or approximation problem more tractable. Then, difficulties like high curvature and the presence of nearby manifolds are not complete unknowns but, in fact, are anticipated by the structure of the template dynamics. This is the approach taken in this dissertation.

## **2.6 Control Theory**

Many of the concerns mentioned in the problem statement in section 1.1 - uncertainty, non-linearity and robustness, have been shared by researchers within the control theory community. Several mathematically sophisticated methods have been developed to deal with them. However, the focus of much of modern control theory is on the local problems of feedback stabilization, around equilibrium points and around specified trajectories. The classical problem of control design is that of designing feedback control laws that modify the behavior of the system in question to enable stabilization and disturbance rejection, subject to various measures of performance such as sensitivity of outputs with respect to disturbance signals, etc. The linear control problem is considered to be completely solved and many efficient algorithms exist for stabilizing linear systems [57]. However, hardly any

system of practical interest in robotics is linear and modeled well enough to directly make use of this technology. Recent research in control theory has focussed on extending this technology in various ways. Robust control [58] theory takes the view that disturbances can be modeled in the same language as system dynamics and sensitivity to errors can be minimized, adaptive control theory [59] takes the view that a changing environment can be dealt with by modifying the controller in response to it and nonlinear control [60] is focused on dealing with nonlinearity by compensating it away - via feedback linearization, sliding modes and other techniques.

Despite the myriad mathematically sophisticated techniques that are available in this area, a key problem remains - as stated by Bernstein [61],

*What makes some control problems hard? Our holistic point of view is that a control problem is hard when multiple impediments occur simultaneously. Constraints on physics, architecture, accessibility, authority, nonlinearity, instability, dimensionality, uncertainty and noise can often be overcome without much difficulty when they are effectively the only operative constraint. However, when multiple constraints are present, the control problem becomes vastly more difficult.*

It is precisely this combination of constraints that makes robot motion problems challenging!

One promising approach to dealing with this issue is to divide and conquer - use multiple models [7], each specialized to small regions of state space. Recent investigations into *hybrid systems and control* develop this approach by considering systems that are composed of logical and dynamic elements - different local models of dynamics and control are orchestrated by a higher level logical automaton. Many interesting results have come out of these studies but we are still far from being able to design robots using this technical machinery. Part of the reason for this is the intractability of the general formulation of hybrid systems problems, even elementary versions of which can be NP-hard as formulated [62].

This suggests the need for an alternate approach to the overall problem. A more pragmatic approach to hybrid system design is found in work such as by Burrige et al. [63], who design several local controllers for primitive behaviors and then sequence them together with a planning algorithm. In the graphics community, a similar approach has been taken by Faloutsos et al. [64] who use support vector machine learning to appropriately sequence hand crafted controllers.

This dissertation adopts a similar approach to controller composition, but local regions and control strategies are formulated differently, in such a way that they would not be restricted to stable basins as required in [63], but could also be more complex orbits or unstable systems.

## 2.7 Control in Biology and Biomimetic Robotics

So far, in this chapter, the focus has been on algorithmic techniques for movement generation and control that were largely developed in the engineering setting. It is well worth remembering that for many robotics problems, natural solutions in the biological world are significantly better than corresponding engineering solutions in terms of a combination of efficiency, flexibility and tolerance for imprecision.

How do animals achieve this? There seem to be many design principles that organize the planning and control process [4, 6]:

- Use of natural dynamics: Locomotion and other behaviors in nature involve significant periods of purely ballistic behavior, utilizing the free swing and motion of limbs and appendages [11, 10, 65, 66].
- Use of motion primitives: Similarly, motion generation in biology seems to be organized modularly, in terms of units that generate primitive behaviors such as the generation of oscillations by central pattern generators [67, 68].
- Modulation of primitives by sensory feedback: While the previous two principles suggest simplification strategies, they are insufficient to explain complex behaviors in unstructured environments. Realistic motion generation must involve the modulation of the above elements based on sensory feedback.

Animals adopt such strategies for a number of reasons, including the slow bandwidth of the neural circuitry that necessitates decentralization and compartmentalization of control elements. Robots do not suffer from such bandwidth limitations but they can still benefit from these same principles. This view is shared by several robotics researchers who have seriously investigated ways to embody the above principles in robots, e.g., [69, 70]. In recent years, numerous robot implementations have utilized CPG like periodic gaits [70]. Westervelt and collaborators [71, 72] have demonstrated that it is possible to generate asymptotically stable limit cycles in planar walking robots, using sophisticated nonlinear

control design. Pratt and collaborators [73] have demonstrated that this kind of periodic behavior is not critically dependent on specialized nonlinear control elements and it can be achieved using relatively simple control strategies, composed from "intuitive" design principles and clever utilization of the natural dynamics.

However, this exclusive emphasis on stable periodic orbits focusses attention on a relatively uncommon operational mode in nature. As observed by Dickinson et al. [4],

*Because steady locomotion is more the exception than the rule for animals traveling through natural environments, mechanistic studies of animal behavior must consider not only how animals generate locomotory patterns, but also how they control them in both time and space.*

In fact, although much of the emphasis in the robotic locomotion literature has been on the stability and performance of closed-loop controllers around limit cycles, much of the task-level stabilizing control authority in legged locomotion seems to come from appropriate foot-placement, with which even passive walkers can be compelled to safely achieve tasks, see, e.g., [74]. With this in mind, the focus of this dissertation will be on techniques for trajectory generation that can accommodate a range of foot placement goals, and emphasis will be placed on the question of how passive dynamics and primitive behavioral orbits can be combined with active planning to achieve any goal within this range.

This brings us to the related point of motor learning. While it is true that important modules such as central pattern generators seem to be hard-wired into the developmental and neural processes of animals, many interesting tasks known to animals are learned over the course of the organism's lifetime in response to a changing environment [3]. Given our primitive understanding of motion generation in robotics, the role of learning can only be greater. How to learn dynamic behaviors in a system that also utilizes these other biological principles is then a very important question that will be addressed in this dissertation.

## Chapter 3

# Task Encoding Using Qualitative Models

An autonomous robot must operate with limited knowledge of the structure of the environment (hence, of future goals) as well as the structure of its own dynamics. It must also achieve tasks that are extended in both time and space. This requirement of tolerance to lack of predefined structure, combined with the inherent complexity of the task, is unusual among engineering systems and for this reason many traditional concepts from planning and control will need to be modified or improved upon to address this need.

To begin with, one needs to have a single consistent representation for what needs to be done - a language for *task encoding*. With this language, one can then develop design techniques and principles to tackle a variety of different problems.

This chapter presents a methodology that involves representing dynamical tasks in terms of structurally stable orbits, which result from classes of dynamical systems that we call *qualitative models*. To the extent that the dynamical task is embedded in a changing environment, structural stability allows us to describe what does, and should, be preserved despite these changes. Furthermore, the fact that the dynamical properties are associated with classes of models enables the design of strategies at a qualitative level, in terms of functional constraints, leaving room for further quantitative specialization to occur at “run-time”.

It is difficult to make broadly applicable statements about nonlinear systems that are also precise enough to capture interesting task specifications. With this in mind, the arguments in this chapter will be presented through the solution of a specific nontrivial design

problem - global control of the inverted pendulum and cart-pole systems. This problem involves a variety of concerns that are common in robotics, including nonlinearity, under-actuation, state and actuation constraints.

The inverted pendulum problem has been studied for several decades now, and an elementary version of this problem is a standard textbook exercise - to stabilize the pole upright in a local neighborhood of the fixed point of the linearized system using linear state feedback control. However, the global problem is nontrivial. For instance, it can be shown that there is no single continuous controller that can stabilize the cart-pole system globally. Such technical issues make the design problem one of continuing research interest, and one that has been addressed extensively in the recent research literature [76, 77, 78, 79, 80]. Speaking more broadly, the cart-pole system studied in this chapter provides a single succinct problem using which we address the variety of concerns that make control design hard [61] - so, although the solution pertains to a specific problem, similar ideas can be applied to numerous other problems that share similar characteristics.

This chapter provides a novel solution to this design problem. The solution not only addresses the need of global stability, but also constraints (finite actuator strength, finite track length) and effects such as static friction. Beyond that, from the point of view of the overall thesis, this chapter uses this concrete example to lay out our approach to specifying controlled behaviors in terms of multiple qualitative models, composed together based on considerations of the *natural dynamics*, such as energy level curves.

We begin with a general introduction to multiple model design, an overview of the qualitative model based approach and some results about the qualitative behavior of second order systems. These concepts are then applied to solve the problem of designing global controllers for the pivot-actuated pendulum and the cart-pole system.

The work presented in this chapter draws on several concepts from nonlinear and robust control theory, adapting and extending them to the particular needs of our problem. These connections are mentioned at various points in this chapter. However, the goal of this chapter is different from other control theoretic investigations in that we wish to develop a planning and control methodology that can be seamlessly combined with machine learning algorithms (to be developed in later chapters), making the methodology applicable to problems that are more demanding in terms of the combination of high-dimensional non-linearity, constraints and model imprecision. Section 3.7 will summarize the main ideas in this chapter that will be developed further in the remainder of this dissertation.

### 3.1 Multiple Model Approaches to Nonlinear Modeling and Control

An efficient way to deal with the problem of complexity in dynamical models is through a decomposition into multiple models, each of which is simpler in some desirable way. The book by Murray-Smith [7] presents an excellent overview of this general approach to modeling and control.

A canonical way to think about multiple model problems is in terms of operating regimes. The essence of this approach is to partition the operating range of the system such that each simpler local model can be associated with an operating regime and a higher level automaton can coordinate these models as discrete symbolic entities. One way to represent the dynamics of the system using multiple models is,

$$\dot{x}(t) = f_{q(t)}(x(t), u(t)) \quad (3.1)$$

$$q(t+1) = \mathcal{A}(q(t), x(t), u(t)) \quad (3.2)$$

where  $x(t) \in \mathfrak{R}^N$  is a continuous state vector,  $u(t) \in \mathfrak{R}^M$  is a corresponding control input vector and  $q(t) \in \mathbb{Z}$  is a discrete variable that indexes the operating regime. So, there are as many continuous models, i.e., equations for  $\dot{x}(t)$ , as there are distinct values of  $q(t)$ . Here the discrete variable is written as evolving on a unit-time base, but there are numerous alternatives that could be considered, including state driven change in the value of  $q$ . The notion of piecing together multiple simpler models has a long history in engineering practice. However, traditional practice has often been based on *ad hoc* design choices whereas we seek a more principled approach that provides at least the following:

- Parametrization of the local models in terms of a few, intuitive, variables
- Methods for model coordination such that properties of the global behavior may be explicitly designed up front (as opposed to the exhaustive testing that typically follows an *ad hoc* design).

Over the past few decades, several techniques for multiple model design have been invented. Some representative examples include:

- Gain scheduling [59]: A very popular approach to dealing with nonlinear dynamics, particularly in aerospace and other mechanical control problems, where multiple



linear models are stitched together by a scheduling variable that captures the nonlinearity. Here,  $q(t)$  is tied to  $\langle x(t), u(t) \rangle$ , and each instance of equation 3.1 is a linear system.

- Fuzzy control [87]: Instead of the hard scheduling implied by equation 3.2 and in techniques such as gain scheduling, one could do a smoother form of composition, whose semantics are derived from fuzzy logic and fuzzy set theory. In fuzzy control, the operation of equation 3.2 is now handled by a more elaborate process involving membership functions and inference mechanisms.
- Graphical models and Basis function methods [7, 88]: It is possible to adopt the membership function idea without reliance on the fuzzy inference process. This is achieved through the use of basis functions. This technique has a sound mathematical foundation in approximation theory and forms an established method in the neural networks literature. The idea here is to sum the effect of all local models, weighted by the basis functions that act as a kernel.

This chapter presents an alternate approach to the definition and use of operating regimes, in terms of families of phase space orbits and aspects of the *natural* dynamics - such as energy exchange arguments.

## 3.2 Qualitative Dynamics and a Heterogeneous Control Problem

The phase space of a dynamical system is the product space of the configuration variables,  $x$ , and the velocities,  $\dot{x}$ . In robotics, configuration variables are typically the positions and angles that can be measured easily and fairly directly. However, knowing the configuration is not sufficient to uniquely describe the underlying dynamics - the system can evolve in different ways from the same initial configuration depending on velocity. The pair of position and velocity is sufficient to uniquely identify behaviors of a dynamical system and to formally describe its properties.

Concretely, consider the phase space of the simple pendulum system depicted in figure 3.1, which will play an important role in later arguments <sup>1</sup>.

---

<sup>1</sup>In fact, to the extent that most robots involve pendulum-like appendages, this phase space is of importance to numerous other robotic manipulation and locomotion applications as well.

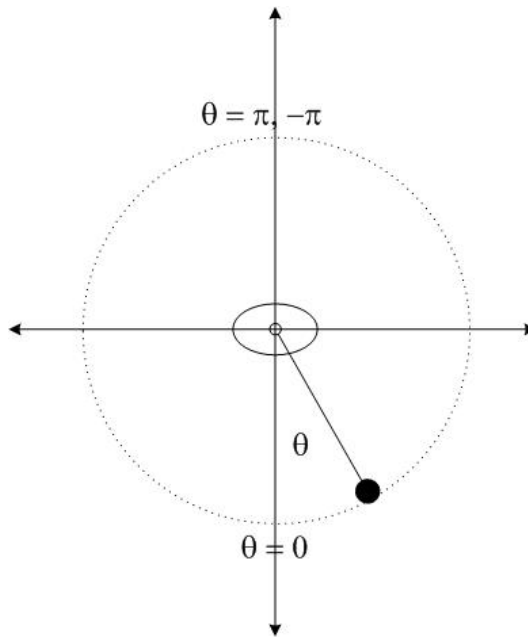


Figure 3.1: A simple pendulum, driven by a torque applied at the pivot.

The equation of motion of the frictionless version of this system takes the form,

$$\ddot{\theta} + k \sin \theta = 0 \quad (3.3)$$

where  $k = \frac{g}{l}$  is a constant that characterizes the motion. Solutions of this equation can be described in terms of the first integral of motion [86],

$$\dot{\theta} = \sqrt{\frac{2g}{l}(\cos \theta - \cos \theta_0)} \quad (3.4)$$

These solutions can be organized in the form of a phase space portrait as shown in figure 3.2. This system has two equilibrium points - the stable elliptic fixed point at  $\theta = 0$  and the unstable hyperbolic fixed point at  $\theta = \pm\pi$ . One of the most important aspects of this phase portrait is the fact that there is a single trajectory, called the *separatrix*, that connects the elliptic and hyperbolic fixed points. This trajectory also divides the phase space trajectories into two families - the “small oscillation” or *libration* trajectories, which oscillate around  $\theta = 0$ , never visiting  $\theta = \pm\pi$ , and the *rotation* trajectories that make full circles. Intuitively, one can think of the libration trajectories as being akin to the motion of

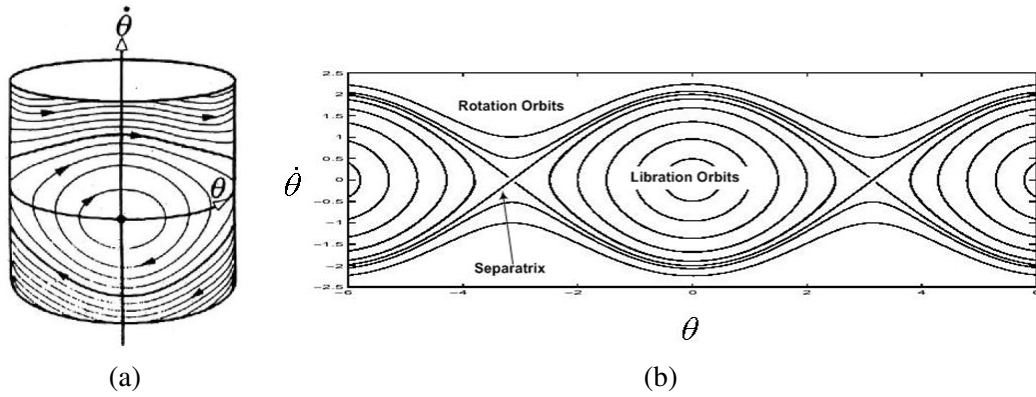


Figure 3.2: Phase portrait of the simple frictionless pendulum, depicting the separatrix trajectory which divides the phase space into two families of orbits - rotation and libration. Part *a* depicts the fact that  $\theta \in S^1$ , while part *b* depicts an “opened out” version of the phase portrait.

child on a swing while the latter is closer to that of a circus acrobat performing tumbling maneuvers.

A particularly interesting aspect of this phase portrait is that all trajectories are parameterized by a single variable - total energy. The separatrix has exactly the same amount of energy at all points as the hyperbolic fixed point,  $(\theta = \pi, \dot{\theta} = 0)$ . All rotation trajectories have more than this much energy and all libration trajectories have less. We will use this fact in subsequent arguments.

Using this phase space, we are in a position to state the first design problem - global control of the pendulum. For this, consider a version of the pendulum that has some friction at the pivot<sup>2</sup>, where it is also driven by a torque  $u$ ,

$$\ddot{\theta} + f(\dot{\theta}) + k \sin \theta + u(\theta, \dot{\theta}) = 0 \quad (3.5)$$

The objective is to design a control law,  $u(\theta, \dot{\theta})$ , that can bring the system to the hyperbolic fixed point from the elliptic fixed point, and stabilize it there. The key requirement is global control, i.e., despite arbitrarily large deviations from  $(\theta = \pi, \dot{\theta} = 0)$  the system should eventually return to stabilize at this point. Moreover, this needs to be achieved with bounded torques,  $u(\theta, \dot{\theta}) < u_{max}$ .

<sup>2</sup>Note that we are deliberately leaving the functional form of the friction term unspecified, to emphasize the difficulty in modeling such terms exactly.

Before presenting the solution to this design problem, we will discuss several results that will make the notion of “qualitative dynamics” more concrete.

### 3.3 Some Results on the Qualitative Dynamics of Second Order Systems

There are numerous techniques for the analysis of dynamical properties of nonlinear systems. In this section, we restrict attention to a relatively simple class of systems that can be represented in terms of qualitative differential equations [21]. This representation, and the way we will use it, is a simple instance of a much more general approach to modeling classes of systems in terms of set valued maps [81]. However, in this chapter, we will find the concrete representation of QDEs to be sufficient for our purposes.

The purpose of the results presented in this section is to identify *qualitative model* classes that can be used to describe the global behavior in concise ways. One way to use this, within the multiple model framework, is to assign to each local model a behavior described by a qualitative model class. This is in the spirit of model-reference control [59, 60], although the details differ due to the fact that we are dealing with model classes as opposed to concrete quantitative models.

In this section, the focus is on systems whose equations of motion, such as the one in equation 3.5, involve “reasonable” functions - defined below along with some useful concepts for expressing qualitative models.

**Definition 1.** Where  $[a, b] \subseteq \mathfrak{R}^* = \mathfrak{R} \cup \{-\infty, \infty\}$ , the function  $f : [a, b] \rightarrow \mathfrak{R}^*$  is a *reasonable function* over  $[a, b]$  if

1.  $f$  is continuous on  $[a, b]$ ,
2.  $f$  is continuously differentiable on  $(a, b)$ ,
3.  $f$  has only finitely many critical points in any bounded interval
4. The one-sided limits  $\lim_{t \rightarrow a^+} f'(t)$  and  $\lim_{t \rightarrow b^-} f'(t)$  exist in  $\mathfrak{R}^*$ . Define  $f'(a)$ ,  $f'(b)$  to be equal to these limits.

**Definition 2.**  $M^+$  is the set of reasonable functions  $f : [a, b] \rightarrow \mathfrak{R}^*$  such that  $f' > 0$  over  $(a, b)$ .

**Definition 3.**  $M_0^+$  is the set of  $f \in M^+$  such that  $f(0) = 0$ . In the nonlinear control literature, these functions are referred to as *Class- $\kappa$*  functions [60].

**Definition 4.**  $[x]_0 = \text{sign}(x) \in \{+, 0, -\}$ .

**Definition 5.**  $P_0^+$  is the set of reasonable functions  $f : [a, b] \rightarrow \mathfrak{R}^*$  such that  $[f(x)]_0 = [x]_0$  over  $(a, b)$ .

A canonical second order system is the following QDE model,

$$\ddot{x} + f(\dot{x}) + g(x) = 0 \quad (3.6)$$

that encompasses a large number of ordinary differential equation models. Here,  $f$  and  $g$  are reasonable functions that might belong to various classes such as  $M_0^+$  or  $P_0^+$ , including a whole variety of nonlinear elements that might be found in engineering practice. The primary benefit of using this representation is this ability to talk about a family of models, all of which share common dynamical properties. One way to reason about such systems is through the use of automated tools, of which QSIM [21] is a good example. In [26, 82], properties of various QDE systems are rigorously established by qualitative simulation and temporal logic theorem proving. This is particularly useful when discussing properties of a network of compartmentalized systems described by QDEs [82] and this is one of the major benefits of adopting this formalism to describe dynamics. Many hybrid system analysis and design techniques have a similar flavor as well, see, e.g., [83, 84].

For the purposes of this chapter, it is much more convenient to directly establish relevant results by hand. The following lemmas are statements about properties of specific types of second order systems that will play a role in subsequent sections.

**Lemma 3.3.1 (Qualitative damped spring model)** *Let  $S$  be a dynamical system described by the QDE*

$$\ddot{x} + f(\dot{x}) + g(x) = 0 \quad (3.7)$$

where  $f, g \in P_0^+$ . Then  $S$  has a Lyapunov function

$$V(x, \dot{x}) = \frac{1}{2}\dot{x}^2 + \int_0^x g(x)dx$$

such that  $V(0, 0) = 0$ ,  $V > 0$  elsewhere, and  $dV/dt < 0$  everywhere except that  $dV/dt = 0$  where  $\dot{x} = 0$ . Therefore,  $S$  is asymptotically stable at  $(0, 0)$ .

**Proof** We rewrite equation 3.7 as,

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2) = x_2 \\ \dot{x}_2 &= f_2(x_1, x_2) = -f(x_2) - g(x_1)\end{aligned}\tag{3.8}$$

Because  $g$  is a reasonable function, we know that  $g'(0)$  is defined. Since  $[g(x)]_0 = [x]_0$ , we conclude that  $g(0) = 0$  and  $g'(0) > 0$ . Similarly for  $f$ . Any fixed point of equation 3.7 must satisfy  $\dot{x}_1 = \dot{x}_2 = 0$ , which implies that the only fixed point is at  $x_1 = x_2 = 0$ .

The qualitative behavior of the nonlinear system (equation 3.7) in the neighborhood of the fixed point at  $(0,0)$  is the same as that of its local linearization [85]:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -f'(0)x_2 - g'(0)x_1\end{aligned}\tag{3.9}$$

Taking the eigenvalues,

$$\lambda_{1,2} = \frac{1}{2} \left[ -f'(0) \pm \sqrt{f'(0)^2 - 4g'(0)} \right]\tag{3.10}$$

Because  $f'(0), g'(0) > 0$ , the eigenvalues have negative real parts, so  $(0,0)$  is a stable fixed point. When the “friction force” term  $f'$  is small relative to the “spring force” term  $g'$ , the eigenvalues will be complex, in which case  $(0,0)$  will be a spiral attractor.

Because  $[g(x)]_0 = [x]_0$ , the “spring force” is always a restoring force, so we can define a Lyapunov function

$$V(x, \dot{x}) = \frac{1}{2}\dot{x}^2 + \int_0^x g(x)dx\tag{3.11}$$

Clearly,  $V(x, \dot{x}) \geq 0$ , and  $V = 0$  only at  $(0,0)$ .

$$\begin{aligned}
\dot{V} &= \dot{x}\ddot{x} + g(x)\dot{x} \\
&= \dot{x}(-f(\dot{x}) - g(x)) + g(x)\dot{x} \\
&= -\dot{x}f(-\dot{x}) \leq 0
\end{aligned} \tag{3.12}$$

Therefore,  $S$  is asymptotically stable at  $(0,0)$ , and has no limit cycles. ■

**Lemma 3.3.2 (Qualitative anti-damped spring model)** *Let  $S$  be a dynamical system described by the QDE*

$$\ddot{x} - f(\dot{x}) + g(x) = 0 \tag{3.13}$$

where  $f, g \in P_0^+$ . Then  $(0,0)$  is the only fixed point of  $S$ , it is unstable, and there are no limit cycles.

**Proof** The proof of this lemma is very similar to the previous one. We rewrite equation 3.13 as

$$\begin{aligned}
\dot{x}_1 &= f_1(x_1, x_2) = x_2 \\
\dot{x}_2 &= f_2(x_1, x_2) = f(x_2) - g(x_1)
\end{aligned} \tag{3.14}$$

As before, because  $g$  is a reasonable function, we know that  $g'(0)$  is defined. Since  $[g(x)]_0 = [x]_0$ , we conclude that  $g(0) = 0$  and  $g'(0) > 0$ . Similarly for  $f$ .

Any fixed point of equation 3.13 must satisfy  $\dot{x}_1 = \dot{x}_2 = 0$ , so the only fixed point is  $x_1 = x_2 = 0$ .

As before, the qualitative behavior of the nonlinear system (equation 3.13) around the fixed point at  $(0,0)$  is the same as that of its local linearization:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= f'(0)x_2 - g'(0)x_1
\end{aligned} \tag{3.15}$$

The eigenvalues of this system are

$$\lambda_{1,2} = \frac{1}{2} \left[ f'(0) \pm \sqrt{f'(0)^2 - 4g'(0)} \right] \tag{3.16}$$

In this case, since  $f'(0) > 0$ , the eigenvalues have positive real parts, and  $(0,0)$  is

an unstable fixed point. If the “friction force” term  $f'$  is small relative to the “spring force” term  $g'$ , then the eigenvalues will be complex, so  $(0,0)$  will be a spiral repeller.

By the Bendixon negative criterion [85], there can be no periodic orbits contained in the phase space of  $S$  because

$$\frac{\partial f_1}{\partial x_1} + \frac{\partial f_2}{\partial x_2} = f'(x_2) \quad (3.17)$$

is always positive. Therefore, while  $S$  has a fixed point at  $(0,0)$ , that fixed point is unstable, and there can be no limit cycles. ■

There is another form of these lemmas that can sometimes be useful. It is possible that the “friction force” term in these models may consist of a product of functions of velocity and position, as shown in the following lemmas.

**Lemma 3.3.3 (Qualitative damped spring model, ver. 2)** *Let  $S$  be a dynamical system described by the QDE*

$$\ddot{x} + f(\dot{x})h(x) + g(x) = 0 \quad (3.18)$$

where  $f, g \in P_0^+$ , and  $h$  is a reasonable function such that  $h(x) \geq 0$  and  $h(0) > 0$ . Then  $S$  is asymptotically stable at  $(0,0)$ .

**Proof** The proof is similar to that of Lemma 3.3.1.

We rewrite equation 3.18 as,

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) = x_2 \\ \dot{x}_2 &= f_2(x_1, x_2) = -f(x_2)h(x_1) - g(x_1) \end{aligned} \quad (3.19)$$

Because  $g$  is a reasonable function, we know that  $g'(0)$  is defined. Since  $[g(x)]_0 = [x]_0$ , we conclude that  $g(0) = 0$  and  $g'(0) > 0$ . Similarly for  $f$ . Further, we define  $h$  to nonnegative, and strictly positive at zero. Any fixed point of equation 3.18 must still satisfy  $\dot{x}_1 = \dot{x}_2 = 0$ , which implies that the only fixed point is at  $x_1 = x_2 = 0$ .

Taking the local linearization,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -f'(0) h(x_1)|_{x_1=x_2=0} x_2 - f(x_2)|_{x_1=x_2=0} h'(0)x_1 - g'(0)x_1 \end{aligned} \quad (3.20)$$



The eigenvalues of this system are

$$\lambda_{1,2} = \frac{1}{2} \left[ -f'(0)h(0) \pm \sqrt{(f'(0)h(0))^2 - 4(g'(0)f'(0)h'(0))} \right] \quad (3.21)$$

Because  $f'(0), h(0) > 0$ , the eigenvalues have negative real parts, so  $(0,0)$  is a stable fixed point.

Because  $[g(x)]_0 = [x]_0$ , the “spring force” is always a restoring force, so we can define a Lyapunov function

$$V(x, \dot{x}) = \frac{1}{2}\dot{x}^2 + \int_0^x g(x)dx \quad (3.22)$$

As in the proof of lemma 1, the only fixed point is at  $(0,0)$  and the local behavior at  $(0,0)$  is stable. Using the Lyapunov function,

$$\begin{aligned} \dot{V} &= \dot{x}\ddot{x} + g(x)\dot{x} \\ &= \dot{x}(-f(\dot{x})h(x) - g(x)) + g(x)\dot{x} \\ &= -\dot{x}f(\dot{x})h(x) \end{aligned} \quad (3.23)$$

which has a consistent negative sign, except for isolated zeros, so the system is asymptotically stable. ■

**Lemma 3.3.4 (Qualitative anti-damped spring model, ver. 2)** *Let  $S$  be a dynamical system described by the QDE*

$$\ddot{x} - f(\dot{x})h(x) + g(x) = 0 \quad (3.24)$$

where  $f, g \in P_0^+$ , and  $h$  is a reasonable function such that  $h(x) \geq 0$  and  $h(0) > 0$ . Then  $(0,0)$  is the only fixed point of  $S$ , it is unstable, and there are no limit cycles.

**Proof** The proof is similar to that of Lemma 3.3.2. We rewrite equation 3.24 as,

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) = x_2 \\ \dot{x}_2 &= f_2(x_1, x_2) = f(x_2)h(x_1) - g(x_1) \end{aligned} \quad (3.25)$$

As before, because  $g$  is a reasonable function, we know that  $g'(0)$  is defined. Since  $[g(x)]_0 = [x]_0$ , we conclude that  $g(0) = 0$  and  $g'(0) > 0$ . Similarly for  $f$ ,  $[f(\dot{x})]_0 = [\dot{x}]_0$

and  $f'(0) > 0$ . We have also assumed that  $h(x) \geq 0$  and  $h(0) > 0$ .

Taking the local linearization,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f'(0) h(x_1)|_{x_1=x_2=0} x_2 + f(x_2)|_{x_1=x_2=0} h'(0)x_1 - g'(0)x_1 \end{aligned} \quad (3.26)$$

Any fixed point of equation 3.24 must satisfy  $\dot{x}_1 = \dot{x}_2 = 0$ , so the only fixed point is  $x_1 = x_2 = 0$ .

The eigenvalues of this system are

$$\lambda_{1,2} = \frac{1}{2} \left[ f'(0)h(0) \pm \sqrt{(f'(0)h(0))^2 - 4(g'(0)f'(0)h'(0))} \right] \quad (3.27)$$

Because  $f'(0), h(0) > 0$ , the eigenvalues have positive real parts, so  $(0,0)$  is an unstable fixed point.

By the Bendixon negative criterion, there can be no periodic orbits contained in the phase space of  $S$  because

$$\frac{\partial f_1}{\partial x_1} + \frac{\partial f_2}{\partial x_2} = f'(x_2)h(x_1) \quad (3.28)$$

is not always zero and does not change sign. Therefore, while  $S$  has a fixed point at  $(0,0)$ , that fixed point is unstable, and there can be no limit cycles. ■

The dynamical systems literature includes many results that could be used to define other similar forms of local models. These results demonstrate that a variety of dynamical behaviors, not just stable equilibria, can be efficiently described by the language of QDEs. One such result, adapted from a theorem due to Levinson and Smith [86], can be used to construct relaxation oscillators. Such an oscillator is a very robust way to obtain a periodic orbit from a large set of initial conditions.

**Lemma 3.3.5 (Relaxation Oscillators: Levinson-Smith)** *Let  $S$  be a dynamical system described by the QDE*

$$\ddot{x} - f(x)\dot{x} + g(x) = 0 \quad (3.29)$$

where  $f, g$  satisfy the following assumptions:

1.  $f$  is an even reasonable function
2.  $g \in P_0^+$  is an odd reasonable function

3.  $\exists a > 0$  such that  $F(x) \equiv \int_0^x f(s)ds < 0$  on  $0 < x < a$ ,  $F(x) > 0$  on  $x > a$  and  $f(x) > 0$  on  $x > a$ .
4.  $G(x) \equiv \int_0^x g(s)ds \rightarrow \infty$  as  $|x| \rightarrow \infty$  and  $F(x) \rightarrow \infty$  as  $x \rightarrow \infty$ .

Then the system has a non-constant, orbitally stable periodic solution  $p(t)$ , that is unique up to translations  $p(t + \tau)$ ,  $\tau \in \mathbb{R}$ .

The proof of this result can be found in standard texts on nonlinear differential equations [86].

**Remark** There exist natural extensions of the above results, regarding qualitative dynamics of second order systems, to higher order systems. However, the dynamics become significantly more complex with increasing dimensionality [75, 31]. In this dissertation, we choose not to deal with this complexity directly - such as by deriving detailed functional conditions for orbits in higher dimensions. Instead, for the purposes of design, we restrict qualitative analyses to suitably low dimensions and we utilize approximation and learning algorithms that will embed the resulting strategies in higher dimensional systems.

As stated earlier, the phase space of the pendulum is parameterized by the total energy and the global behavior can be largely described with the aid of this one variable. For this reason, the total energy will play a useful role in subsequent sections, which will be devoted to global properties of control strategies designed using results from this section.

The following results regarding the work done by the system are also sometimes useful. In particular, they describe the decomposition of the work done into a conservative component, which flows back and forth between kinetic and potential energy, and a non-conservative component, which is dissipated or injected into the system. In this way, we can reason about long term properties of trajectories in terms of the conservative and non-conservative energy components. Among other things, these lemmas allow us to describe energy in a discrete setting such as via a Poincaré map.

**Lemma 3.3.6** *Consider a system that can be decomposed into two components*

$$\ddot{x} = F(x, \dot{x}, t) = C(x) + N(x, \dot{x}, t). \quad (3.30)$$

*Consider two times  $t_1, t_2$  and let  $x_1 = x(t_1)$  and  $x_2 = x(t_2)$ , and let  $v_1 = \dot{x}(t_1)$  and*

$v_2 = \dot{x}(t_2)$ . Then

$$\frac{1}{2}v_2^2 - \frac{1}{2}v_1^2 = \int_{x_1}^{x_2} C(x)dx + \int_{t_1}^{t_2} N(x, \dot{x}, t)\dot{x}dt$$

**Proof** Consider the work done by the system over  $[t_1, t_2]$ ,

$$\int_{t_1}^{t_2} \ddot{x}\dot{x}dt = \int_{t_1}^{t_2} \dot{x} \frac{d\dot{x}}{dt} dt = \int_{v_1}^{v_2} \dot{x}d\dot{x} = \frac{1}{2}v_2^2 - \frac{1}{2}v_1^2 \quad (3.31)$$

Now consider an alternate transformation of the same work:

$$\int_{t_1}^{t_2} \ddot{x}\dot{x}dt = \int_{t_1}^{t_2} F(x, \dot{x}, t)\dot{x}dt = \int_{t_1}^{t_2} C(x)\dot{x}dt + \int_{t_1}^{t_2} N(x, \dot{x}, t)\dot{x}dt \quad (3.32)$$

Setting the two terms equal, we obtain the desired result. ■

The kinetic energy theorem in [21] is a more general application of this method.

This result can be specialized to investigate the behavior of a trajectory of the system on successive times that it intersects a particular line in the phase plane - the Poincare section. This helps us analyze systems in a discrete setting, where energy or state variables take on extremal values. First, consider the change in velocity between times when  $x(t)$  is equal to a particular value,

**Corollary 3.3.7** Consider a system that can be decomposed into two components

$$\ddot{x} = F(x, \dot{x}, t) = C(x) + N(x, \dot{x}, t). \quad (3.33)$$

Let  $t_1, t_2$  be two times such that  $x(t_1) = x(t_2)$ , and let  $v_1 = \dot{x}(t_1)$  and  $v_2 = \dot{x}(t_2)$ . Then

$$\frac{1}{2}v_2^2 - \frac{1}{2}v_1^2 = \int_{t_1}^{t_2} N(x, \dot{x}, t)dt \quad (3.34)$$

**Corollary 3.3.8** Consider a system that can be decomposed into two components

$$\ddot{x} = F(x, \dot{x}, t) = C(x) + N(x, \dot{x}, t). \quad (3.35)$$

Let  $t_1, t_2$  be two times such that  $v = \dot{x}(t_1) = \dot{x}(t_2) = 0$ . Then

$$\int_{x_1}^{x_2} C(x)dx = - \int_{t_1}^{t_2} N(x, \dot{x}, t)\dot{x}dt \quad (3.36)$$

Third, consider the special case where an initial kinetic energy is transferred to potential energy:  $(0, v_1) \rightarrow (x_2, 0)$ .

**Corollary 3.3.9** Consider a system that can be decomposed into two components

$$\ddot{x} = F(x, \dot{x}, t) = C(x) + N(x, \dot{x}, t). \quad (3.37)$$

Let  $t_1, t_2$  be two times when  $x(t_1) = 0, v_1 = \dot{x}(t_1), x_2 = x(t_2)$ , and  $v_2 = \dot{x}(t_2) = 0$ . Then

$$\int_0^{x_2} C(x)dx = -\frac{1}{2}v_1^2 - \int_{t_1}^{t_2} N(x, \dot{x}, t)\dot{x}dt \quad (3.38)$$

### 3.4 A Global Control Strategy for the Pendulum

One of the key properties of the pendulum system is that if it is initialized on the separatrix trajectory then its dynamics will evolve in such a way that it eventually reaches the hyperbolic fixed point. Moreover, this trajectory is uniquely characterized by the total system energy, which must equal the energy of the point  $\langle \theta = \pi, \dot{\theta} = 0 \rangle$ . If the system begins with less energy than this point, such as when it begins close to the elliptic fixed point,  $\langle \theta = 0, \dot{\theta} = 0 \rangle$ , then energy needs to be injected into the system in order for it to reach the desired goal. Correspondingly, if the system has more energy, then energy needs to be drained in order for it to reach the point  $\langle \theta = \pi, \dot{\theta} = 0 \rangle$ . Note that we are not just interested in reaching the vertical position,  $\theta = \pi$ , instead we need to reach a phase space point that also involves stopping at this position, i.e.,  $\dot{\theta} = 0$ . Once on the separatrix, if the energy is maintained then the system will evolve towards the desired equilibrium point <sup>3</sup>.

This is a simple, highly intuitive, scheme that can be used to achieve the task of globally stabilizing the pendulum. However, a naive application of this strategy is not sufficient. For instance, we could find the system oscillating between having too much and too little energy, never staying on the separatrix long enough to be brought to the desired

---

<sup>3</sup>The separatrix trajectory reaches the goal asymptotically which means that it takes infinite time. However, the designs presented in this chapter will have much better convergence properties, which has been verified in practice.

goal. So, in fact, the design must address not only the local issue of energy control, but also the global issue of task achievement.

**Remark** In this chapter, for ease of exposition, we will use two coordinate frames to refer to the same system dynamics -  $\langle \theta, \dot{\theta} \rangle$  and  $\langle \phi, \dot{\phi} \rangle$  which are *separated* by an angle of  $\pi$  such that the hyperbolic fixed point is at  $(\phi = 0, \dot{\phi} = 0)$  and the elliptic fixed point is at  $(\theta = 0, \dot{\theta} = 0)$ .

### 3.4.1 Stabilizing the Pendulum: Balance Mode

The QDE describing the dynamics of the pendulum near the hyperbolic fixed point is,

$$\ddot{\phi} + f(\dot{\phi}) - g(\phi) + u(\phi, \dot{\phi}) = 0 \quad (3.39)$$

where  $g(\phi) = k \sin \phi$ .

In order for the system to stabilize at the desired goal,  $\langle \phi = 0, \dot{\phi} = 0 \rangle$ , its dynamics must be that of the system in Lemma 3.3.1. This provides a simple sufficient condition: make the pendulum behave like a monotonic, but otherwise possibly nonlinear, damped spring. Define the controller for the Balance region to be:

$$\begin{aligned} u(\phi, \dot{\phi}) &= g_b(\phi) \\ [g_b(\phi) - g(\phi)]_0 &= [\phi]_0 \end{aligned} \quad (3.40)$$

Since  $g(\phi) = k \sin \phi$  increases monotonically with  $\phi$  over  $[-\frac{\pi}{2}, \frac{\pi}{2}]$ ,  $g_b(\phi)$  must increase at least as fast in order to ensure that  $[g_b(\phi) - g(\phi)]_0 = [\phi]_0$ . We can get faster convergence by augmenting the natural damping  $f(\dot{\phi})$  with a damping term  $h(\dot{\phi})$  included in the control law, giving us

$$\begin{aligned} u(\phi, \dot{\phi}) &= g_b(\phi) + f_b(\dot{\phi}) \\ [g_b(\phi) - g(\phi)]_0 &= [\phi]_0 \\ [f_b(\dot{\phi})]_0 &= [\dot{\phi}]_0 \end{aligned} \quad (3.41)$$

If there is a bound on the control action  $u < u_{\max}$ , then the limiting angle  $\phi_{\max}$  beyond which the controller cannot restore the pendulum to  $\phi = 0$  is given by the constraint

$$u_{\max} = g(\phi_{\max}) = k \sin \phi_{\max} \quad (3.42)$$

The maximum velocity  $\dot{\phi}_{\max}$  that the Balance controller can tolerate at  $\phi = 0$  is then determined by the constraint

$$\frac{1}{2}\dot{\phi}_{\max}^2 = \int_0^{\phi_{\max}} g_b(\phi) - g(\phi)d\phi \quad (3.43)$$

which represents the conversion of the kinetic energy of the system (equation 3.39) at  $\langle 0, \dot{\phi}_{\max} \rangle$  into potential energy at  $\langle \phi_{\max}, 0 \rangle$ .

We would like to define the boundary of the Balance region as a level curve of the Lyapunov function for the controlled system (equation 3.39), from Lemma 3.3.1,

$$V(\phi, \dot{\phi}) = \frac{1}{2}\dot{\phi}^2 + \int_0^{\phi} g_b(\phi) - g(\phi)d\phi \quad (3.44)$$

It is easy to check that  $V(\phi_{\max}, 0) = V(0, \dot{\phi}_{\max})$ , so their intercepts lie on the same level curve of  $V$ . When a trajectory intersects this level curve,  $V(\phi, \dot{\phi}) = \frac{1}{2}\dot{\phi}_{\max}^2$ . Expanding and dividing by  $\frac{1}{2}\dot{\phi}_{\max}^2$ , we get

$$\frac{\dot{\phi}^2}{\dot{\phi}_{\max}^2} + \frac{\int_0^{\phi} g_b(\phi) - g(\phi)d\phi}{\frac{1}{2}\dot{\phi}_{\max}^2} = 1 \quad (3.45)$$

Substituting the definition of  $\dot{\phi}_{\max}^2$  (equation 3.43) into the second term gives

$$\frac{\dot{\phi}^2}{\dot{\phi}_{\max}^2} + \frac{\int_0^{\phi} g_b(\phi) - g(\phi)d\phi}{\int_0^{\phi_{\max}} g_b(\phi) - g(\phi)d\phi} = 1 \quad (3.46)$$

Because  $[g_b(\phi) - g(\phi)]_0 = [\phi]_0$ , we know that both the integrals are non-negative, so equation (equation 3.46) defines an “ellipse-like” curve that intersects the axes at  $\pm\phi_{\max}$  and  $\pm\dot{\phi}_{\max}$ . Furthermore, the curve changes monotonically between the intercepts.

In the special case where  $g_b(\phi) - g(\phi) \cong \phi$ , we can evaluate the integrals and show that the level curve of  $V$  is an ellipse:

$$\frac{\dot{\phi}^2}{\dot{\phi}_{\max}^2} + \frac{\phi^2}{\phi_{\max}^2} = 1 \quad (3.47)$$

Note that the shapes of the non-linear functions  $g$  and  $h$  are only very weakly constrained. The qualitative constraints in (equation 3.41) provide weak sufficient conditions

guaranteeing the stability of the inverted pendulum controller. However, there is plenty of freedom available to the designer to select the properties of  $g_b$  and  $f_b$  to optimize any desired criterion.

### 3.4.2 Escaping the Elliptic Fixed Point: Pump Mode

For any point lying within the libration region of the pendulum phase space (see figure 3.2), the goal is to pump energy into the pendulum, swinging it progressively higher, until it reaches the region where the inverted pendulum controller can balance it in the upright position.

The QDE describing motion is similar to equation 3.39, but is written in terms of  $\theta, \dot{\theta}$ ,

$$\ddot{\theta} + f(\dot{\theta}) + g(\theta) + u(\theta, \dot{\theta}) = 0. \quad (3.48)$$

Without control action, since  $[\sin \theta]_0 = [\theta]_0$  over  $-\pi < \theta < \pi$ , the model exactly matches the monotonic damped spring model of Lemma 3.3.1, so we know that it is asymptotically stable at  $(\theta, \dot{\theta}) = (0, 0)$ . Unfortunately, this is not where we want it.

Instead, Lemma 3.3.2 gives us a sufficient condition to transform the stable fixed point into an unstable repellor. Define the controller for the Pump region so that the system is modeled by a spring with negative damping, pumping energy into the system. That is,

$$\begin{aligned} u(\theta, \dot{\theta}) &= -f_p(\dot{\theta}) \\ f_p - f &\in P_0^+ \end{aligned} \quad (3.49)$$

Starting with an infinitesimal perturbation from the elliptic fixed point, or from any point inside the libration region, this controller will pump the pendulum to higher and higher swings. Lemma 3.3.2 is sufficient to assure us that there are no limit cycles in the region  $-\pi < \theta < \pi$  to prevent the trajectory from approaching  $\theta = \pi$  so the Balance control law can stabilize it.

### 3.4.3 Handling High Energy Rotations: Spin Mode

For any point lying within the rotation region of the pendulum phase space (see figure 3.2), the goal is to slow down the rotations - either by allowing friction to work its course, or



with additional damping. For this, define

$$\begin{aligned} u(\theta, \dot{\theta}) &= f_s(\dot{\theta}) \\ f_s &\in P_0^+ \end{aligned} \quad (3.50)$$

### 3.4.4 Ensuring Correct Mode Transitions: Separatrix as a Sliding Mode

The above arguments ensure that all trajectories beginning in the Libration region will eventually reach the separatrix, as do all trajectories beginning in the rotation region. The key point that needs to be verified to ensure task achievement is that these crossings are not just wild oscillations but in fact constitute progress towards the desired goal.

It turns out that the choices made so far are sufficient to endow the separatrix trajectory with the properties of a *sliding mode* [60] between the Pump and Spin modes.

All points on the separatrix trajectory have the total system energy,

$$KE + PE = \frac{1}{2}\dot{\theta}^2 + \int_0^\theta k \sin \theta d\theta = 2k \quad (3.51)$$

Evaluating the integral and simplifying, we can represent the separatrix as an equation of the form  $s(\theta, \dot{\theta}) = 0$ , with the Spin region corresponding to  $s > 0$  and Pump to  $s < 0$ .

$$s(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^2 - k(1 + \cos \theta) = 0 \quad (3.52)$$

Differentiating (equation 3.52) and substituting for  $\ddot{\theta}$ , we get

$$\begin{aligned} \dot{s} &= \dot{\theta}\ddot{\theta} + k \sin \theta \dot{\theta} \\ \dot{s} &= \dot{\theta}(-f(\dot{\theta}) - k \sin \theta - u(\theta, \dot{\theta})) + k \sin \theta \dot{\theta} \\ \dot{s} &= -\dot{\theta}f(\dot{\theta}) - \dot{\theta}u(\theta, \dot{\theta}) \end{aligned} \quad (3.53)$$

Now, examine the Pump region, inside the separatrix where  $s < 0$ , and substitute the Pump control law (equation 3.49).

$$\begin{aligned} \dot{s}_{pump} &= -\dot{\theta}f(\dot{\theta}) + \dot{\theta}f_p(\dot{\theta}), f_p - f \in P_0^+ \\ \dot{s}_{pump} &= \dot{\theta}(f_p - f)(\dot{\theta}) \geq 0 \end{aligned} \quad (3.54)$$

Similarly for Spin (equation 3.50),

$$\begin{aligned}\dot{s}_{spin} &= -\dot{\theta}f(\dot{\theta}) - \dot{\theta}f_s(\dot{\theta}), f_s \in P_0^+ \\ \dot{s}_{spin} &= \dot{\theta}(f + f_s)(\dot{\theta}) \leq 0\end{aligned}\tag{3.55}$$

This establishes that all trajectories, no matter where they begin will eventually reach the goal and stabilize there. Once there, arbitrarily large disturbances may have the effect of sending the system to a different point in phase space, but the subsequent trajectory is guaranteed to reach the goal. The behavior of the resulting controller can be summarized in the form of the following result.

**Theorem 3.4.1** *Any pendulum system of the qualitative form of equation 3.39, subject to a hybrid control strategy which involves the control functions defined by equations 3.41, 3.49, 3.50, the constraint defined by equation 3.46 and the switching logic,*

*If  $\alpha \leq 1$ , Then Balance*

*Else If  $s < 0$ , Then Pump*

*Else Spin*

where  $\alpha = \frac{\int_0^\phi g_b(\phi) - g(\phi) d\phi}{\int_0^{\phi_{\max}} g_b(\phi) - g(\phi) d\phi} + \frac{\dot{\phi}^2}{\dot{\phi}_{\max}^2}$  and  $s = \frac{1}{2}\dot{\phi}^2 - k(1 - \cos \phi)$ , is globally asymptotically stable at  $\phi = 0, \dot{\phi} = 0$ .

**Proof** The proof follows from the following assertions that have been established in the above arguments:

- For any trajectory beginning in the libration region,  $\dot{s}(\theta, \dot{\theta}) > 0$ . Therefore, the maximum swing amplitude  $\theta_{max}|_{\dot{\theta}=0}$  must increase. This value must continue to increase as  $\theta_{max} \rightarrow \pm\pi$  until one of two events occur - either the point  $(\theta_{max}, 0)$  lies inside the region of applicability of the Balance controller, or the value of  $s$  reaches that of the separatrix and the trajectory has reached the sliding mode.
- Similarly, for any trajectory beginning in the rotation region,  $\dot{s}(\theta, \dot{\theta}) < 0$ . Therefore, the swing velocity  $\dot{\theta}_{\theta=\pm\pi}$  must decrease. This value must continue to decrease until one of two events occur - either the point  $(\theta, \dot{\theta})$  lies inside the region of applicability of the Balance controller, or the value of  $s$  reaches that of the separatrix and the trajectory has reached the sliding mode.

- Any trajectory reaching the sliding mode, i.e., the separatrix, will *chatter* in a suitably small neighborhood until the trajectory is within the region of applicability of the Balance controller.
- Any trajectory inside or at the boundary of the Balance region will asymptotically stabilize at the hyperbolic fixed point,  $\phi = 0, \dot{\phi} = 0$ .

These assertions are summarized in figure 3.3. ■

**Remark** In fact, there exists one singular point that does not fall in the basin of attraction of  $\phi = 0, \dot{\phi} = 0$  and this is the elliptic fixed point,  $\theta = 0, \dot{\theta} = 0$ . So strictly speaking, this theorem only guarantees *almost* global asymptotic stability. However, in practice, even an infinitesimal perturbation away from this point is sufficient to bring the system into the desired basin and so this is not an issue.

Figure 3.4 depicts the operation of this control strategy as a very weak controller pumps the pendulum up from  $\theta \simeq 0$  and balances it at  $\phi = 0, \dot{\phi} = 0$ . We define an instance of the pendulum model and the local control laws:

**Plant:**  $\ddot{\phi} + c\dot{\phi} - k \sin \phi + u(\phi, \dot{\phi}) = 0 \quad c = 0.01, k = 10, u_{\max} = 4$

**Balance Control:**  $u = (c_{11} + k)\phi + c_{12}\dot{\phi} \quad c_{11} = 0.4, c_{12} = 0.3$

**Spin Control:**  $u = c_2\dot{\phi} \quad c_2 = 0.5$

**Pump Control:**  $u = -(c + c_3)\dot{\phi} \quad c_3 = 0.5$

The plant model is chosen with normal gravity, slight friction, and a maximum control action too weak to lift the pendulum directly up. The local control laws are all linear for simplicity, although they could be designed to be nonlinear. The desired behavior is achieved as long as the parameters  $c_i$  are all positive. Given the maximum control action  $u_{\max}$  and the gain  $c_{11} + k$  of the Balance controller, we can determine the bounds  $\phi_{\max} = 4$  and  $\dot{\phi}_{\max} = 0.3$  for the Balance region from equations (3.42) and (3.43), respectively.

In the above example, specific values for  $c_i$  were simply chosen by hand from the feasible intervals. In section 3.6, we discuss the use of numerical optimization to automate this controller synthesis step.

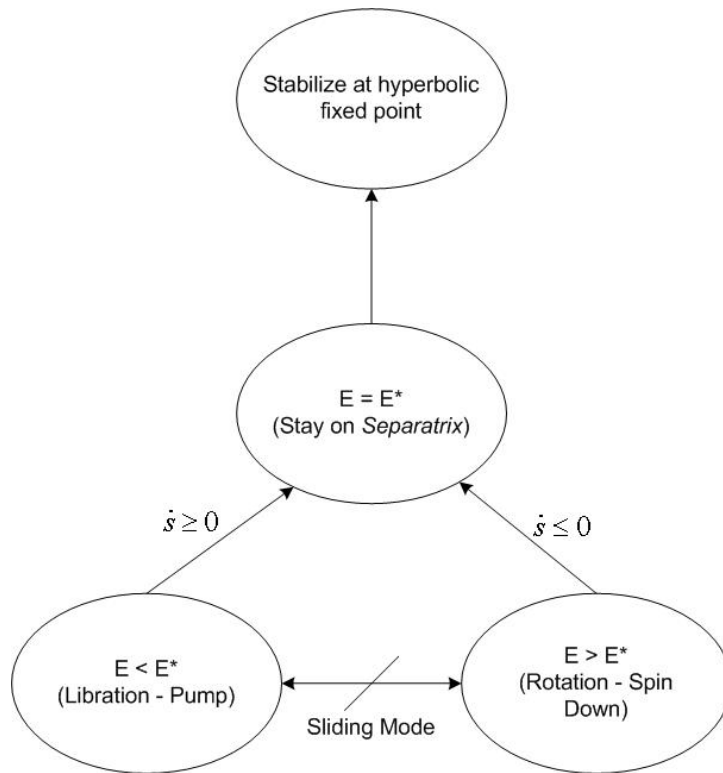


Figure 3.3: Discrete state transition behavior of the pendulum control strategy of section 3.4

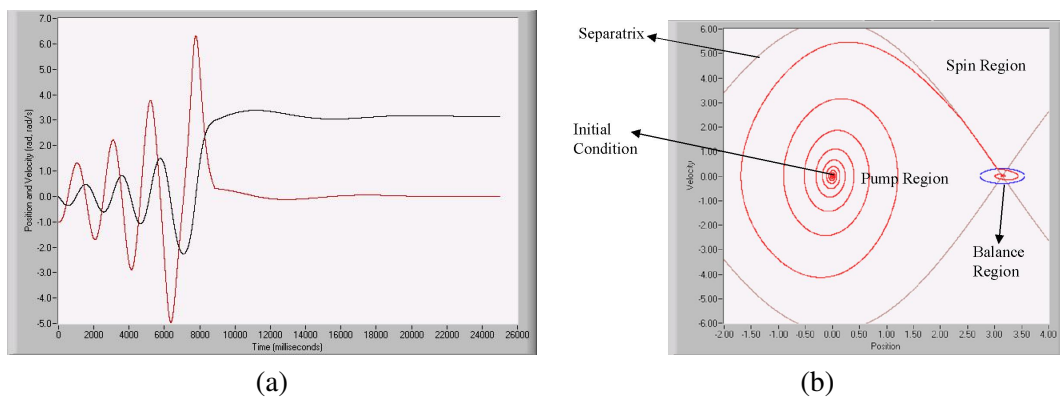


Figure 3.4: Typical trajectories corresponding to the example in section 3.4.4. Part *a* depicts the time domain trajectories of  $\theta(t)$  and  $\dot{\theta}(t)$ , while part *b* depicts the phase space.

### 3.5 A Global Control Strategy for the Cart-Pole System

Having understood the issues associated with global control of the pivot-actuated pendulum, we are ready to consider its higher dimensional analog - the pendulum on a cart. This system is depicted in figure 3.5. It consists of a cart that moves on a horizontal track of finite length. The pendulum, i.e., a pole, is represented by a point mass attached to the end of a massless thin rod of length  $l$  that is attached to the cart at a pivot capable of complete  $360^\circ$  rotation. It is worth noting that this system is underactuated in that there is only one control, the force applied to the cart, that must be used to simultaneously stabilize the cart and the pole.

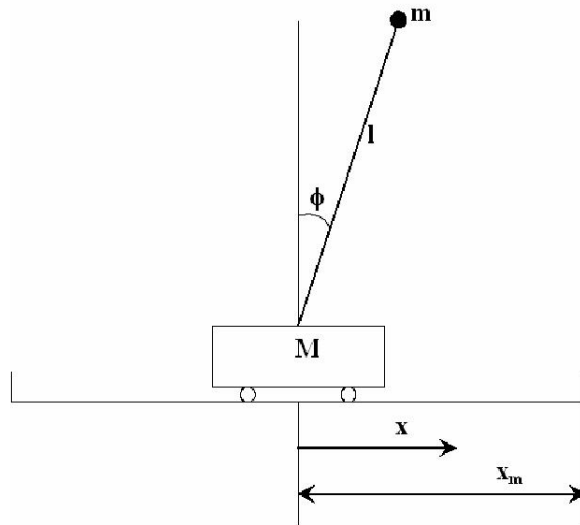


Figure 3.5: The cart-pole system. The cart runs on a finite track of length  $x_m$  while the pole is capable of full  $360^\circ$  rotation.

The objective is to stabilize this system at  $[x, \dot{x}, \phi, \dot{\phi}] = [0, 0, 0, 0]$ , starting from  $[0, 0, \pi, 0]$ , despite arbitrarily large disturbances to the pole (such as an adversary who delivers arbitrarily large impacts to the pole). As before, this global control must be achieved using bounded forces on the cart, and eventually with bounds on the track length. The strategy adopted here is to extend the strategy defined in section 3.4 to this higher dimensional system.

The dynamic model for the cart-pole system is given by,

$$(M + m)\ddot{x} + ml \cos \phi \ddot{\phi} - ml \sin \phi \dot{\phi}^2 = F - f_c(\dot{x}) \quad (3.56)$$

$$ml \cos \phi \ddot{x} + ml^2 \ddot{\phi} - mgl \sin \phi = -f_p(\dot{\phi}) \quad (3.57)$$

where  $x, \phi$  represent the cart position and pole angle respectively,  $F$  is a force applied to the cart and  $f_c, f_p$  are frictional terms. In fact,  $F$  can be suitably chosen such that arbitrary  $\ddot{x}$  (within a suitably small bounded interval) can be achieved and the problem can be reduced to that of controlling the system in equation 3.57. This equation can be expressed as the following QDE. There are two versions of the equation, one around  $\phi = 0$  and the other around  $\theta = 0$ .

$$\ddot{\phi} + f(\dot{\phi}) - k \sin \phi + \ddot{x} \cos \phi = 0 \quad (3.58)$$

$$\ddot{\theta} + f(\dot{\theta}) + k \sin \theta - \ddot{x} \cos \theta = 0 \quad (3.59)$$

### 3.5.1 Using Time-Scale Separation to Deal with Underactuation

Although we are dealing with a 4-dim system, in terms of  $(\theta, \dot{\theta}, x, \dot{x})$ , there are clearly two subsystems here and the task can be decomposed into one of global control of the pendulum, much as in section 3.4, and stabilization of the cart. The complexity in the dynamics of the cart is derived from the nonlinear dynamics of the pendulum.

Any realization of  $\ddot{x}(t)$  that satisfies the conditions of theorem 3.4.1 is sufficient to stabilize the pole at  $(\phi = 0, \dot{\phi} = 0)$ . The goal of the design then becomes to find a suitable  $\ddot{x}(t)$  within this qualitatively constrained space that can also stabilize the cart at  $(x = 0, \dot{x} = 0)$ .

Consider a control action of the form,

$$\ddot{x} = \text{varsat} \left\{ -\frac{f_c(\dot{x})}{\cos \phi} - \frac{g_c(x)}{\cos \phi} + \frac{u(\phi, \dot{\phi})}{\cos \phi} \right\} \quad (3.60)$$

where  $f_c(\dot{x}), g_c(x)$  represent any reasonable functions satisfying Lemma 3.3.1, used to regulate the position and velocity of the cart. The term  $u(\phi, \dot{\phi})$  represents the control law for the three control modes of the pole from section 3.4. *varsat* refers to a saturation action whose magnitude depends on the controller mode. This allows the designer to set local control law parameters in such a way that the system never saturates in Balance while the

Pump and Spin control actions can be restricted in strength.

$$\begin{aligned} \text{varsat}(x) &= \begin{cases} \text{sat}(x, x_{\max \text{ bal}}), \text{controller} = \text{Balance} \\ \text{sat}(x, x_{\max \text{ pump}}), \text{controller} = \text{Pump} \\ \text{sat}(x, x_{\max \text{ spin}}), \text{controller} = \text{Spin} \end{cases}, & (3.61) \\ \text{sat}(x, x_{\max i}) &= \begin{cases} -x_{\max i}, x < -x_{\max i} \\ x, -x_{\max i} \leq x \leq x_{\max i} \\ x_{\max i}, x > x_{\max i} \end{cases} \end{aligned}$$

Let us now derive the constraints on this cart control action, corresponding to Pump and Spin modes of the pole controller. Taking the derivative of the expression for  $s$ , expressed in terms of the variables  $\phi$  and  $\dot{\phi}$ ,

$$\dot{s}(\phi, \dot{\phi}) = -\dot{\phi}f(\dot{\phi}) + \dot{\phi}f_c(\dot{x}) + \dot{\phi}g_c(x) - \dot{\phi}u(\phi, \dot{\phi}) \quad (3.62)$$

Substitute the Pump control law,

$$\dot{s} = \dot{\phi}(f_p - f)(\dot{\phi}) + \dot{\phi}f_c(\dot{x}) + \dot{\phi}g_c(x) \quad (3.63)$$

In order to have  $\dot{s} \geq 0$ ,

$$[(f_p - f)(\dot{\phi}) + f_c(\dot{x}) + g_c(x)]_0 = [\dot{\phi}]_0 \quad (3.64)$$

If we substitute the Spin control law,

$$\dot{s} = -\dot{\phi}(f + f_s)(\dot{\phi}) + \dot{\phi}f_c(\dot{x}) + \dot{\phi}g_c(x) \quad (3.65)$$

In order to have  $\dot{s} \leq 0$

$$[(f + f_s)(\dot{\phi}) - f_c(\dot{x}) - g_c(x)]_0 = [\dot{\phi}]_0 \quad (3.66)$$

We know that  $[(f_s - f)(\dot{\phi})]_0 = [\dot{\phi}]_0$  and  $[(f + f_d)(\dot{\phi})]_0 = [\dot{\phi}]_0$ . From equation 3.64 and equation 3.66, we see that  $f_c(\dot{x}) + g_c(x)$  needs to be sufficiently small with respect to  $(f_p - f)(\dot{\phi})$ ,  $(f + f_s)(\dot{\phi})$  for the sign equality to be preserved (even if these terms are opposed to each other). It is conceivable that for small values  $\dot{\phi} \rightarrow 0$ , these constraints may be violated for  $f_c(\dot{x}) + g_c(x) \neq 0$ . Fortunately, the sliding mode constrains the trajectory

to be on the separatrix where it is bounded away from  $\dot{\phi} = 0$  by the width of the Balance region. Therefore, it is possible to select  $f_c$  and  $g_c$  to be sufficiently small to satisfy equation 3.64 and equation 3.66. This amounts to requiring a sufficient separation between the time-scales of the pole and the cart controllers in order to ensure that the separatrix continues to act as a sliding mode even for this augmented system.

The parametrization in equation 3.60 uses an unbounded control action (due to the  $\frac{1}{\cos \phi}$  term) that is then bounded by a nonlinearity. This enables us to present the connection between the cart-pole controller and the pendulum controller in a straight forward way, in terms of the above constraints that ensure appropriate  $[\dot{s}]_0$ .

One could also adopt a smoother control strategy as shown below.

Define the cart-pole controller as,

$$\ddot{x}(\phi, \dot{\phi}, x, \dot{x}) = -f_c(\dot{x}) - g_c(x) + \frac{u(\phi, \dot{\phi})}{\cos \phi} \quad (3.67)$$

The term  $u(\phi, \dot{\phi})$  is for pole control,  $f_c(\dot{x})$  and  $g_c(x)$  are reasonable functions in  $M_0^+$ .

The Balance control component of  $u(\phi, \dot{\phi})$  is defined exactly as before, equation 3.41. Given a maximum cart acceleration of  $a_{\max}$ , the maximum pole angle in Balance is defined as,

$$a_{\max} = k \tan(\phi_{\max}) \quad (3.68)$$

from which the maximum  $\dot{\phi}$  can be computed as before, following equation 3.46.

The Spin controller is modified to take the following form,

$$u_s(\phi, \dot{\phi}) = f_s(\dot{\phi})h_s(\phi), \text{ where } \left| \frac{h_s(\phi)}{\cos \phi} \right| \leq 1 \quad (3.69)$$

for a  $h_s(\phi) \geq 0$  with only isolated zeros, and  $h_s(0) > 0$ . One such function is  $h_s(\phi) = \cos^2 \phi$ .

Similarly, for Pump,

$$u_p(\theta, \dot{\theta}) = -f_p(\dot{\theta})h_p(\theta), \text{ where } \left| \frac{h_p(\theta)}{\cos \theta} \right| \leq 1 \quad (3.70)$$

where  $h_p(\theta) \geq 0$  with only isolated zeros, and  $h_p(0) > 0$ .

With these choices, the pendulum sliding mode is also preserved by the controller



in equation 3.67 as shown below. Begin with the expression,

$$\dot{s}(\phi, \dot{\phi}) = -\dot{\phi}f(\dot{\phi}) + \dot{\phi}f_c(\dot{x}) \cos \phi + \dot{\phi}g_c(x) \cos \phi - \dot{\phi}u(\phi, \dot{\phi}) \quad (3.71)$$

With the Pump control law as in equation 3.70,

$$\dot{s}(\phi, \dot{\phi}) = \dot{\phi}(f_p(\dot{\phi})h_p(\phi) - f(\dot{\phi})) + \dot{\phi}(f_c(\dot{x}) + g_c(x)) \cos \phi \quad (3.72)$$

Over a single cycle  $[t_1, t_2]$ , the change in energy is in the desired direction:

$$\begin{aligned} 0 &\leq s(t_2) - s(t_1) \\ &= \int_{t_1}^{t_2} \dot{s} dt = \int_{t_1}^{t_2} \dot{\phi} \left( f_p(\dot{\phi}) h_p(\phi) - f(\dot{\phi}) \right) dt + \int_{t_1}^{t_2} \dot{\phi} (f_c(\dot{x}) + g_c(x)) \cos \phi dt \end{aligned} \quad (3.73)$$

Over a single cycle, due to symmetry of the oscillation,

$$\int_{t_1}^{t_2} \dot{\phi} (f_c(\dot{x}) + g_c(x)) \cos \phi dt \approx 0 \quad (3.74)$$

We then derive the constraint,

$$\int_{t_1}^{t_2} \dot{\phi} \left( f_p(\dot{\phi}) h_p(\phi) - f(\dot{\phi}) \right) dt > 0 \quad (3.75)$$

Similarly, with the Spin control law as in equation 3.69,

$$\dot{s}(\phi, \dot{\phi}) = \dot{\phi}(f_p(\dot{\phi})h_p(\phi) - f(\dot{\phi})) + \dot{\phi}(f_c(\dot{x}) + g_c(x)) \cos \phi \quad (3.76)$$

Over a cycle, the change in the energy is negative (toward the separatrix) if

$$\begin{aligned} 0 &\geq s(t_2) - s(t_1) \\ &= \int_{t_1}^{t_2} \dot{s} dt = \int_{t_1}^{t_2} -\dot{\phi} \left( f_s(\dot{\phi}) h_s(\phi) - f(\dot{\phi}) \right) dt + \int_{t_1}^{t_2} \dot{\phi} (f_c(\dot{x}) + g_c(x)) \cos \phi dt \end{aligned} \quad (3.77)$$

Again, the latter term is negligible and this constraint effectively amounts to,

$$\int_{t_1}^{t_2} \dot{\phi}(f_s(\dot{\phi})h_s(\phi) - f(\dot{\phi}))dt \leq 0 \quad (3.78)$$

$$f, f_s \in P_0^+, h_s \geq 0$$

The controlled cart system is represented as,

$$\ddot{x} + f_c(\dot{x}) + g_c(x) = \frac{u(\phi, \dot{\phi})}{\cos \phi} \quad (3.79)$$

Our next task is to ensure that the driving term does not disrupt the stability of the damped spring model for the cart.

In the Balance region, the cart subsystem is described by

$$\ddot{x} = -g_c(x) - f_c(\dot{x}) + \frac{g_b(\phi) + f_b(\dot{\phi})}{\cos \phi} \quad (3.80)$$

In order to show that the cart system converges to  $(x, \dot{x}) = (0, 0)$  in spite of the driving influence of the pole system, we appeal to Corollary 3.3.8 of Lemma 3.3.6. Consider the Poincaré section of the points where the trajectory of the cart intersects the line  $\dot{x} = 0$ .

$$\int_{x_1}^{x_2} g_c(x)dx = - \int_{t_1}^{t_2} f_c(\dot{x})\dot{x}dt + \int_{t_1}^{t_2} \frac{g_b(\phi) + f_b(\dot{\phi})}{\cos \phi} \dot{x}dt \quad (3.81)$$

We will show that successive extremal points  $x_i$  of the cart's position approach the origin. Assuming an unbounded track, and selecting extrema  $x_1, x_2$  on the same side of the track's origin, we wish to show that the left hand side of the above equation is negative, implying that  $|x_2| < |x_1|$ , since  $g_c \in P_0^+$ .

Since  $f_c \in P_0^+$ , we know that

$$- \int_{t_1}^{t_2} f_c(\dot{x})\dot{x}dt \leq 0$$

We can guarantee correct signs of equation 3.81 by requiring that,

$$\left| \int_{t_1}^{t_2} \frac{g_b(\phi) + f_b(\dot{\phi})}{\cos \phi} \dot{x} dt \right| < \int_{t_1}^{t_2} f_c(\dot{x}) \dot{x} dt \quad (3.82)$$

We now consider the cart-pole controller specialized for the case when pole is in the Spin region:

$$\ddot{x} = -g_c(x) - f_c(\dot{x}) + f_s(\dot{\phi}) \frac{h_s(\phi)}{\cos \phi} \quad (3.83)$$

where  $[f_s(\dot{\phi})]_0 = [\dot{\phi}]_0$ . As before, we consider the Poincaré section consisting of points where  $\dot{x} = 0$ ,

$$\int_{x_1}^{x_2} g_c(x) dx = - \int_{t_1}^{t_2} f_c(\dot{x}) \dot{x} dt + \int_{t_1}^{t_2} f_s(\dot{\phi}) \frac{h_s(\phi)}{\cos \phi} \dot{x} dt \quad (3.84)$$

We can show that the left hand side of this equation is negative, implying that successive extrema approach the origin by requiring that,

$$\left| \int_{t_1}^{t_2} f_s(\dot{\phi}) \frac{h_s(\phi)}{\cos \phi} \dot{x} dt \right| < \int_{t_1}^{t_2} f_c(\dot{x}) \dot{x} dt \quad (3.85)$$

One way to achieve this is to define  $h_s(\phi)$  to be small over most of the cycle, e.g.,  $h_s(\phi) = |\cos^n \phi|$  for  $n \geq 2$ .

The cart system under the Pump controller becomes,

$$\ddot{x} = -g_c(x) - f_c(\dot{x}) + f_p(\dot{\theta}) \frac{h_p(\theta)}{\cos \theta} \quad (3.86)$$

As before, we derive,

$$\int_{x_1}^{x_2} g_c(x) dx = - \int_{t_1}^{t_2} f_c(\dot{x}) \dot{x} dt + \int_{t_1}^{t_2} f_p(\dot{\theta}) \frac{h_p(\theta)}{\cos \theta} \dot{x} dt \quad (3.87)$$

$$\left| \int_{t_1}^{t_2} f_p(\dot{\theta}) \frac{h_p(\theta)}{\cos \theta} \dot{x} dt \right| < \int_{t_1}^{t_2} f_c(\dot{x}) \dot{x} dt \quad (3.88)$$

In the Pump region, we face two competing objectives. Firstly,  $h_p$  must be restricted relative to the damping friction term  $f_c$ , to avoid making the cart unstable. On the other

hand, making it too restrictive would imply that energy loss may not be overcome, resulting in an ineffective Pump controller. This is a tradeoff that needs to be considered in the optimization process.

These arguments may be summarized in the following result.

**Theorem 3.5.1** *The controlled cart-pole system specified by,*

*a controller of the form of equation 3.67*

*with pole controller function  $u(\phi, \dot{\phi})$  specified by equations 3.41, 3.69, 3.70*

*subject to the constraints specified in equations 3.68, 3.75, 3.78, 3.82, 3.85, 3.88*

*is globally asymptotically stable at  $\phi = 0, \dot{\phi} = 0, x = 0, \dot{x} = 0$ .*

**Proof** Let us define the set  $\Lambda$  of states  $\mathcal{X} = \{\phi, \dot{\phi}, x, \dot{x}\}$  (with  $x \in \mathfrak{R}, \phi \in S^1$ ) as an inner approximation of the invariant set with equilibrium point  $(\phi = 0, \dot{\phi} = 0, x = 0, \dot{x} = 0)$ .

Let us define the set  $\Lambda_{\phi \sim 0}$  as the set of  $\mathcal{X} = \{\phi, \dot{\phi}, x, \dot{x}\}$  which is an invariant neighborhood of  $(\phi = 0, \dot{\phi} = 0)$ , irrespective of the states  $(x, \dot{x})$ . Similarly, define the set  $\Lambda_{x \sim 0}$  as the set of  $\mathcal{X} = \{\phi, \dot{\phi}, x, \dot{x}\}$  which is an invariant neighborhood of  $(x = 0, \dot{x} = 0)$  irrespective of the states  $(\phi, \dot{\phi})$ .

The set of all points  $(\phi, \dot{\phi})$  inside the Balance region for the pole (based on equation 3.68) is a suitable instance of  $\Lambda_{\phi \sim 0}$ . However, realize that this is not the same as  $\Lambda$  because it is possible to design strategies that balance the pole without stabilizing the cart at the desired equilibrium.

Therefore, we wish to ensure two things:

- a. The term  $u(\phi, \dot{\phi})$  is sufficient to ensure convergence to  $\Lambda_{\phi \sim 0}$ , despite the action of the cart-component  $(-f_c(\dot{x}) - g_c(x))$  in the controller 3.67.
- b. The cart-component  $(-f_c(\dot{x}) - g_c(x))$  is sufficient to ensure convergence to  $\Lambda_{x \sim 0}$ , despite the action of  $u(\phi, \dot{\phi})$ .

Satisfaction of the constraint inequalities 3.68, 3.75, 3.78 ensures that statement (a) is true. Similarly, satisfaction of the constraint inequalities 3.82, 3.85, 3.88 ensures that statement (b) is true.

Therefore, eventually,  $\mathcal{X} \in \Lambda_{\phi \sim 0} \cap \Lambda_{x \sim 0} = \Lambda$ . This yields the desired result - global asymptotic stability at  $\phi = 0, \dot{\phi} = 0, x = 0, \dot{x} = 0$ . ■

The above result characterizes the condition for global asymptotic stability of the cart-pole system. A key feature of this approach is that it enables separation between the two components and allows us to implement a simple and intuitive strategy designed for a subsystem, the pendulum, using the more complex underactuated system. This represents a simple example of the use of the template-anchor decomposition.

However, many of the functional inequalities derived above do not support direct computation of parameter ranges for the controllers. Nonetheless, they do specify a class of qualitatively constrained functions from which a suitable controller may be synthesized. Section 3.6 will address this issue further.

### 3.5.2 Handling Nonsmooth Nonlinearities

#### Static Friction

Most robotic systems involve joints and drives with static friction. The effect of static friction is that there is a dead zone where small actuation forces have no effect until some threshold is crossed and the system jerks into motion. The standard way to deal with such systems in positioning applications is to estimate the effect of static friction and provide an extra impulse prior to entering the deadzone. An alternate approach, that we explore primarily to demonstrate an important behavioral mode, is to implement a periodic orbit around the dead zone thus avoiding that regime entirely.

Consider the damped oscillator with static, or Coulomb, friction  $F_c$  as well as damping friction  $f(\dot{x})$ . The resistance of static friction is a constant force  $F_c = \eta$  opposing the direction of motion, as long as motion is taking place. Once motion stops, a larger frictional resistance  $F_c = \eta + \varepsilon$  must be overcome to get it started again. This transforms the damped oscillator model from  $\ddot{x} = -g(x) - f(\dot{x}) - F_c$  to

$$\ddot{x} = \begin{cases} \dot{x} \neq 0 & \rightarrow -g(x) - f(\dot{x}) - [\dot{x}]_0 \eta \\ \dot{x} = 0 & \rightarrow -[x]_0 \max(|g(x)| - \eta - \varepsilon, 0) \end{cases} \quad (3.89)$$

To avoid being captured by static friction, we require that the system obey the constraint

$$\dot{x} = 0 \Rightarrow |g(x)| > \eta + \varepsilon \quad (3.90)$$

This constraint excludes the stable fixed-point, where  $x = \dot{x} = \ddot{x} = 0$ . Therefore, we will be required to approach a more complex orbit, such as a limit cycle.

A limit cycle can be enforced by transforming the pendulum system into the form of a Lienard equation (e.g., the van der Pol oscillator) [85, 86]. This system is defined in terms of the model  $\ddot{x} + f(x)\dot{x} + g(x) = 0$  where, broadly speaking,  $f(x)$  is positive when  $|x|$  is large and negative when  $|x|$  is small, and  $g$  is such that, in the absence of the damping term  $f(x)\dot{x}$  we expect periodic solutions for small  $x$ . The property was summarized in Lemma 3.3.5, in section 3.3.

We utilize this lemma and define a switched system inside Balance. Starting with the system,

$$\ddot{\phi} + c\dot{\phi} - k \sin \phi + \ddot{x} \cos \phi = 0 \quad (3.91)$$

In the region,  $-a < \phi < a$ , we define the controller,

$$\ddot{x} = \frac{(k + p_2)\phi - (c + p_1 - \phi^2)\dot{\phi}}{\cos \phi} \quad (3.92)$$

so that the closed loop system takes the form (assume  $\cos \phi = 1, \sin \phi = \phi$ ),

$$\ddot{\phi} + (\phi^2 - p_1)\dot{\phi} + p_2\phi = 0 \quad (3.93)$$

In the region  $(-\phi_{\max} < \phi < -a) \cup (a < \phi < \phi_{\max})$ , the controlled system is defined as,

$$\begin{aligned} \ddot{x} &= \frac{(c_{11}+k)\phi+(c_{12}+\phi^2)\dot{\phi}}{\cos \phi} \\ \ddot{\phi} + (\phi^2 + p_1)\dot{\phi} + p_2\phi &= 0 \end{aligned} \quad (3.94)$$

Setting  $(c + c_{12}) = p_1 > a^2; c_{11} = p_2$  satisfies the conditions of Lemma 3.3.5.

The effect of including this region is that the cart-pole system executes a limit cycle as seen in figure 3.6.

### Finite Track Length

The analysis in section 3.5.1 utilized a linear spring model for cart control. In a system with a finite track, it may in fact be preferable to use a nonlinear spring controller that avoids the limit stops. The nonlinear spring controller provides the advantage that the cart position can be kept away from the ends of the track, i.e.,  $|x| < x_m$ , by the potential barrier of the

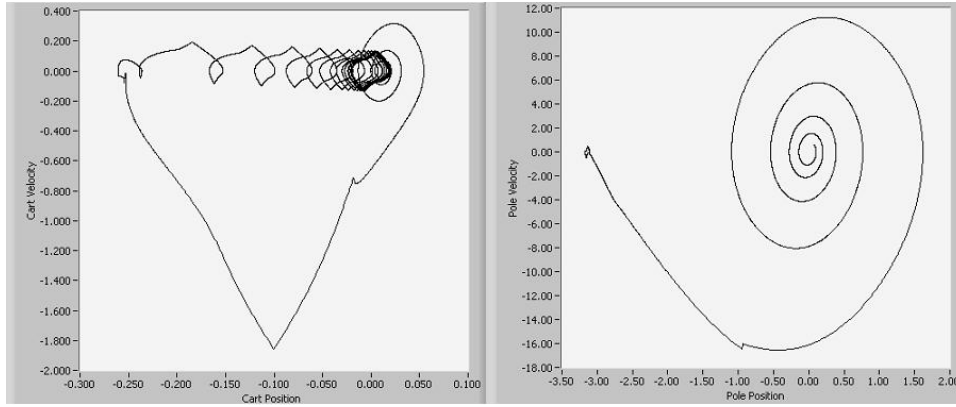


Figure 3.6: Phase plots for the cart  $x - \dot{x}$  ( $m, m/s$ ) and pole  $\theta - \dot{\theta}$  ( $rad, rad/s$ ) subsystems - simulation results. These plots illustrate the system executing a limit cycle in both the cart and pole subsystems.

spring action. Consider an instance of a nonlinear spring defined by,

$$\ddot{x} = sat \left\{ -\frac{d_1}{\cos \phi} \tanh^{-1} \left( \frac{x}{x_m} \right) - \frac{d_2}{\cos \phi} \dot{x} + \frac{u(\phi, \dot{\phi})}{\cos \phi} \right\} \quad (3.95)$$

The first term provides the potential barrier necessary to prevent the cart from hitting the ends of the track. Note that  $\tanh^{-1}$  (which is in  $M_0^+$ ) is very near linear over  $[-0.5, 0.5]$  and diverges to  $\pm\infty$  at  $\pm 1$ , respectively. As  $|x| \rightarrow x_m$ , the nonlinear cart control action (equation 3.95) asymptotically approaches infinity and theoretically prevents the cart from reaching the end of the track. In practice, if  $\ddot{x}$  saturates, then this potential barrier cannot always prevent the cart from hitting the end of the track. However, by reducing the size of the Balance region - by reducing  $\phi_{max}$  and hence  $\dot{\phi}_{max}$ , and by bounding the value of  $f_d(\dot{\phi})$  in the Spin region, it is possible to constrain the system so that the saturated  $\ddot{x}$  is sufficient to keep the cart from hitting the ends of the track.

### 3.5.3 Experiments with a Physical Cart-Pole System

The controller designed above was implemented on a physical system, shown in figure 3.7, and the result of its operation is depicted in figure 3.8. This system is a physical implementation of the model in equation 3.58, with  $k = 80$  and a small amount of friction



Figure 3.7: Picture of the physical cart-pole system used in the experiments described in section 3.5.3.

<sup>4</sup>,

$$\ddot{\phi} + 0.25\dot{\phi} - 79.898 \sin \phi + 8.1446\ddot{x} \cos \phi = 0 \quad (3.96)$$

$\ddot{x}$  was of the form of equation 3.95 (with  $u(\phi, \dot{\phi})$  being the linear control functions described in section 3.4.4, with  $d_1 = d_2 = 0.5$ ,  $c_{11} = 1$ ,  $c_{12} = 5$ ,  $c_2 = 0.1$ ,  $c_3 = 2$ ).

Figure 3.9 depicts the response of this control strategy to a very hard impact applied to the pole (instantaneously changing the pole velocity by over 20 rad/s). We see the system recover naturally.

### 3.6 Synthesizing Nonlinear Controllers in a Qualitatively Constrained Space

One benefit of defining weak sufficient conditions, as done above, is to be able to specify a constrained space within which numerous quantitative controllers may be synthesized, subject to different cost functions. This is the notion of defining the *least restrictive design*,

---

<sup>4</sup>Note that the value of  $k$  is derived from mechanical design considerations whereas the friction term is estimated by parameter fitting against experimental data.



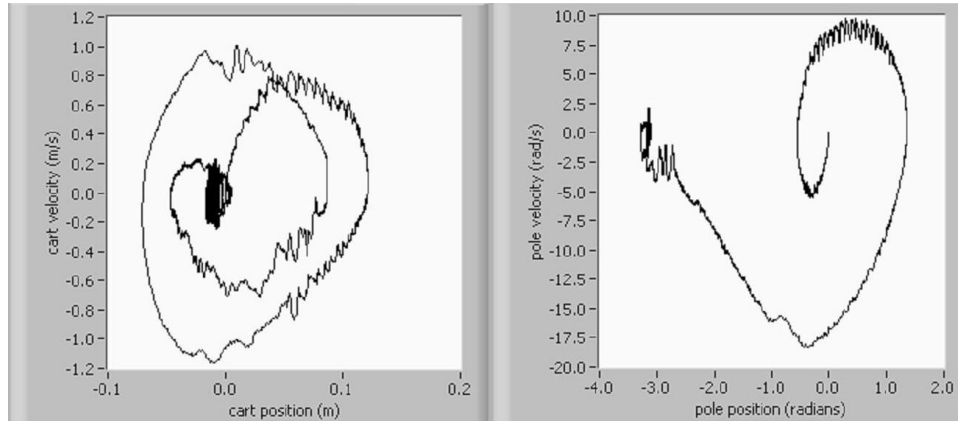


Figure 3.8: Operation of the global control strategy in physical experiment - trajectories in the  $x - \dot{x}$  and  $\theta - \dot{\theta}$  phase spaces. This corresponds to a single quick swingup - going through the three modes in a time optimal manner.

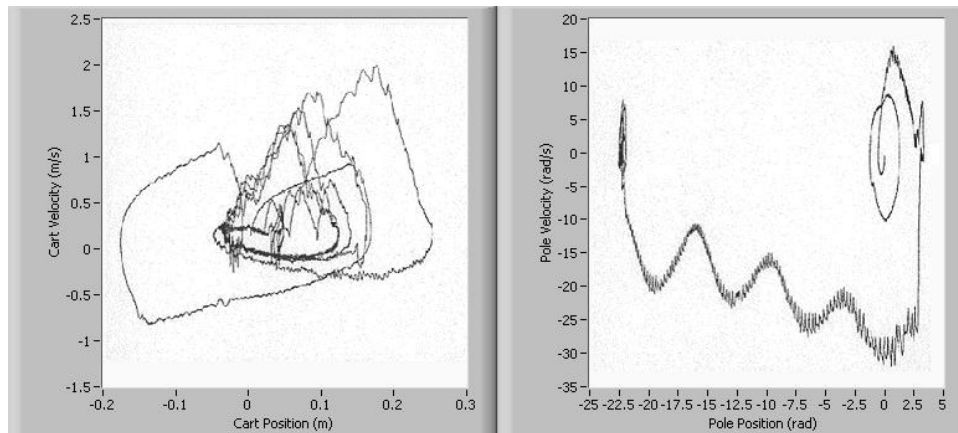


Figure 3.9: Recovery from a hard impact applied to the pole (which instantaneously changes the velocity by over 20 rad/s). The data is taken from physical experiment and trajectories are plotted in the  $x - \dot{x}$  and  $\theta - \dot{\theta}$  phase spaces. After the large impact, we see the system spend several full revolutions of the pole in the Spin mode. This is also seen as somewhat less symmetric cycles in the cart phase space.

which is in line with observations of natural tasks where *actions are always embedded in larger behavioral contexts* [4].

Here we present an approach to this design, which is based on parameterizing the qualitative controller functions used in the above arguments and then solving a parameter optimization problem using standard numerical optimization tools.

The procedure is as follows:

- Define a parametrization for the controller functions. Note that for nonlinear parameterizations, region definitions - particularly equations 3.42, 3.43 for the Balance, may need to be rederived.
- Define an optimization problem in terms of a cost, we adopt a standard linear quadratic cost  $\int_0^T x(t)'Qx(t) + u(t)'Ru(t)dt$ , with suitably chosen positive definite weight matrices  $Q, R$ .
- Evaluate the cost using a finite horizon ( $T = 20$  seconds) dynamic simulation <sup>5</sup>.
- Solve the optimization problem <sup>6</sup> subject to qualitative constraints defined in the previous sections.

This procedure was used to optimize multiple parameterizations, including:

- Linear local controllers:

$$\begin{aligned}
 - u(\phi, \dot{\phi})_b &= (c_{11} + k)\phi + c_{12}\dot{\phi} \\
 - u(\dot{\phi})_p &= (c + c_2)(\dot{\phi}) \\
 - u(\dot{\phi})_s &= (c_3)(\dot{\phi})
 \end{aligned}$$

- Nonlinear (sigmoidal) local controllers: With the sigmoid defined as,

$$\sigma(a, m, \phi) = a \left( \frac{1 - e^{-m\phi}}{1 + e^{-m\phi}} \right),$$

$$\begin{aligned}
 - u(\phi, \dot{\phi})_b &= k\phi + \sigma(a_{bp}, m_{bp}, \phi) + \sigma(a_{bv}, m_{bv}) \\
 - u(\phi, \dot{\phi})_p &= -c\dot{\phi} - \sigma(a_p, m_p, \dot{\phi}) \\
 - u(\phi, \dot{\phi})_s &= \sigma(a_s, m_s, \dot{\phi})
 \end{aligned}$$

<sup>5</sup>Implemented using the RK45 variable step size solver in the LabVIEW<sup>TM</sup> Simulation Module, ver 8.2.

<sup>6</sup>Achieved through the use of the Constrained Nonlinear Optimization function in LabVIEW<sup>TM</sup>, ver 8.2.

The results of the optimization process are depicted in figure 3.10. In particular, note the last case where the control action is only applied some of the time, according to the rule:  $u' = u_{QHC}$  iff  $r > 0.25$ , where  $r$  is a uniform random variable in  $(0, 1)$ . The actual  $u$  is saturated at  $u_{max}$ . This simulates variance in loop rates, a common problem in many robotic systems, especially ones with network connections. We see that the energy based control strategy is insensitive to such variations.

Experiments were also conducted with several other parameterizations, including a neural network, where instead of  $\sigma(a, m, \phi)$ , a 2-layer network was used. However, it turns out that for the cart-pole problem, there is little benefit offered by the extra flexibility and both the simple sigmoid and the more complex multilayer network achieve very similar optimal costs.

### 3.7 Discussion: What is Hard about Task Encoding?

This chapter has involved a detailed study of the solution for a specific design problem - the global control of pendulum systems. Given the fact that this is a classic benchmark problem, which has been studied for several decades now, there are numerous prior solutions in the literature, e.g., [77], [78], [76], [79]. There are many similarities between this prior work and the work presented in this chapter. For instance, the idea of defining a Balance region and then defining strategies to deliver the system to this region is common in all attempts to this solution. However, the way this has been achieved is not always straightforward, and the strategies presented in works such as [77] are fairly intricate and delicate. Other strategies such as [78] do not take into account any of the constraints discussed in the present work. [76] addresses only one of the modes of the hybrid strategy - swingup.

However, quite apart from these specific issues, the primary purpose of the above exposition has been to develop an approach to thinking about difficult control problems, taking into account the full range of issues - ranging from smooth nonlinearities to state and actuation constraints. Through the study of this benchmark design problem, we have established that it is possible to accommodate a variety of task specifications in a common framework - involving the definition of target dynamical systems, natural region boundaries for each of these target dynamical systems, derivation of weak sufficient conditions for the achievement of these targets and the derivation of specialized constraints that preserve the above guarantees in the presence of deleterious effects that plague practical robotic systems. This notion of incrementally accumulating constraints is useful because it enables the user

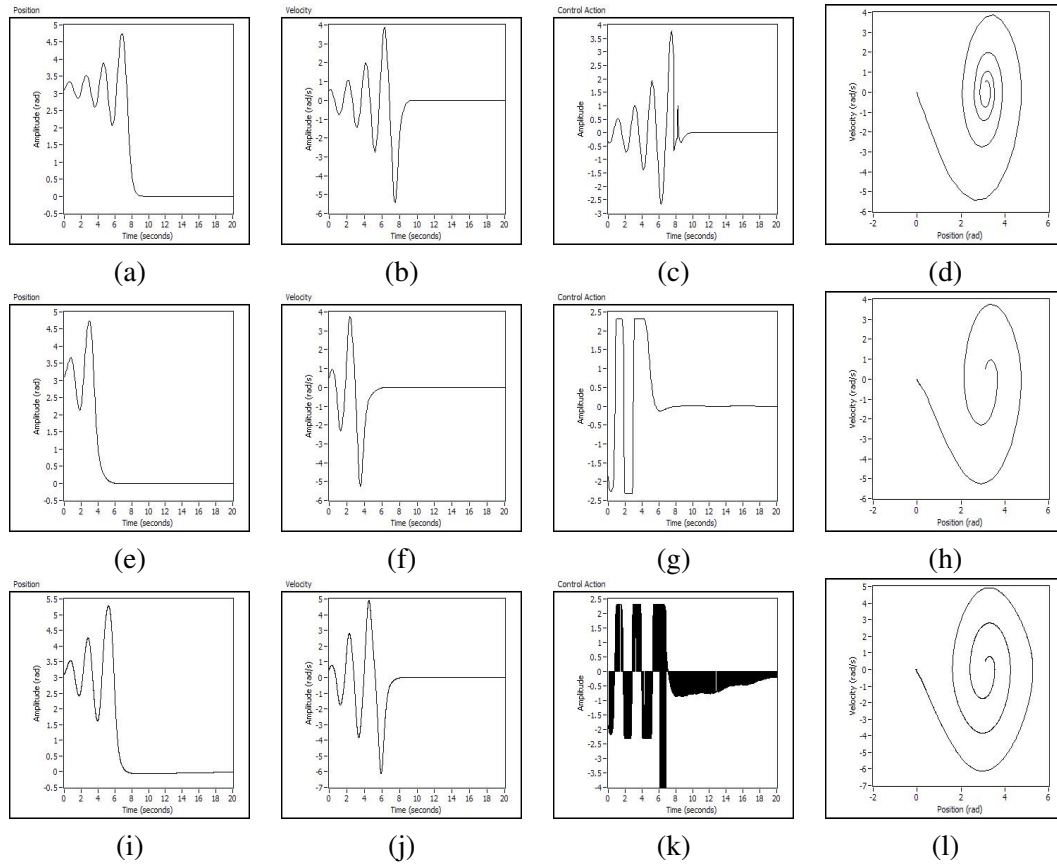


Figure 3.10: Results of dynamic simulation of the pivot-torque pendulum global control system, synthesized using numerical parameter optimization for selected architectures. Plots (a - d): Linear local controllers. The system stabilizes within the Balance region in just under 10 seconds. Plots (e - h): Nonlinear (sigmoidal) local controllers. The system has stabilized within 6 seconds. Plots (i - l): Nonlinear local controllers, with random variation in the time base for  $u(\theta, \dot{\theta})$ . i.e., random sampling time. This takes longer to settle, 8 seconds, but this is still faster than the linear case. This illustrates tolerance of the energy control strategy to imperfect implementation.

to stop the analysis at any point, when the available information is insufficient, and solve the rest of the problem using optimization or learning algorithms. This is an important benefit in the face of imprecise knowledge.

Most robotic tasks of practical interest are more complex than the cart-pole system, due to higher dimensionality, tighter coupling with an environment that could act adversarially and more imprecision regarding the dynamics. Nonetheless, many of the insights gained from the analysis in this chapter continue to be relevant. Importantly, this chapter lays the foundation for and argues in favor of a three step methodology involving problem decomposition into low/high-dimensional versions, reasoning about the task in a low-dimensional system and embedding the solution in the high-dimensional system through optimization or machine learning.

The simplicity of the pendulum and cart-pole problems in this chapter allowed us to solve two distinct problems at the same time, but the more difficult tasks treated in the rest of this dissertation will require us to distinguish between:

1. Motion planning: the definition of the global behavior in terms of a hybrid system that uses phase space geometrical structures (manifolds such as homoclinic orbits, energy level sets, etc.).
2. Local controller design for following the planned trajectories.

The latter problem, of local design, can be solved in a variety of different ways.

When a dynamical model is available, one could draw on several advances in model based design techniques. The approach to local design in this chapter is loosely related to the model reference control [59, 60] technique. Geometric control theory, e.g., see the book [89], further extends this in the form of energy shaping. Both of these would be excellent ways to (locally) stabilize the system around a dynamically realizable trajectory.

In future chapters, we will begin to focus our attention on the lack of detailed information about high-dimensional dynamics. In such scenarios, a preferred approach is predictive control (also known as receding horizon control), augmented by function approximation schemes such as neural networks<sup>7</sup>, see, e.g., [90, 91].

So, once a dynamically realizable trajectory has been computed, existing methods are sufficient to ensure that they are followed. Even in the absence of system models, techniques such as in [90, 91] can be used to arrive at stabilizing and tracking control laws.

---

<sup>7</sup>In fact, such a design can be carried out with the aid of software toolboxes, e.g., using the neural network predictive controller implemented in the *Matlab*<sup>®</sup> Neural Network Toolbox.

However, the global problem of generating trajectories to achieve a complex task, despite the absence of good high-dimensional dynamics models or perfect observations of a pre-existing solution, is relatively unexplored although it is crucial. This will form the focus of later chapters.

## Chapter 4

# Bipedal Walking: Motion Planning for a Template Model

### 4.1 Introduction to the Dynamic Bipedal Walking Problem

Legged robots represent great promise for transport in unstructured environments. Particularly, in environments such as swamps and rocky landscapes, and in applications requiring low impedance controlled behaviors - e.g., rescue or exploration, legged locomotion may be the only feasible modality. For this reason, they have captured the imagination of the researcher and the lay-person alike, and resulted in several decades of intense research activity. Over the years, several legged robots, often bipeds, have been constructed - to walk [73], hop [2], run [69, 92] and even perform gymnastic maneuvers [93]. Yet, we are far from having motion planning algorithms that allow us to replicate the efficiency and flexibility of human walking over natural *irregular* terrains, e.g., while stepping over a sequence of rocks on a pond.

While robotics researchers struggle to achieve these ambitious goals, we see that a majority of land animals travel on legs [4] and they do so in a very efficient, reliable and graceful manner, with minimal cognitive effort and attention. How can we replicate this success? In the spirit of functional biomimesis, can we design motion planning and task level control algorithms that can achieve this elusive combination of efficiency and flexibility? This is the primary focus of this chapter, and the next.

As such, there are only a few hard requirements that specify the task of *walking*. The legs of the bipedal walker must swing to intermittently make contact with the ground

at a sequence of progressively spaced footholds, and this must be achieved while the torso is being held upright, i.e., without falling down on the ground or adopting other modalities such as crawling. At this level of generality, the task is ill-posed. It is this ill-posedness that permits the incredible flexibility that is seen in animal locomotion <sup>1</sup>. So, the algorithmic challenge is twofold - (a) to develop *least restrictive* ways to regularize the motion planning problem, without completely losing the inherent flexibility afforded by the task and (b) to ensure that the resulting strategies are well behaved, e.g., stable on regular terrain.

To make the problem concrete, consider the terrain depicted in figure 4.1. In order for a bipedal walker to safely cross the river, it would be desirable to step precisely on the red dots which are good footholds. This is the only real constraint arising from the environment. Typically, the walker has a finite sequence of such footholds corresponding to a horizon within which sensing is reliable. The goal of motion planning is to generate trajectories in the phase space of the walker that can achieve this external goal, in the incremental or on-line fashion that these goals become known to the walker, and in an energy efficient and safe way. By *safe*, we mean that the dynamical properties of the algorithm should be amenable to analysis and should support some measure of reliability.

Moreover, as stated in section 1.1, this problem needs to be solved in the absence of analytical models of the high-dimensional nonlinear dynamics. This is coupled with the fact that even simple walking machines are capable of surprising complexity [94].

## 4.2 Some Important Ideas in Dynamic Bipedal Walking

The bipedal walking problem has been studied for several decades and the complete list of different techniques used to address this problem is too long to be cited here. References that serve as landmarks in this field include [95, 2].

Early work in bipedal walking treated the walker as being similar to any other manipulator, the only differences being in the way the payload, i.e., the torso, was being transferred between manipulators that maintained contact with the ground. However, this required the robot to act as a statically stable system so that traditional trajectory generation strategies could be adopted without having to account for the intermittent nature of leg contacts. The necessary condition for static stability is that the center of mass must lie within

---

<sup>1</sup>For a light hearted, but insightful, look at the possible variety of walking gaits, the reader is invited to watch the television show, Monty Python's *Ministry of Silly Walks*. Many of the gaits described there would be very hard to achieve with state of the art planning and control techniques.





Figure 4.1: A typical terrain requiring the ability to deal with intermittent footholds. The sequence of good footholds (red dots) represents the primary constraint placed on the walker by the environment.

the support polygon which is defined as the convex hull of the feet that are supported by the ground. For a bipedal robot, with only one small foot on the ground, the set of statically stable poses is very small and so gaits based on this requirement are inefficient and unnatural.

In an attempt to bring in the notion of dynamic stability, which allows for a larger set of configurations to be utilized by a trajectory generator, several measures and requirements were developed. The most popular ones are the Center of Pressure (CoP) and the Zero Moment Point (ZMP) [96]. The Center of Pressure is defined as the distance weighted average of the pressures on the feet, shown in figure 4.2 for an infant's foot. The Zero Moment Point is the point on the ground where the *tipping* moment acting on the biped, due to gravity and inertia forces, equals zero; the tipping moment being defined as the component of the moment that is tangential to the supporting surface. With these concepts, the stability condition can be specified concisely as - keep the CoP and ZMP within the support polygon. This is a necessary condition for dynamic stability but not sufficient. So, these principles are often combined with motion planning algorithms, e.g., to drive a constrained search process in the pose space [97]. Many modern robots utilize this approach, the most famous example being the Honda *Asimo* robot.

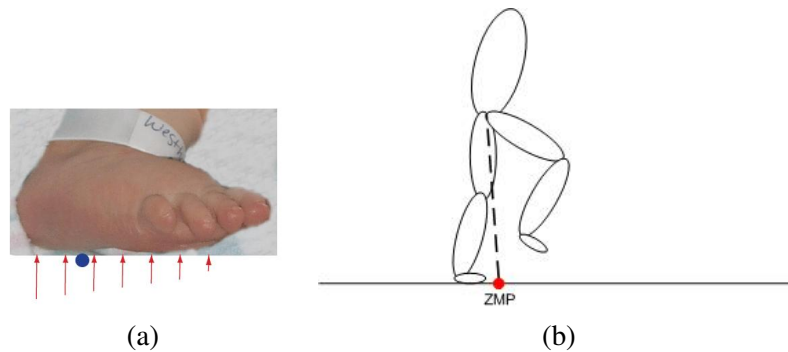


Figure 4.2: (a) Picture illustrating the Center of Pressure. The red arrows denote the magnitude of the pressure at a few discrete points and their distance weighted average is the blue dot, the CoP. (Photograph courtesy of Dr. Jerry Pratt, Florida Institute of Human and Machine Cognition). (b) Illustration of the Zero Moment Point - the point about which the net gravitational and inertial torques are zero.

There remains the need for a more comprehensive set of principles to encode the task of walking, that enable more dynamically dexterous behaviors than the above mentioned heuristic approaches. In recent years, this problem has been addressed by several teams of researchers, resulting in admirable successes [73, 48, 32, 72, 98]. One common theme on which each of these techniques represents a variation is the definition and use of low-dimensional task descriptions in terms of different physical and mathematical principles.

The earliest work in this series, by Pratt et al. [73], was based on the notion of *Virtual Model Control* wherein a set of “intuitive” heuristics were used to formulate a hybrid control strategy of the following form.

- Local Control Laws: Proportional-Derivative controllers (i.e., enforcing spring-damper dynamics) for each of the following,
  - Height: maintain constant height above the ground, which amounts to controlling a more directly measurable quantity - leg length.
  - Pitch: maintain a level pitch or follow some predefined trajectory.
  - Forward speed: change stride length and or transition time between modes
  - Swing leg placement: Either let the swing leg move passively, after an initial push, or control its trajectory as a function of time or state

- Mode Transition Rules: Transition between double support (i.e., both legs on the ground) and single support (only one leg supporting the torso) according to the following rules,
  - Go from double to single support if the body is within certain distance of next support leg
  - Go to single support if the body is more than a certain distance away from previous support leg
  - Go to double support if a certain threshold time, for single support phase, is exceeded
  - Lift the swing leg if the force on the leg falls below a threshold

The primary benefits of this approach is that each rule is a direct, and simple, implementation of a local behavior. The desired global behavior can then be synthesized by empirical tuning.

Tedrake [48] took this further by realizing that if one were to begin with a passive dynamic robot, that is inherently stable albeit for some narrow range of down-slope conditions, then the use of these local control principles enables a concise encoding of the task of actuated-passive walking on suitably regular and smooth slopes. Moreover, the process of empirical tuning may be replaced by machine learning (policy gradient reinforcement learning) such that the robot can improve its performance and adapt to changes in the environment.

The difficulty with the virtual model control approach, however, is that it is hard to establish properties of the global behaviors. How do the rules interact? What are the limits of applicability of each local mode? When does the overall strategy lead to walking? As discussed in chapter 3, these global issues are crucial for task achievement. One way to overcome these difficulties is through the use of nonlinear control design techniques.

In the work of Hofmann et al. [32], the starting point is a feedback linearization controller [60], based on a detailed analytical model of a simulated robot's dynamics, that reduces the control target to a decoupled linear system. From the point of view of task encoding, this model based control design process has the same effect as the use of the passive dynamic walker in the work of Tedrake [48], in that it eliminates the complexity of nonlinear dynamics and reduces the higher level planning/learning problem to one of tuning with a much more benign system. [32] includes a detailed discussion of the use

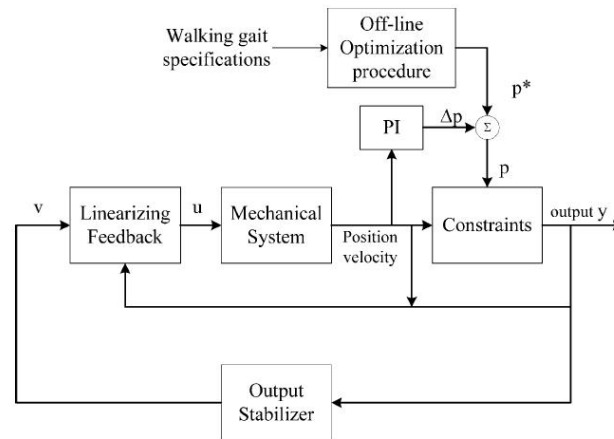


Figure 4.3: Block diagram of the controller used to perform walking using virtual constraints, adapted from [98].  $p$  represents degrees of freedom of the low-dimensional system after the virtual constraints have been actively enforced.

of temporally flexible planning algorithms to synthesize trajectories that can be executed within architecture to achieve walking on regular and irregular terrains.

Westervelt et al. [72, 71] formalize this intuitive notion of virtual model control in mathematically rigorous terms. They utilize the notion of *virtual constraints* [98] to specify an encoding of the desired outputs of a high-dimensional nonlinear system. Virtual constraints are relations among the links of the mechanism that are dynamically imposed through feedback control. Their function is to coordinate the evolution of the various links throughout a step which is another way of saying that they reduce the degrees of freedom with the goal of achieving a closed-loop mechanism that naturally gives rise to a desired periodic motion. As long as suitable virtual constraints, limit cycle periodic trajectories in [72], are being actively enforced, the control problem reduces to that of ensuring the stability of the zero dynamics [60]. The resulting procedure is diagrammed in figure 4.3. It is even possible to modify the virtual constraints over time, e.g., to switch between walking speeds. However, the resulting mathematical analysis still requires sufficient regularities. So, for instance, the switch may correspond to sufficiently slow transitions between stable limit cycles but not to an irregular foot placement sequence such as in figure 4.1.

The above mentioned approaches, all of which were proposed only within the last decade and are still under active development, provide very valuable insights and guidance for the way forward. However, they do not address the requirements stated in section 1.1.

The rigorous techniques based on nonlinear control theoretic approaches [32, 72, 98] are critically dependent on having precise and detailed analytical models. Moreover, adapting them to scenarios involving imprecise models is an entirely nontrivial undertaking [99]. On the other hand, machine learning based approaches [48], that are geared to handle model imprecision, are not yet capable of dealing with the instabilities and dynamical complexity of realistic robots. They depend on having a preexisting solution that reduces the learning problem to one of performance improvement.

Lastly, as stated in the introduction to Raibert's influential book [2], one of the major driving forces behind the study of legged robots is the possibility that it may yield insights about similar controlled processes in nature. However, many existing approaches based on engineering control theory do not strive for this connection to biological reality - instead, assumptions and simplifications are driven primarily by mathematical and analytical convenience. The work of Pratt et al. [73] is an important exception to this rule in that they strive to utilize and embody aspects of natural walking, e.g., passive dynamics, in their designs - with admirable success.

The goal of this chapter, and the next, is to present a novel approach to movement generation for bipedal robots. This approach will draw essential insights from the previous works mentioned above - the use of low-dimensional virtual models to encode the task, the use of passive dynamics to simultaneously address the goals of underactuation and low-attention, the use of learning to adapt to an unknown environment, etc. - but each of the constituent algorithmic components will be structured differently, allowing us to more directly address the goals stated in section 1.1.

### **4.3 Observations Regarding Biological and Biomimetic Robotic Walking**

Depending on one's perspective, bipedal locomotion seems surprisingly difficult or trivially easy. Locomotion results from complex dynamical interactions between the walker and its environment. The walker, whether natural or artificial, is a complex multi-link mechanism. The environment is also complex and constantly changing. Yet, legged animals do not seem to be bogged down by this complexity.

Nature penetrates the wall of complexity through the clever use of synergies and symmetries. In [6], Full and Koditschek present a strong argument that animals handle

complexity by devising control strategies based on dynamically simple *template* models and then embedding them into the higher dimensional physical system, the *anchor*. There is reason to believe that this is a generally applicable law of nature. Even animals with an essentially infinite-dimensional actuator, e.g., the octopus, synergize the motion down to low-dimensional stereotyped movements [100] that act as templates. Observations of infants [101] during their first walking months demonstrate that the process of learning to walk includes, as a critical step, the process of learning to reliably synergize their muscles to make each leg behave like a rigid pendulum.

With these observations in mind, the discussion of virtual models and constraints in the previous section should seem entirely natural. However, the templates observed in nature are often different from the choices made in earlier engineering designs - where decisions were driven primarily by considerations of analytical convenience.

The dominant hypothesis regarding templates for bipedal walking is that each leg acts as a pendulum, executing a *compass gait* [11], [5] - walking consists of a synchronized sequence of vaulting over an inverted pendulum and swinging a suspended pendulum, in a suitably synchronized fashion. Roboticists have constructed systems that are direct physical embodiments of this model, e.g., [102, 9]. In [8], we find mechanical embodiments of actuated versions of simple compass gait robots and an empirical demonstration that they are capable of walking on flat ground. While these machines are elegantly simple, they are also fragile and unsuitable for rugged environments. In [9, 94], we find mathematical analysis of the dynamics of this class of *simple walkers*. The walker is analyzed as a nonlinear system, to locate periodic orbits whose stability is understood in terms of convergence in a Poincare map [86]. This analysis provides insight but it does not make it possible for the walker to navigate rugged terrain. Nonetheless, the work on passive walking highlights an organizing principle in nature - the dynamics of the interaction with the environment can be gainfully utilized to implement energy storage and exchange mechanisms, bringing down total energy consumption [4]. Even in situations where energy storage mechanisms, e.g., springs, are not physically available in the robot, it is often more energy efficient to simulate the passive component than it is to pursue alternate active strategies.

Concretely, the compass gait model for walking is depicted in figure 4.4. This model focusses attention on the torso, which is the heavy payload and whose movement is the primary reason for application of forces. The torso is vaulted over the support leg which is alternately the right and left leg. Forces need to be applied in each double support phase, to redirect the torso as shown. Also, the swing leg needs to move forward to become the

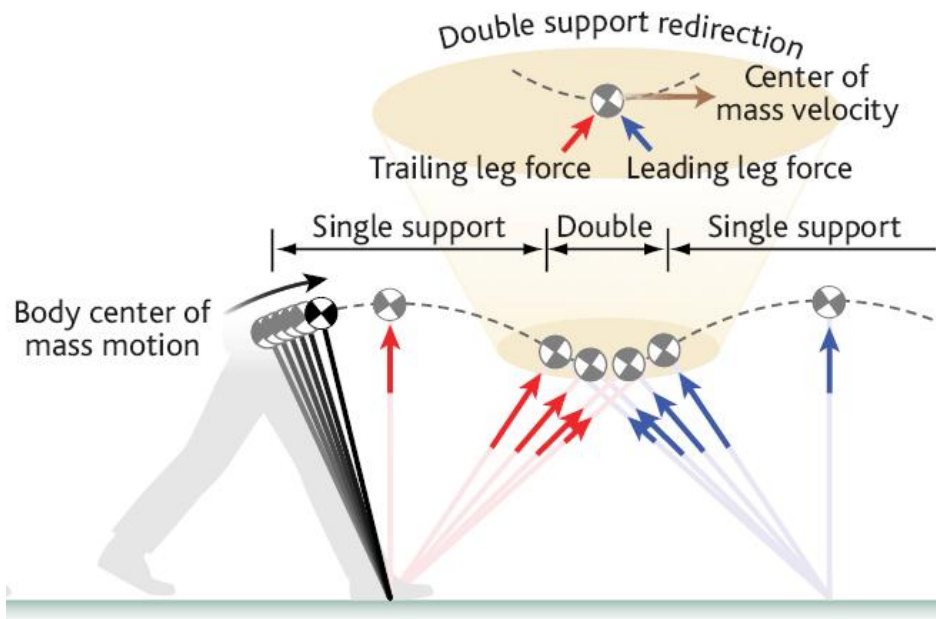


Figure 4.4: Conceptual schematic of the compass gait model of human walking (Reproduced, with permission, from [103]).

next stance leg - which can be achieved passively or ballistically in this model - also the case in human walking [10].

An additional piece of evidence that supports this choice of abstraction is based on a recent analysis by Srinivasan and Ruina [104]. They used a minimal model that describes some key aspects of walking, running and a variety of other gaits - a point mass on telescoping massless legs. This was used with numerical optimization to find energetically optimal gaits (see figure 4.5). At low speeds, this process “discovers” the compass gait and at high speeds it discovers running. So, it is not just an accident that humans walk this way but in fact there is a basis for this abstraction in the energetics of the task.

The point of the above discussion is to motivate the design decisions made in the current work. Based on the biological and biomechanical evidence supporting the compass gait, we choose to use it as the template model for task encoding. Loosely speaking, it plays the role of a *virtual constraint* as discussed in section 4.2 or the structural aspect of the qualitative models in chapter 3. However, all of these biological studies are descriptive in nature, i.e., their goal is to build models that describe what happens in nature. Our goal, as roboticists, is to go in the other direction - to design strategies that will compel the robot

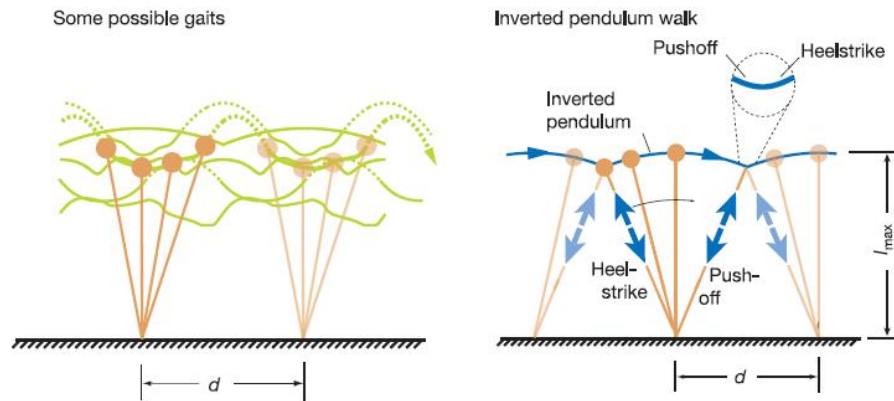


Figure 4.5: Some sample paths from the space of possible trajectories that may be achieved by the simple walker abstraction and the energetically optimal compass gait (Reproduced, with permission, from [104]).

to behave in suitable ways.

Moreover, walking in the real world requires tradeoffs [4]. In animals, we see tradeoffs between energy efficiency and the need for rapid reliable maneuvers, e.g., while escaping from a predator. In robotics, we seek machines that can perform useful tasks as they walk, e.g., carry a rescue victim while avoiding stepping on debris along the path. This suggests that gaits ought to be designed to be capable of rapid adaptation to a changing environment. It has been challenging to implement such adaptation while maintaining safety. Robots that are designed according to the passive/simple walking paradigm and are amenable to principled analysis are not well suited to unstructured environments. Correspondingly, several empirically successful adaptive machines have no clear basis for stability or correctness arguments.

With all of the above considerations in mind, some concrete technical questions arise:

1. Can we actuate the compass gait model such that it can be made to walk on irregular terrain, such as in figure 4.1? In other words - given an underactuated hybrid dynamical system that behaves as an *unactuated* inverted double pendulum in single support and is briefly actuated in double support, can we design a motion plan that enables the system to safely and reliably achieve a precise sequence of irregularly spaced footfalls?



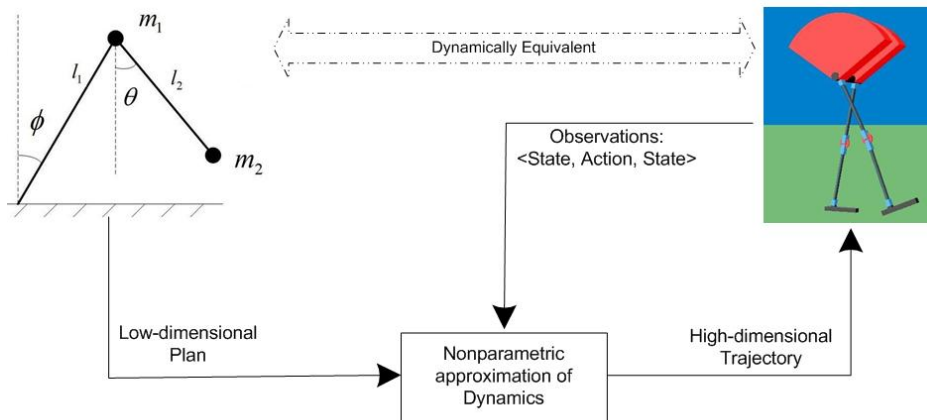


Figure 4.6: Block diagram schematic representing the approach followed in chapters 4 and 5. A low-dimensional template model is used to synthesize a plan that addresses the variations in terrain. The choice of the template model ensures dynamical equivalence to the higher-dimensional robot. This implies that once the dynamical properties of the template plan are verified, its embedding in the phase space of the higher-dimensional robot can be treated independent of task and terrain variations - achieved in chapter 5 via a novel manifold approximation algorithm.

2. Can we do something similar with a higher dimensional robot - with multi-link legs?

Question 1 will be answered in this chapter. This will then serve as the basis for answering question 2 in chapter 5.

## 4.4 Outline of a Strategy for Dynamic Bipedal Walking

The high level structure of our approach to dynamic bipedal walking is diagrammed in figure 4.6.

The overall goal is to generate a trajectory in the phase space of a simulated multi-link legged robot to traverse a sequence of irregularly spaced footholds. This robot is modeled after the physical robot described in [73] and is implemented in a physically realistic simulator developed by the same team - the *Yobotics Simulation Construction Set*. The structure of this robot is depicted in figure 4.7.

In this chapter, we will design a motion planning algorithm for a template model based on the compass gait walker. The quantitative parameters of this model will be chosen such that it will be dynamically equivalent to the multi-link legged robot, i.e., the anchor.

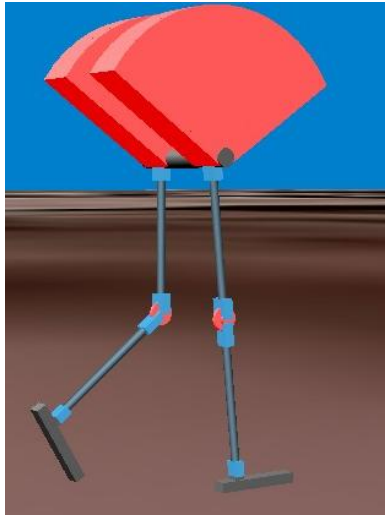


Figure 4.7: The simulated robot used in our experiments.

Once a low-dimensional trajectory is generated for the template model, it can be embedded in the phase space of the multi-link legged robot.

This yields a separation between different concerns in the following sense:

- Imprecision in the high-dimensional nonlinear dynamics is handled by data-driven approximation of the relationship between the template model and the anchor model. As such, this relationship is not affected by the task and is only dependent on drift in the system dynamics.
- Terrain variations are addressed by the motion planning algorithm for the template, and to the extent that the constraints of the anchor model can be translated to corresponding constraints in the template model (e.g., maximum pushoff force, maximum ankle torque or joint rotation limits), the low-dimensional plan is dynamically realizable.
- By its very nature, the template model is an abstraction of the anchor dynamics and there may be some constraints or un-modeled dynamics that are not captured by the template plan, e.g., singularities due to additional degrees of freedom. However, we generate the high-dimensional trajectory using an approximation algorithm based on observed high-dimensional dynamics - using the template trajectory primarily to regularize this approximation. So, to the extent that the template and anchor differ,

the resulting trajectory will be altered to follow the true dynamics of the anchor - whose subtleties are, in fact, captured by the data.

Note that all planning and approximation will be done in the phase space - representing the full dynamics, although results will often be depicted only in terms of configurations over time.

Posing the problem of dynamic bipedal walking in this manner yields some unique advantages over the prior work surveyed in section 4.2, as described below.

A walking robot traces out a trajectory in its phase space. Such trajectories can be described by a low-dimensional mapping,  $\mathcal{M}_L : \mathcal{S}_L \mapsto \mathcal{S}_L$ , for the template model with the state space  $\mathcal{S}_L \subset \mathbb{R}^M$  and a high-dimensional mapping,  $\mathcal{M}_H : \mathcal{S}_H \mapsto \mathcal{S}_H$ , for the anchor model with the state space  $\mathcal{S}_H \subset \mathbb{R}^N$ . For many locomotion and manipulation tasks, these curves are in fact restricted to evolve on specific submanifolds<sup>2</sup> in the phase space [53]. To aid concrete visualization, note that the separatrix trajectory of the pendulum, described in chapter 3, is a 1-dimensional submanifold corresponding to a particular level of energy. In fact, each periodic orbit of the undamped pendulum is a 1-dimensional manifold.

For the task of bipedal walking, the global structure of the manifold can be quite complex. Firstly, there is a stratified structure to the configuration manifold (only some combinations of joint angles are possible for a given robot geometry and the evolution is structured according to the current support leg). Even just at this level, involving no dynamics, the robot's movement traces trajectories along a stratified manifold, see figure 4.8 for a conceptual illustration. In fact, there are multiple trajectories that can execute any kinematic trajectory on this configuration manifold. When the dynamics are considered as well, i.e., in the phase space (generalized positions vs. corresponding derivatives), there is further structure imposed on the trajectories. The end result is that the task is achieved using a family of manifolds, each with special structure such as stratification.

When the bipedal walker traverses an irregularly spaced sequence of footholds, it must make several choices on each step. It must alter its gait continuously - in effect, jumping between different families of trajectories in phase space (which is analogous to selection of the appropriate periodic trajectory in a pendulum - a 1-dim manifold). The selection of trajectories lying on this manifold must be synchronized with the requirements of the external world - in space and time.

---

<sup>2</sup>The reader is referred to [14] for an excellent exposition of differential geometric concepts such as manifolds, and their relevance to dynamical systems.

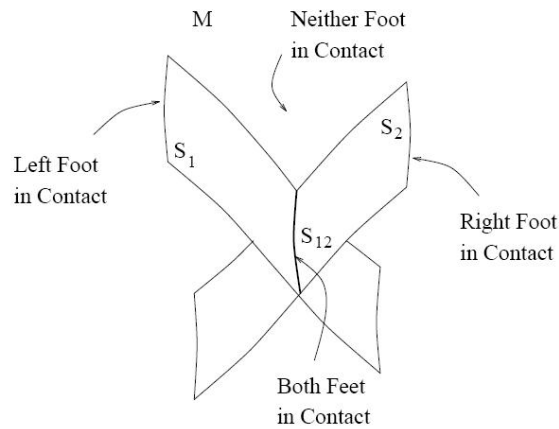


Figure 4.8: Conceptual illustration of the stratification of the configuration manifold of a bipedal walker, adapted from [105]. The axes in this figure are the configuration variables, i.e., joint angles. The manifold structure encodes the fact that - when walking, only some configurations are feasible. If we were to also include the dynamics, which manifests itself as further restrictions on the shape of curves on this configuration manifold, then the resulting phase space manifold will be even more complex.

The problem of kinematic planning in stratified configuration spaces (i.e., without considering dynamics or explicit foot placement goals) has been addressed before in works such as [105], given suitable geometric models of the robot and the environment. When dynamics are included, the problem has typically been studied as an area within nonlinear control theory (e.g., as described in section 4.2). Typically, the process involves two stages. Firstly, the *plan* is restricted to regular subsets and a preferred trajectory on the manifold - as opposed to switching between parameterized manifolds and synchronizing to the irregular terrain. This allows the use of linearization and related techniques, in conjunction with analytical models, to enforce the preferred periodic gait. However, in a sense, this is sidestepping the primary motion planning issues that are crucial to the task of irregular terrain walking.

Correspondingly, the fact that successful achievement requires such topological structure to be enforced is one of the major reasons why machine learning has been difficult in this domain. Machine learning algorithms that address this problem in a generic function approximation setting are confounded by these topological issues. Local function approximation techniques that have the expressive power to avoid this issue can require an

impracticably high data density along the manifold. So, in general, it has been hard to use data-driven approaches to induce behaviors over a reasonably large operating range. Even more sophisticated algorithms, such as the recent advances in manifold learning, run into difficulties due to the inherently ill-posed nature of this problem - see [34] for a discussion on the difficulty of using manifold learning algorithms to infer the existence of parameterized, high-curvature manifolds, etc.

In contrast, we separate the concerns of complexity and uncertainty. The template model captures the essential structure of the task and enables a much simpler encoding of the task. The problem of uncertainty is handled by data-driven manifold approximation, which is made tractable by using the template model to regularize an inherently ill-posed approximation problem. So, the essence of our approach to computing high-dimensional trajectories is to establish an equivalence between the maps  $\mathcal{M}_H$  and  $\mathcal{M}_L$ . Due to the fact that  $\mathcal{M}_H$  is capable of more complex higher-dimensional dynamics, we take this equivalence to be in the nature of an approximation.

Specifically, a good template model mirrors the anchor in two ways:

- It mimics the stratified topology of the anchor phase space.
- On each smooth open subset of the manifold, its trajectories can be described as a projection (computable from variables that can be readily observed) of the corresponding anchor trajectories.

So, we will require that the following diagram must commute:

$$\begin{array}{ccc} \mathcal{S}_H & \xrightarrow{\mathcal{M}_H} & \mathcal{S}_H \\ \downarrow \pi & & \downarrow \pi \\ \mathcal{S}_L & \xrightarrow{\mathcal{M}_L} & \mathcal{S}_L \end{array}$$

The use of this relationship to structure the manifold approximation problem will be addressed in chapter 5. The rest of this chapter will be dedicated to the design of the mapping  $\mathcal{M}_L$  to encode the task of irregular terrain walking.

## 4.5 Motion Planning with the Template Model

### 4.5.1 Template Model for Dynamic Bipedal Walking

The template model we use is a variant of the one shown in figure 4.4. Figure 4.9 depicts the pendulum systems involved in the compass gait abstraction. It consists of three point masses, one corresponding to the torso and two corresponding to the legs and feet. In the *swing* phase, one foot is on the ground and the system has only two moving masses, the torso and the swinging foot. Viewed from the vantage point of the stationary foot, the system looks like a double pendulum (when not actuated) and like the acrobot (when actuated). However, there is one difference in that each leg/link is modeled as being capable of achieving any desired length quickly, to overcome the terrain. A retraction in this model would correspond to a movement of the foot and bending of the knee in a multi-link robot. In numerical experiments, we will use a noticeable retraction (leg length at 80% of full extension, i.e.,  $\rho = 0.8$ ). However, this variable may be tuned to the terrain conditions.

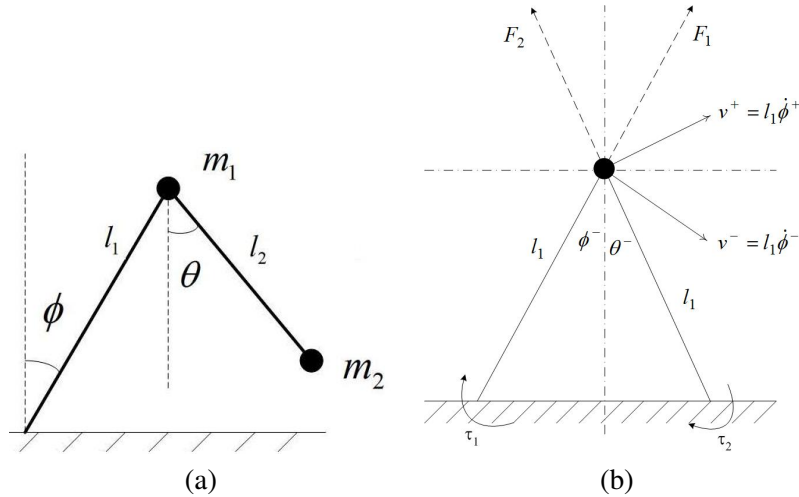


Figure 4.9: Schematic of the dynamic model of the template walking machine along with the relevant variables. Parts *a* and *b* correspond to the single support and double support phases, respectively. Note that the single support phase is *un*-actuated, i.e., ballistic.

For the single support phase, the equations of motion for this system can be derived using the Euler-Lagrange equations that encode the physical principle of least action [38].

The Lagrangian is the difference between kinetic and potential energies of the system,

$$L = \frac{1}{2}(m_1 + m_2)l_1^2\dot{\phi}^2 + \frac{1}{2}m_2l_2^2\dot{\theta}^2 + m_2l_1l_2\dot{\phi}\dot{\theta}\cos(\phi + \theta) - (m_1 + m_2)gl_1\cos\phi + m_2gl_2\cos\theta \quad (4.1)$$

where  $\theta$  is the angle made by the swing leg with the vertical axis and  $\phi$  is the angle made by the stance leg with the vertical axis.

From  $L$ , we can derive the equations of motion as two coupled nonlinear ordinary differential equations. The equations of motion are,

$$\ddot{\phi} - k_1 \sin\phi = \Delta_1 \quad (4.2)$$

$$\ddot{\theta} + \frac{k_1}{\rho} \sin\theta = \Delta_2 \quad (4.3)$$

$$\Delta_1 = -\mu\rho\cos(\phi + \theta)\ddot{\theta} + \mu\rho\sin(\phi + \theta)\dot{\theta}^2 + \alpha_1\tau_\phi \quad (4.4)$$

$$\Delta_2 = -\frac{1}{\rho}\cos(\phi + \theta)\ddot{\phi} + \frac{1}{\rho}\sin(\phi + \theta)\dot{\phi}^2 + \frac{\alpha_2}{\rho^2}\tau_\theta \quad (4.5)$$

where,

$$\mu = \frac{m_2}{m_1 + m_2} \quad (4.6)$$

$$\rho = \frac{l_2}{l_1} \quad (4.7)$$

$$k_1 = \frac{g}{l_1} \quad (4.8)$$

$$\alpha_1 = \frac{1}{(m_1 + m_2)l_1^2} \quad (4.9)$$

$$\alpha_2 = \frac{1}{m_2l_1^2} \quad (4.10)$$

and  $\tau_\theta$  and  $\tau_\phi$  are hip and ankle torques. The ankle is not actuated, i.e.,  $\tau_\phi = 0$ , for the purposes of handling terrain uncertainty, although it remains available for lower level controllers to compensate noise, etc.

These equations describe two pendulum subsystems, one inverted and one suspended, each perturbed by the other. In most realistic walking systems,  $\mu\rho \sim 0$ , so that  $\Delta_1 \sim 0$  when  $\tau_\phi = 0$ . However, the effect of the stance leg on the swing leg,  $\Delta_2$ , need not be negligible. In fact, this coupling term continues to influence the motion once the system

is in single support.

In the double support phase, both legs are on the ground and the system dynamics is much more constrained. For our purposes, the most relevant dynamics is that of the torso, which has four forces and torques acting on it to redirect its motion. The system comes into double support with a state,  $(\phi^-, \theta^-, \dot{\phi}^-, \dot{\theta}^-)$ . Following [65] and [66], the double support phase is treated as being short enough that the configuration of legs is roughly unchanged (hence, after swapping roles,  $\phi^+ = -\theta^-$ ,  $\theta^+ = -\phi^-$ ) but long enough that feasibly bounded forces can be applied to influence the velocity variables.

The primary mode of actuation in our strategy is to apply  $F_1, F_2, \tau_1, \tau_2$  during double support, which takes a nonzero time,  $\delta t_{DSP}$ <sup>3</sup>. These quantities may be computed to achieve a desired  $\dot{\phi}^+$ . Equating the applied forces to the rate of change of momentum during double support, and rearranging terms, we get,

$$\begin{aligned} \begin{bmatrix} \cos \phi^- & \cos \theta^- & \frac{-\sin \phi^-}{l_1} & \frac{-\sin \theta^-}{l_1} \\ \sin \phi^- & -\sin \theta^- & \frac{-\cos \phi^-}{l_1} & \frac{-\cos \theta^-}{l_1} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ \tau_1 \\ \tau_2 \end{bmatrix} \\ = \frac{m_1 l_1}{\delta t_{DSP}} \begin{bmatrix} \sin \theta^- & \sin \phi^- \\ \cos \theta^- & -\cos \phi^- \end{bmatrix} \begin{bmatrix} \dot{\phi}^+ \\ \dot{\phi}^- \end{bmatrix} + \begin{bmatrix} m_1 g \\ 0 \end{bmatrix} \end{aligned} \quad (4.11)$$

A similar analysis yields the torque  $\tau_\theta$  to achieve desired  $\dot{\theta}^+$ . With the exception of these short pulses during double support, the system dynamics will be permitted to evolve naturally without actuation. This is consistent with empirical observations of human walking, where muscles are *silent* during most of the swing phase and all activity is restricted to short intervals at the beginning and end of the swing. Similarly, for the use of toe-off forces, biologically implemented via plantarflexion.

The equations of motion define a nonlinear map, in terms of the state space,  $\mathcal{S}_1 \equiv \{\phi, \theta, \dot{\phi}, \dot{\theta}\}$ , a subspace,  $\mathcal{S}_2 = \{\dot{\phi}, \dot{\theta}\}$  and the action space,  $\mathcal{A} = \{F_1, F_2, \tau_1, \tau_2, \tau_\theta\}$ . The dynamic evolution of the system yields mappings between these spaces. In particular, the single support phase represents the uncontrolled map,  $\mathcal{M}_{SSP} : \mathcal{S}_1 \mapsto \mathcal{S}_1$  and the double support phase represents a map,  $\mathcal{M}_{DSP} : \mathcal{S}_2 \times \mathcal{A} \mapsto \mathcal{S}_2$  which does not influence the

<sup>3</sup>This is a design decision. We make this choice of actuation schemes to mirror the capabilities of the robot that will eventually be the anchor for this template model. However, one is free to dictate other choices of double support actuation schemes and the rest of the argument would still go through.



kinematic variables,  $\theta$  and  $\phi$ . At the algorithmic level and for planning purposes, it is assumed that these maps admit functional evaluation, either through a dynamic simulation or by experiments on a physical robot.

Note that the system is underactuated (in fact, unactuated) in the single support phase. So, the problem of motion planning is one of designing an appropriate strategy such that, given a sequence of irregularly spaced footholds, a corresponding sequence of double support forces can be generated, or the problem can be declared infeasible, *a priori*. The latter statement is essential to safety.

#### 4.5.2 A ballistic parametrization for irregular terrain gaits

Motion capture traces of human walking reveals that gait is almost never as regular as a simple limit cycle [106]. This is even more true when we consider walking on natural terrains which might involve irregular sequences of footholds. However, people do walk for long periods of time, without becoming unstable or falling down, and without really thinking about this process of walking. In fact, the key feature of walking is that one could continuously switch between different forms of gaits, in principle - drawn from an infinite family of such gaits - and still be “stably walking”. What this means is that the essential requirement is not that of convergence to a preferred periodic orbit but in fact a form of *weak invariance* [81] that ensures containment within a set of suitable behaviors. This is what imparts the flexibility required over irregular terrain. We will now describe a scheme that will allow us to construct a *family* of trajectories with this property.

The behavior of the stance leg of the compass gait biped is represented by the family of rotation orbits, and the swing leg behavior is represented by libration orbits. A composite orbit obtained by switching between representative trajectories from these qualitatively defined families of orbits is capable of describing the walking behavior. Of course, not all compositions will be admissible and further conditions will be necessary to achieve foot placement goals and satisfy constraints.

This process of composition is depicted in figure 4.10. The swing leg and stance leg trajectories will have to synchronize for the robot to successfully walk. Further constraints, e.g., desired step length, will enter this picture indirectly, as constraints on the sets from which trajectory segments may be chosen. This composition results in mappings of the form,  $\mathcal{M}_{DSP} \circ \mathcal{M}_{SSP} : \mathcal{S}_1 \times \mathcal{A} \mapsto \mathcal{S}_1$ . Steady walking consists of a suitable composite sequence,  $\mathcal{M}_{DSP} \circ \dots \circ \mathcal{M}_{SSP} \circ \mathcal{M}_{DSP} : \mathcal{S}_1 \times \mathcal{A} \mapsto \mathcal{S}_1$ . The goal of planning is to select

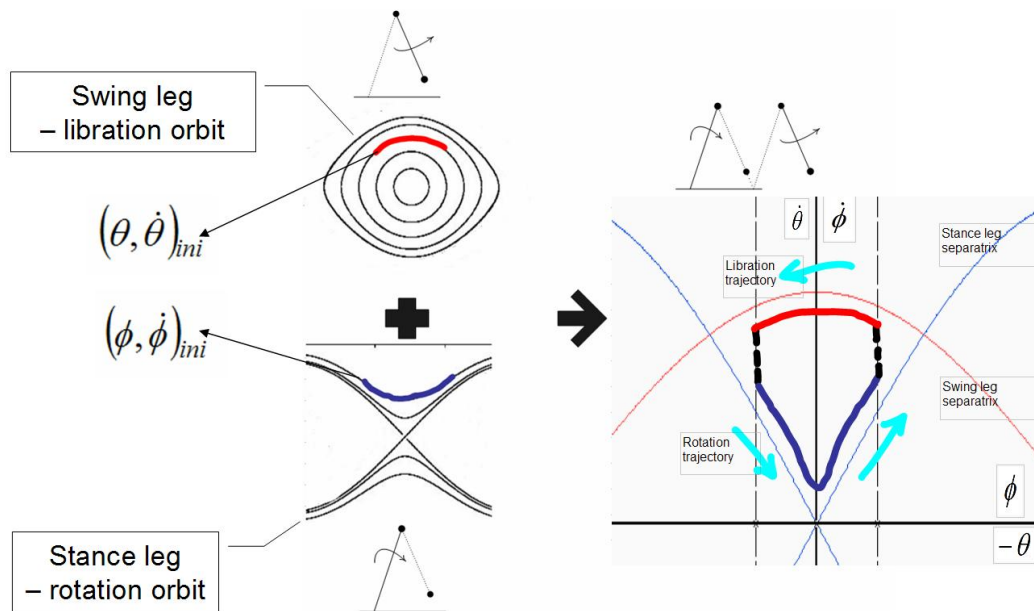


Figure 4.10: Synthesizing composite trajectories based on selections from families of qualitatively similar orbits. This figure depicts only a few representative rotation and libration trajectories. In fact, these are dense families and there are infinitely many such curves to choose from. Note that, for clarity of exposition, the trajectory segments have been aligned and transition is a simple vertical line. In general, these trajectories could be asymmetrical and aligned differently along the velocity axis, depending on kinematic conditions and terrain.

a policy of forces and torques that will impart certain properties to this map.

The basic property we wish to ensure in these maps is that the swing leg and stance leg trajectory segments are synchronized, i.e., they take the same amount of time to reach the planned footfall position. For a pendulum, the swing time may be determined analytically [107]. In terms of a parameter describing energy,  $\kappa = \sqrt{\frac{1}{2}(1 + \frac{E}{mgl})}$  (where  $m$  and  $l$  stand for  $m_1/m_2$  and  $l_1/l_2$ , as appropriate), we have for small angles,

$$\tau(\kappa) = \begin{cases} 2\pi\sqrt{\frac{l}{g}} & \kappa < 1 \\ \frac{\pi}{\kappa}\sqrt{\frac{l}{g}} & \kappa > 1 \end{cases} \quad (4.12)$$

and for larger angles,

$$T(\kappa, \tau) = \begin{cases} \tau \frac{K(\kappa)}{K(0)} & \kappa < 1 \\ \tau \frac{K(\kappa^{-1})}{K(0)} & \kappa > 1 \end{cases} \quad (4.13)$$

where  $K(\kappa) = \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1-\kappa^2\sin^2\theta}}$ .

Making sure that the swing and stance leg have similar periods is then a matter of choosing an appropriate  $\kappa$  for each leg, which amounts to a choice of  $\dot{\theta}, \dot{\phi}$  at the beginning of each step.

With this, we may directly ensure synchronization by a condition such as the following,

$$\int_0^{\phi_{des}} \frac{d\phi}{\sqrt{\dot{\phi}^2 + 2\frac{g}{l_1}(1 - \cos\phi)}} = \int_0^{\theta_{des}} \frac{d\theta}{\sqrt{\dot{\theta}^2 - 2\frac{g}{l_2}\cos\theta}} \quad (4.14)$$

This assumes that the stance and swing legs are decoupled. Strictly speaking, the swing leg *is* driven at the pivot by the movement of the torso. This complicates the dynamics of the swing leg and raises interesting questions about what sorts of footholds are in fact achievable. In fact, dynamic analyses of passive dynamic walkers, e.g., [94], bring up this very issue in the form of emergence of chaotic gaits.

It is well known that the driven simple pendulum is chaotic [107]. The primary implication of this fact is that the 1-dim manifolds in figure 4.10 are going to be distorted such that closed form equations such as 4.12 and 4.13 no longer hold exactly, and this brings up the question of how trajectories can be selected according to the scheme proposed in figure 4.10.

The source of this chaos is a resonance between the natural periodic dynamics of one leg (swing) and the periodic components of the external driving force due to the other

(torso on top of stance leg). A key feature of the transition to chaos is that it is gradual. As the driving amplitude is smoothly varied from a small value, only a few trajectories near critical points and the separatrix are affected. This means that only a few critically slow gaits are invalidated. Chaos begins here and slowly affects the rest of phase space. For our purposes, this implies that families of topologically equivalent orbits do not suddenly disappear. Instead, these regions gradually shrink. Eventually, as the driving amplitude increases, no meaningful qualitative statements can be made - global unpredictability sets in. If we can identify operating regimes and phase space regions where regular trajectories persist then we can still implement the strategy discussed so far, safely ignoring the more complex dynamics in other regions of phase space.

So, at moderate walking speeds, these nonlinear dynamic effects are not significant. More importantly, for moderate levels of driving forces, the primary effect is that the periodic orbits are distorted and equations such as 4.12, 4.13 become approximations. To the extent that the primary role of the template model is to provide guidance on the topology, in order to regularize the approximation problem outlined in figure 4.6, these distortions are not significant.

However, in order to demonstrate that the template model can be used without such approximations, we will present a numerical search algorithm that can effect the desired synchronization. The properties of the resulting scheme can be analyzed and used to infer safety on specific terrains.

Prior to that, we describe the results of a numerical experiment intended to estimate the regimes where the dynamics are regular. Our method is to observe the phase space behaviors through the evolution of randomly distributed points in phase space. By allowing an ensemble of points to evolve for a short period of time, we are able to visualize the deformation of the phase portrait. Figure 4.11 depicts a representative result of these experiments. We have overlaid the original trajectory shapes from figure 4.10 over our random dots results to aid visualization.

Regular regimes correspond to bounded  $\dot{\phi}, \dot{\theta}$ , i.e., bounded subsets of  $\mathcal{S}_1$ . In particular, the following constraints will yield an operating region in state space where the chaotic effects will not be significant:

1. Swing and stance leg angles,  $\phi, \theta$ , are restricted to a magnitude of  $45^\circ$ , from basic mechanical considerations.
2. The stance leg trajectories are restricted to “separatrix-like” trajectories, i.e., to a

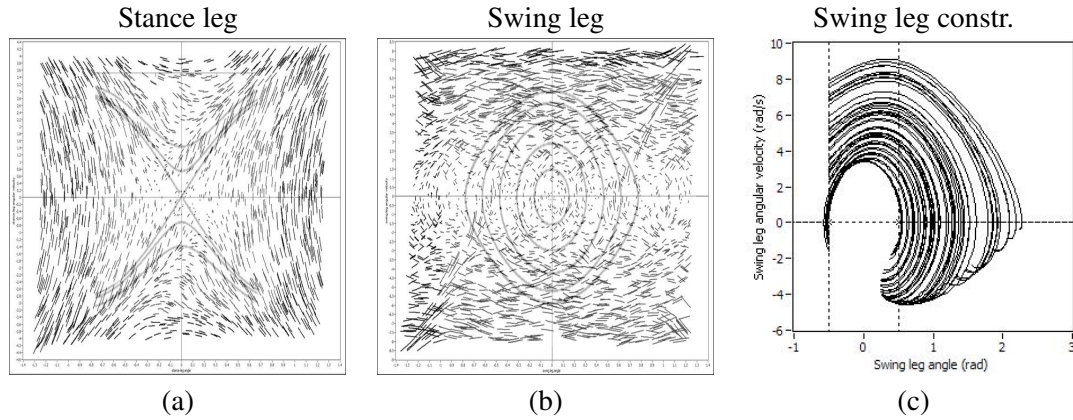


Figure 4.11: *Random dots* based phase portraits for the nonlinear system of equations 4.2 and 4.3. Phase space trajectories representing the unperturbed pendulum have been overlaid on the results to enable visualization of the deformation due to chaos. Note that these phase portraits correspond to the full 4-dim system. Each plot represents a projection onto a 2-dim phase plane, with angular position as the  $x$ -axis and angular velocity as the  $y$ -axis. As seen in part *b*, the swing leg phase space has been significantly perturbed. However, by placing the constraints outlined in the prose, we are able to extract a family of regular trajectories, as shown for the swing leg in part *c*.

1-dim manifold defined by  $\dot{\phi} = \beta \sqrt{\frac{2g}{l_1}(1 - \cos \phi)}$ , where  $\beta = 1$  describes the separatrix. This equation defines the chosen value of  $\dot{\phi}^+$  in the double support phase.

3. The magnitude of swing leg velocity,  $\dot{\theta}$ , is bounded above in our experiments  $[0, 8]$  rad/s.

### 4.5.3 An algorithm for intermittently actuated passive walking

We are now able to describe the motion planning algorithm 1. The concrete problem we solve is that of walking on a series of footholds, presented sequentially - one step at a time, in terms of a desired displacement  $(x_d, y_d)$  measured from the current stance leg foothold. There are several ways to select footholds, based on a variety of considerations that are commonly used in deliberative planning. However, footholds selected by any technique can be translated to a requirement in  $(x_d, y_d)$ .

The only step involving active control with energy injection/dissipation is double support, representing the application of constant forces and torques for a small period of time,  $\delta t_{DSP}$ . In our experiments, we fix this time period in order to compute forces and

torques. However, this value is a variable that can be changed for other experiments, perhaps based on empirical data.

---

**Algorithm 1** Intermittently actuated passive walking algorithm

---

**INPUT:** Initial state,  $\phi^-, \theta^-, \dot{\phi}^-, \dot{\theta}^-$ , and an assignment of swing and stance legs (e.g., stance = right, swing = left)

**OUTPUT:** Intermittently-actuated ballistic trajectories

**while**  $(x_d, y_d)$  is ahead of  $(x, y)$  **do**

- (*Double Support*) Apply  $F_1, F_2, \tau_1, \tau_2$  to implement desired value of  $\dot{\phi}^+$ . The stance leg velocity is given by  $\dot{\phi}^+ = \beta \sqrt{\frac{2g}{l_1}} (1 - \cos \phi^+)$ . The desired swing leg velocity,  $\dot{\theta}^+$ , will be computed by numerical optimization (described in the prose). This value can be achieved using  $\tau_\theta$ . This represents the execution of the map,  $\mathcal{M}_{DSP} : \mathcal{S}_2 \times \mathcal{A} \mapsto \mathcal{S}_2$ .
- (*Transition*) Retract the swing leg to length  $l_2$ .
- (*Single Support*) Allow the dynamics to evolve passively, according to the map  $\mathcal{M}_{SSP} : \mathcal{S}_1 \mapsto \mathcal{S}_1$ , until a *stopping condition* is reached (described in the prose).
- (*Transition*) When this stopping condition is reached, assign new values to  $\phi^-, \theta^-, \dot{\phi}^-, \dot{\theta}^-$  and extend the swing leg from length  $l_2$  to  $l_1$ .
- (*Transition*) Swap the roles of legs (e.g., for the first step, we will have stance = left and swing = right) and assign  $\phi^+ = -\theta^-, \theta^+ = -\phi^-$ .

**end while**

---

The desired amount of energy injection is computed by numerical optimization. In effect, this computes the inverse mapping  $\mathcal{M}_{SSP}^{-1}$ . The dynamic simulation for  $\mathcal{M}_{SSP}$  is executed until a stopping condition is reached, specified by geometrical considerations. The requirement is that  $\phi(t) = \phi_{des}$  and  $\theta(t) = \theta_{des}$  for some value of  $t$ . If  $A(x_1, y_1)$ ,  $B(x_2, y_2)$  and  $C(x_d, y_d)$  are the coordinates of the torso COM, swinging foot and desired foothold respectively, then  $\phi_{des}$  and  $\theta_{des}$  result from the consideration that AB and BC lie on the same line and AC has length  $l_1$ . We will use the former requirement and terminate the simulation when the following quantity crosses zero:  $\left[ x_d - x_1 + \frac{(y_d - y_1)(x_2 - x_1)}{(y_2 - y_1)} \right]$ . The constraint that  $l_1 = |AC|$  is handled in the optimization. This condition is illustrated in figure 4.12.

The computation of  $\gamma = \{\dot{\theta}^+, \beta\}$  needs to be a constrained minimization. The constraints involved are listed below:

1.  $\phi \in [-45^\circ, 45^\circ], \theta \in [-45^\circ, 45^\circ]$
2.  $\dot{\theta} \in [0, 8]$  rad/s

---

**Algorithm 2** Computation of  $\gamma = \{\dot{\theta}^+, \beta\}$  in the *Double Support* phase

---

**INPUT:** Procedures implementing (via a Runge-Kutta solver) the nonlinear map,  $\mathcal{M}_{SSP} : \mathcal{S}_1 \mapsto \mathcal{S}_1$ , based on equations 4.2 and 4.3. This defines  $\phi_\gamma(t)$  and  $\theta_\gamma(t)$ .

**INPUT:** Procedure for the solver termination event - geometric stopping condition (described in the prose)

**OUTPUT**  $\gamma = \{\dot{\theta}^+, \beta\}$

- Generate a set  $\mathcal{G}$  of  $N$  well-distributed (within specified finite intervals) values for  $\gamma = \{\dot{\theta}^+, \beta\}$

**for** each element in  $\mathcal{G}$  **do**

- Evaluate  $J = |w_1(x_d - [x_3(t_2)]_\gamma)^2 + w_2(y_d - [y_3(t_2)]_\gamma)^2 + w_3(\dot{\phi}_\gamma^+)^2|$
- With  $J_1 = |w_1(x_d - [x_3(t_2)]_\gamma)^2 + w_2(y_d - [y_3(t_2)]_\gamma)^2|$  and  $J_2 = |w_3(\dot{\phi}_\gamma^+)^2|$ , compute the best multiobjective optimum from the above set (subject to constraints mentioned above) using a vector distance from the origin,  $\sqrt{J_1^2 + J_2^2}$ .

**end for**

- Solve a sequential quadratic program for the  $\gamma = \{\dot{\theta}^+, \beta\}$  from the previous step to refine the solution.
- 

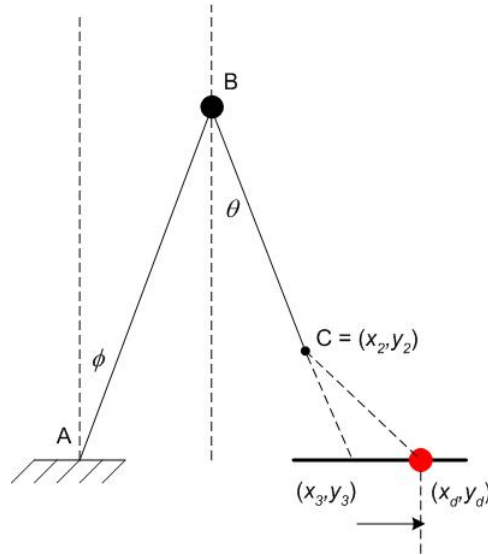


Figure 4.12: Illustration of the zero-crossing event that terminates the dynamic simulation used to optimize for  $\gamma = \{\dot{\theta}^+, \beta\}$ .

$$3. 0 < F_1 < F_{max}, 0 < F_2 < F_{max}, |\tau_1| < \tau_{max}, |\tau_2| < \tau_{max}$$

The last constraint, for finite actuation, is handled as follows:

With,  $A, B$  defined as (from equation 4.11),

$$A = \begin{bmatrix} \cos \phi^- & \cos \theta^- & \frac{-\sin \phi^-}{l_1} & \frac{-\sin \theta^-}{l_1} \\ \sin \phi^- & -\sin \theta^- & \frac{-\cos \phi^-}{l_1} & \frac{-\cos \theta^-}{l_1} \end{bmatrix} \quad (4.15)$$

$$B = \begin{bmatrix} \sin \theta^- & \sin \phi^- \\ \cos \theta^- & -\cos \phi^- \end{bmatrix} \quad (4.16)$$

Solve the quadratic program,

$$x = \arg \min x' I x$$

$$\text{such that, } Ax = B$$

where  $x = \{F_1, F_2, \tau_1, \tau_2\}'$ . This yields a minimum norm solution to the force assignment problem. Additional requirements could be incorporated into the above step, through the use of the more general quadratic cost,  $\frac{1}{2}x' Q x + c x$ .

Solving the above optimization problem amounts to achieving the kinematic specification  $(x_d, y_d)$  through the use of a low energy, rotation trajectory for the stance leg. The kinematic specifications specify a curve in the 2-dimensional space (see figure 4.13) of  $\gamma$ , not a unique point. An optimization problem that is based on just this requirement (with  $w_3 = 0$ ) would be underspecified and the solution would oscillate along this feasible curve. This is undesirable because it implies significant unnecessary force expenditure to switch between orbits. The term,  $w_3(\dot{\phi}^+)^2$ , is intended to encourage convergence towards a low energy gait, in keeping with our goal of energy efficiency. In a multi-objective and multi-task environment, the above cost functions may be augmented without impacting the structure of the algorithm.

Note that, unlike with purely passive walkers, e.g., [94], we are actuating the system in double support to achieve desired footfall goals,  $(x_d, y_d)$ . This changes the emphasis of the analysis - in particular, the effects of impacts and other minor errors accumulated along the step can be compensated directly, in a deadbeat fashion, during double support. Moreover, the constrained optimization framework allows us to ensure that the trajectories that are being chosen are actually realizable given the limited authority of the robot's actuators.



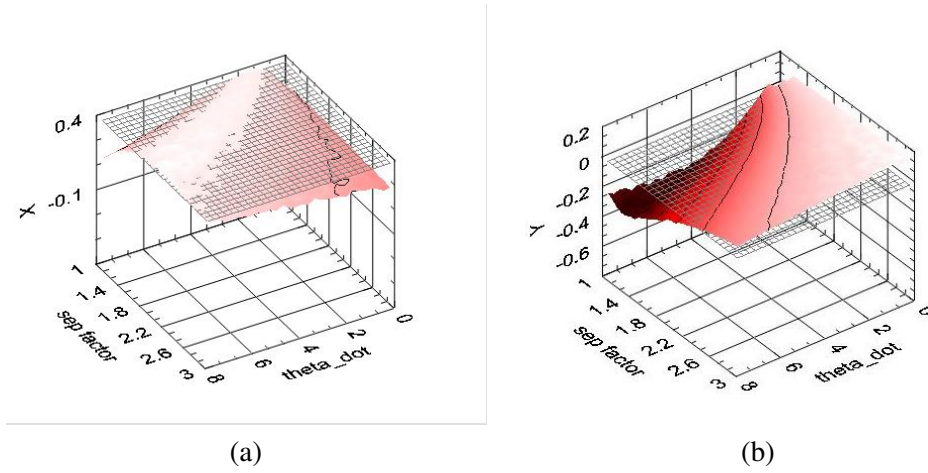


Figure 4.13: Illustration of the feasible surface from which optimization selects a single value of  $\gamma$ . This plot is generated by running the dynamic simulation (until the stopping event is reached, and from a fixed initial configuration) for various values of  $\gamma$  spanning a finite interval. The meshed planes represent  $x_d, y_d$ . We can see that there are multiple values of  $\gamma$  that will minimize the kinematic cost alone (along the intersection between the curved surface and the plane), and this is why the problem should be treated in a multiobjective fashion.

The procedure described in this section is a regularized global optimization algorithm. As such, this requires more computation than one might wish to invest in an on-line scenario. However, to the extent that it is completely specified analytically, one could use the above procedure off-line to acquire a more direct mapping between  $\alpha = (x_d, y_d)$  and  $\gamma = \{\dot{\theta}^+, \beta\}$ . A candidate representation for such a mapping would be radial basis functions of the form,

$$\gamma_j = \frac{\sum_i w_i(\alpha) \gamma_{j_i}}{\sum_i w_i(\alpha)} \quad (4.17)$$

$$w_i(\alpha) = \frac{1}{2\pi\Sigma^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\alpha - \alpha_i)' \Sigma^{-1} (\alpha - \alpha_i)\right\} \quad (4.18)$$

This approach of approximating a computation-intensive solution for the purposes of on-line operation is a common tactic in many modern predictive control works such as [90, 91]. In fact, the procedure described in this section is also a model-predictive algorithm.

The resulting map is a simple, explicit representation of the strategy, allowing the above algorithm to be used online in realistic robots. In biology, such a map may be ac-

quired developmentally, through continuous learning. The point of the design in this chapter is to constructively arrive at a similar result.

#### 4.5.4 Energy control within the template model

One of the benefits of parameterizing the families of feasible trajectories that are used in the search process of section 4.5.3 according to energy levels is that it leads to a fairly natural definition of the objectives of disturbance rejection algorithms. Specifically, to the extent that synchronization of the stance and swing legs is an important requirement, we can utilize the smooth relationship between energy and swing time to correct for disturbances.

Consider the following simple experiment. Define a rule for energy control of the swing leg as,  $\tau_\theta = -\epsilon(E - E^*)$ . The effect of this control strategy in the face of a typical disturbance - one that changes the value of  $\dot{\theta}$  instantaneously, is shown in the phase space in figure 4.14. We see that both the ideal and controlled trajectories quickly converge to the same region of phase space, without compromising global requirements of synchrony, etc. The small residual error seen in the figure is due to the simple nature of the proportional control law. A more sophisticated controller would yield significantly improved performance. The reader is referred to books such as [89] for an in depth discussion on these local control issues. For our current purposes, the point of this simple illustration is merely to note that the trajectories that are being generated by this algorithm can, in fact, be locally stabilized against disturbances.

#### 4.5.5 Simulation results

We evaluate this algorithm by simulating dynamic walking under progressively more irregular terrain conditions - going from flat ground, to climbing a set of regularly spaced stairs to walking on random terrain. In each of these simulations, the robot begins at rest. In every case, the maximum  $x$ -error is within 0.9 mm and the maximum  $y$ -error is within 0.3 mm. These are tight error bounds (for a robot with  $l_1 = 0.84\text{m}$ ) and suggest that this algorithm can be safely used in situations where only small discontinuous footholds are available.

The base case experiment<sup>4</sup> is to make the robot walk along equally spaced footholds on flat ground. For this case, figure 4.15 depicts the phase space behavior (corresponding

---

<sup>4</sup>All experiments in this section were carried out using a simulated robot, based on a dynamic model whose parameters (mass, leg length) are chosen to be equivalent to the high-dimensional robot model that will be discussed in chapter 5. The simulation was implemented using a Runge-Kutta solver in the LabVIEW<sup>TM</sup> Simulation Module 8.2.

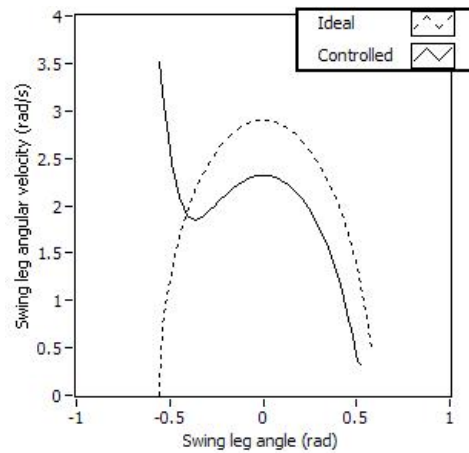


Figure 4.14: Effect of the simple energy control strategy on the phase space trajectory of the swing leg.

to the conceptual schematic of figure 4.10).

This corresponds to a steady gait at a forward velocity of approximately 1.1 m/s. Note that the resulting periodic orbit is not explicitly specified but results naturally from the nature of the algorithm. Further, we wish to verify whether the periodic trajectory chosen for this model has any bearing on trajectories that might be selected by more complex multi-link legged walkers. One way to answer this question is by comparing the speed versus step-length, for the stable periodic orbit that the algorithm has converged on, against humans. It is known [65] that humans adopt a preferred speed for each step length. This relationship is depicted in figure 4.16. The operating point of our algorithm is also depicted on this plot, and it shows that the chosen operating regime is only slightly away from the preferred speed curve for humans, at the 110% metabolic cost level. This implies that not only is the algorithm capable of recovering the stable limit cycle behavior of other walking control strategies, but also, it naturally results in realistic velocity behavior.

Moreover, figure 4.17 depicts the change in momentum between steps in this base case experiment. It is clear that within a couple of steps, the algorithm has converged to a stable behavior.

Next, we wish to demonstrate that not only does this strategy recover periodic gaits but is also has the flexibility to traverse irregular terrain. For this, we make the robot climb a flight of stairs, with results depicted in figures 4.18 and 4.19. From figure 4.16, we note the empirical observation that humans walk faster when step lengths increase. This is indeed

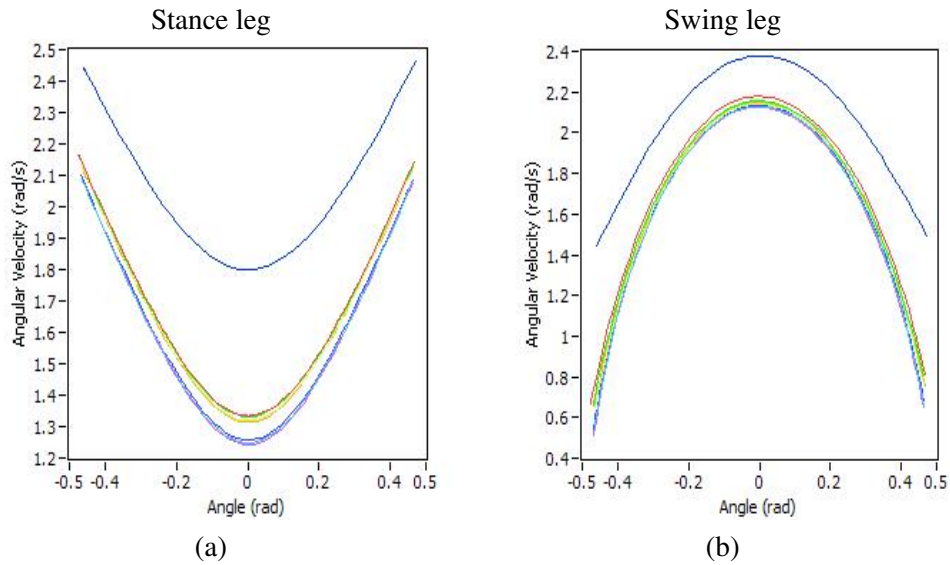


Figure 4.15: Hybrid phase space trajectories for simple flat ground walking. Notice that, barring transients, the trajectories settle into a stable limit cycle comparable to that resulting from traditional techniques.

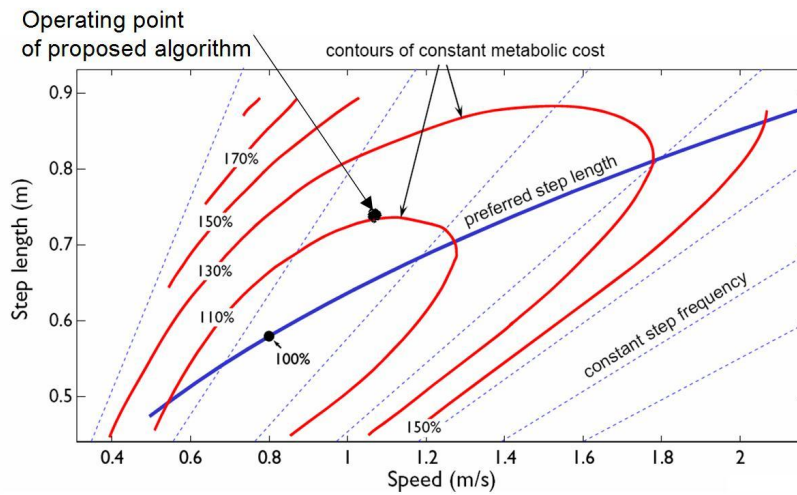


Figure 4.16: Relationship between preferred forward speed and step length in human walking. The blue line depicts empirical data that also minimizes metabolic cost. The absolute minimum and the operating point of our algorithm are depicted by black dots.

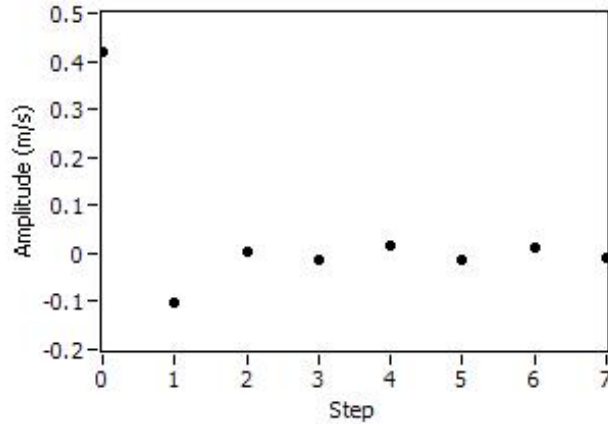


Figure 4.17: Change in momentum (depicted in terms of the difference,  $\dot{\phi}^+ - \dot{\phi}^-$ ) between steps for the base case experiment. Notice that the algorithm leads to a stable choice of a periodic trajectory. This means that there is no change in momentum due to switches between energy level, although there is some baseline energy that must be pumped in to redirect the torso on level ground.

our observation as well and the trajectories cluster around two different orbits according to this change in speed.

Lastly, figures 4.20 and 4.21 depict results for irregular footholds in two dimensions. Due to the wide variations in footholds, each step requires a different trajectory. We believe this to be the first demonstration of actuated passive walking on an irregular sequence of footholds.

The stability or correctness of gaits resulting from the proposed strategy may be characterized by the availability of a nonempty family of feasible trajectories, given a realization of the foothold sequence. The primary constraint that dictates what is achievable is that of energy injection or dissipation. It follows from equation 4.11 that there is a bound on the maximum change in momentum that can be achieved within a single step,

$$\left| \frac{1}{2} m l_1^2 (\dot{\phi}^{+2} - \dot{\phi}^{-2}) \right| \leq \delta E \quad (4.19)$$

Given specific limits on  $\delta E$ , we have a limit on  $\delta \dot{\phi}^2$  which can then be translated into maximum  $x$  and  $y$  variations in step length.

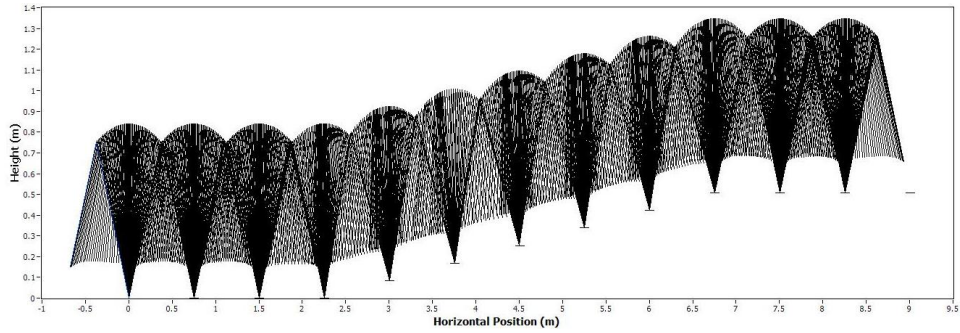


Figure 4.18: Configuration, over time, of 12 steps of the simulated robot climbing a flight of stairs.

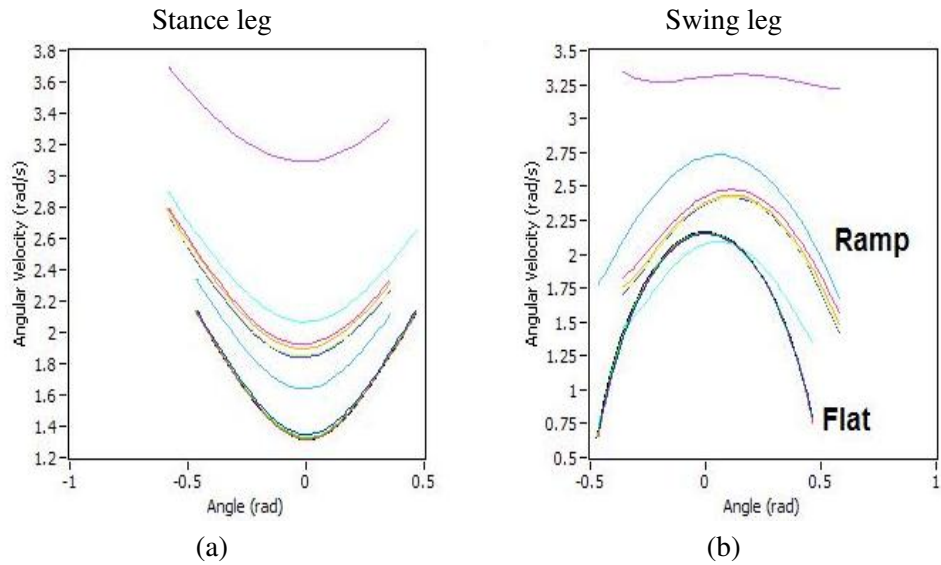


Figure 4.19: Hybrid phase space trajectories for stair climbing. Notice that trajectories separate into two categories, for flat ground and ramp respectively. Also, notice that the trajectory segments corresponding to the ramp are asymmetrical, in terms of the angle ( $\phi, \theta$ ). This is a natural consequence of the geometric configuration of the robot as it transitions between steps.

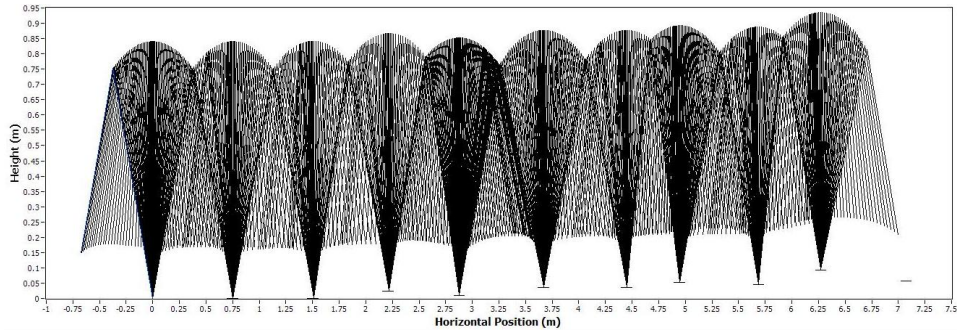


Figure 4.20: Configuration, over time, of 10 irregularly spaced steps. The step length is a Gaussian random variable,  $N(0.75, 0.1)$  resulting here in an interval  $[0.494, 0.799]$  m and step height is  $N(0.0, 0.025)$  resulting here in the interval  $[-0.037, 0.045]$  m.

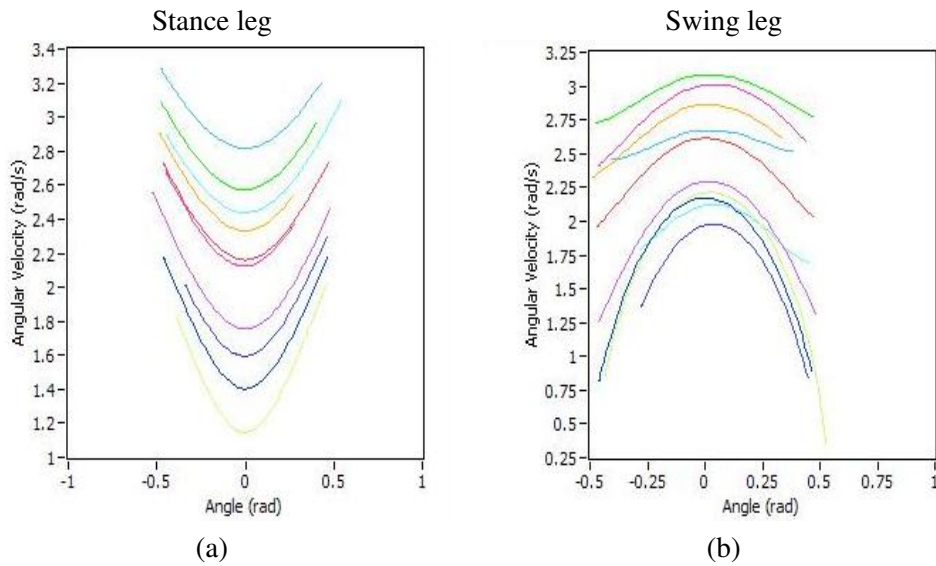


Figure 4.21: Hybrid phase space trajectories for irregularly spaced footholds with variations in 2-dimensions.

## Chapter 5

# Bipedal Walking: Learning the Template-Anchor Embedding

### 5.1 Relationship between Template and Anchor Models

The primary objective of a good trajectory generation algorithm is to efficiently, and to the extent possible - constructively, generate trajectories that satisfy the constraints of the task. When a robot agent begins with little knowledge of system and environmental dynamics, a period of exploratory data gathering is unavoidable. In this setting, the goal is an efficient representation and an algorithm that allows the agent to organize and use the data it has collected such that, eventually, it reaches the point where it can safely and constructively generate trajectories that satisfy all task constraints. This is the primary benefit of learning - the ability to acquire reliable strategies using data gathered directly from imperfect experience.

With dynamical tasks such as bipedal walking, it has been difficult to learn planning and control strategies directly from data, as already described in chapter 2. One of the main reasons for this difficulty is the topological structure of the task. Even for a simple compass gait walker, the discussion in chapter 4 showed that the structure of the strategy is nontrivial - involving coordinated switching between families of trajectories. By the time we address the more complex robot that is really of interest, the stratified structure of phase space trajectories makes learning hard.

When one begins with a relatively clean trace of a single clearly defined behavior, e.g., a periodic gait, it has been possible to extract corresponding low-dimensional



manifolds [53]. There have also been some other attempts to utilize subspaces extracted from Principal Components Analysis to define control laws to enforce periodic trajectories [55, 56]. However, in these works, the learning algorithm essentially mimics a known good solution and the resulting strategies do not support trajectory generation with novel goals. A similar difficulty is faced by many other machine learning approaches. For instance, the success of the Toddler robot [48], which used a version of policy gradient reinforcement learning to quickly adapt to smooth slope changes, was crucially dependent on the fact that the robot was mechanically designed to be stable even when unactuated.

In contrast, we are explicitly interested in beginning with imperfect data, gathered from a system that can explore interesting parts of the phase space but not reliably perform the task (e.g., imagine an infant who is still only able to stumble along a few steps at a time, before learning to walk reliably), and finding algorithmic ways to utilize this data to generate trajectories. This chapter outlines such an algorithmic approach.

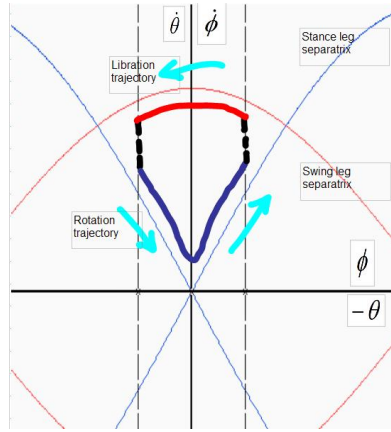
The key idea behind the approach of this chapter is the use of the template model to shape learning of the dynamics. The main problem faced by a research program that tries to learn from a *tabula rasa* is that the problem is highly ill-posed. For instance, learning is ineffective without some *a priori* guidance regarding essential phase-space geometrical and topological issues such as points of high-curvature, intersections, etc. [108, 34]. For our purposes, the template model - which concisely encodes the task without reliance on explicit models of the high-dimensional system, provides this information and efficiently regularizes the ill-posed problem.

The essential relationship between the template and the anchor is defined by the commutativity relationship between the dynamics. In order to explain the significance of this, consider the caricature of the high-dimensional dynamics in figure 5.1.

Beginning with no knowledge of the actual structure of the anchor dynamics - i.e., not knowing the true shape of the manifold, we are still able to collect observational data as the robot performs some imperfect actions. This data is in the nature of sample points or very small trajectory segments - some that will be on the desired manifold and some that will be away from it (i.e., not corresponding to the task at all). Moreover, in the anchor *phase* space, we will have a parameterized manifold structure.

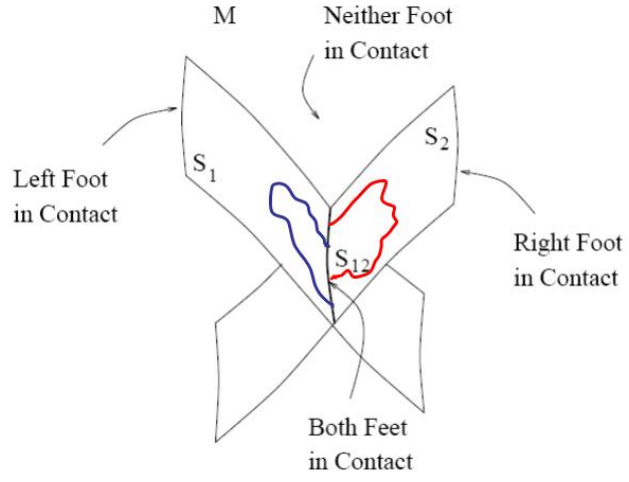
If the robot were executing the task as desired, then the following diagram must

### Template Phase Space



(a)

### Anchor Configuration Space



(b)

Figure 5.1: A conceptual illustration of the relationship between the template and the anchor. Part (a) depicts a typical template trajectory for a single leg, switching between single and double support phases. Part (b) depicts a caricature of how this might map onto the stratified configuration space of the multi-link legged robot. This is just a caricature in that the “sheets” in the configuration manifold are really in much higher dimensions and so the true trajectory is much more complex. However, features such as a switch between trajectory segments and the topology of the curve are shared by both models, due to their dynamical equivalence.

commute:

$$\begin{array}{ccc}
 \mathcal{S}_H & \xrightarrow{\mathcal{M}_H} & \mathcal{S}_H \\
 \downarrow \pi & & \downarrow \pi \\
 \mathcal{S}_L & \xrightarrow{\mathcal{M}_L} & \mathcal{S}_L
 \end{array}$$

What this really means is that any candidate trajectory in the anchor space, when projected to the template space, must be very close to the trajectories generated by the algorithms in chapter 4. So, even though we do not know the structure of the anchor manifold, given a specific trajectory - we can verify whether it is a geodesic on this manifold (if it is, its projection must be the template trajectory between the projected start and end points). So, the trajectory generation problem reduces to that of regularized trajectory approximation. The task manifold is the set of all geodesics that satisfy the commutativity relationship.

Before describing the concrete algorithm, we make the following remark about the

high level methodology. In this chapter, we will describe an algorithmic procedure that makes use of the template model designed in chapter 4. For the task of bipedal walking, and given the specific needs of the experiment described in section 5.3, this is a desirable approach. However, the algorithm that achieves this template-anchor embedding is agnostic towards the details of the template model, except for the requirement that there must be dynamical equivalence in the sense of topology of phase space trajectories. So,  $\mathcal{M}_L$  may be based on any vector field whose effect on the states can be evaluated in the form  $x^+ = f_{\delta t}(x^-)$  (i.e., a short trajectory segment of duration  $\delta t$ ). In particular, if one were to create different template models - via nonlinear control design, via learning, etc. then it is still possible to follow the methodology of this chapter to embed the resulting gait in the higher dimensional system. Correspondingly, the primary requirement regarding  $\mathcal{M}_H$  is that we have a sufficiently dense set of observations  $\langle x^-, \tau, x^+ \rangle_i$ . By continually updating the set of observations, this procedure can naturally handle variations in system and environmental dynamics.

## 5.2 Algorithm for Data-driven Approximation of Task Manifolds

Based on the above argument, we can state the essential idea behind the algorithm concisely: Given a start and end pose (corresponding to a single step), synthesize a trajectory in anchor phase space, using suitable segments drawn from the observed data, such that the template and anchor trajectories are equivalent under projection.

At any given point along the step, it is possible to define an error,

$$e_{comm} = f(\pi(x) - y), x \in \mathcal{S}_H, y \in \mathcal{S}_L \quad (5.1)$$

The projection  $\pi(x)$  can be computed from geometric/kinematic considerations <sup>1</sup>, so this error can be evaluated both point-wise and path-wise. In particular, it is clear that this error can be defined for arbitrary trajectory segments lying anywhere in phase space. If one were to fix a template phase space trajectory (i.e., plan what needs to be done to achieve the upcoming foothold) in  $\mathcal{S}_L$ , then it is possible to associate an error with every possible anchor trajectory segment lying in  $\mathcal{S}_H$ . This induces a vector field in  $\mathcal{S}_H$  in terms of

---

<sup>1</sup>Interestingly, some neurophysiological experiments show that such projections are directly coded for in vertebrate nervous systems [109].

geodesic paths that satisfy the commutativity requirement stated above. Task level control amounts to following this vector field.

Now,  $\mathcal{M}_L$  is explicitly defined in chapter 4 but  $\mathcal{M}_H$  is not known explicitly. We have access to a set of observations of the form  $\langle x^-, \tau, x^+ \rangle_i$  acquired over very small time intervals (e.g.,  $\delta t = 10$  ms), as the robot performs some random sequence of actions in order to explore the phase space.  $\mathcal{M}_H$  is some submanifold in this space - i.e., a subset of our complete set of observations defines the anchor task manifold.

So, the trajectory generation problem is to compute the sequence  $\{x_j\}$ , given the start and end poses for a foot step, such that  $e_{comm}$  is minimized. From kinematic considerations, a foot placement goal corresponds to an end pose. By continuously solving this trajectory generation problem at each foot step (e.g., in the Double Support phase), the robot is able to achieve a sequence of foot placement goals and navigate irregular terrain. Note that the core calculation involves a trajectory from a start to end pose, selecting from a family of possible trajectories. External effects such as impact and dissipation are naturally handled - as perturbations to the start pose from which the computation for the current foot step is performed.

The procedure we adopt is:

1. Allow the robot to perform a sequence of random actions and acquire a collection of points  $\langle x^-, \tau, x^+ \rangle_i$ . In practice, it is often quicker to begin with some *ad hoc* strategy that serves as a seed for reasonable behaviors. However, the procedure will work even for random babbling.
2. Organize observations in the form of a graph. Weight each edge of the graph in terms of the error  $e_{comm}$ .
3. Compute shortest paths in this graph, i.e., geodesics on the manifold, to obtain a sequence of  $x_j$ .

The primary role of  $e_{comm}$  is to filter out spurious paths that do not result in stable walking. If the observations were obtained exclusively from an already experienced stable walker capable of performing all behaviors of interest, then this weight may not be necessary. However, observations are necessarily imperfect and the set of observations includes numerous trajectory segments that may not lead to task achievement.

Also, the graph/manifold may be endowed with the property of convergence to a desired trajectory in the sense that disturbances away from the desired trajectory (a sequence

$\{x_j\}$ ), that moving it to a disturbed state  $\hat{x}$ , can be naturally corrected by following the shortest path from this new state to the goal, via a modified sequence  $\{\hat{x}_j\}$ . This is the notion of (global) *task-level* control.

The nature of the nonparametric approximation problem being solved by the manifold learning algorithm may be understood in terms of the simple conceptual picture in figure 5.2. In this figure, the planned template trajectory, i.e., the sequence  $\{y_j\}$  ( $y_j \in \mathcal{M}_L$ ), is the circular arc. Given an imperfect set of observed behaviors, many sequences  $\{y_i\} = \{\pi(x_i)\}$  would not be included in the set in their entirety. These imperfect observations are depicted in the figure as dashed lines. However, suitable portions of these observed trajectories may be synthesized into a trajectory that approximates the desired plan. As the density of these observations goes up, quality of approximation improves. Spurious trajectories in the set of observations - some realization of the dynamics of the robot that is incompatible with the specifications of the task - will be penalized (i.e., higher  $e_{comm}$ ) more than any of the other observations and will not feature in the computed geodesic. In effect, these spurious observations will be filtered out using the properties of the template model (which is one of the benefits of the proposed scheme over a naive approximation of geodesics).

The problem boils down to that of finding the sequence  $\{x_j\}$ , given a template plan sequence  $\{y_j\}$ :

$$\{x_j\} = \arg \min \sum_{j=1}^N f(\pi(x_j) - y_j), x_j \in \mathcal{S}_H, y_j \in \mathcal{S}_L \quad (5.2)$$

The concrete algorithm is described below in Algorithm 3. The most demanding steps in this algorithm are construction of the nearest neighbor graph and the shortest path algorithm. As stated earlier, this can be achieved for any  $\mathcal{M}_L$  that is compatible with the anchor dynamics. One could use such a weighted graph in two ways. Firstly, one could generate segments of trajectories (such as between start points of the Double Support phase). This becomes a single point-to-point shortest path computation. Alternatively, one could generate trajectories “on-line” - by continuously looking up the shortest path from the current state to the desired state. This enables continuous handling of errors and disturbances as the robot executes the plan.

Figure 5.3 visually depicts the key steps in algorithm 3. Note that this depiction is really a caricature in that the dimensionality of the problem being solved in section 5.3 is much higher.

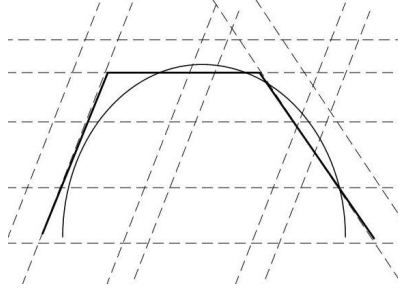


Figure 5.2: Conceptual structure of the approximation problem. The observations are depicted as thin dashed lines - each dash representing a segment of experience. The desired trajectory is the circular arc. In order to achieve the desired trajectory, the system makes use of observed trajectories along the heavy line. Also, note the presence of spurious observed trajectories that could connect start and end points, but do not follow the curve.

---

**Algorithm 3** Nonparametric approximation of  $\mathcal{M}_H$

---

**INPUT:** Observations of  $\{x^-, \tau, x^+\}_i, i = 1, \dots, N$ , Specific realization of  $\mathcal{M}_L$

**OUTPUT:** Weighted graph representing  $\mathcal{M}_H$

Project the observed points ( $y = \pi(x)$ ) to obtain  $\{y^-, y^+\}_i$

Compute inverse covariance matrix,  $\Sigma_H^{-1}$  using the set  $\{x_i^-\}$

Define connectivity graph,  $G = (V, E)$ .

**for all**  $i$  **do**

    Add vertex  $V_i \equiv x_i^-$

**end for**

**for all**  $V_i$  **do**

    Determine  $k$ -nearest neighbors  $V_j$

    using the Mahalanobis distance  $\left(D = \sqrt{(x_i^+ - x_j^-)' \Sigma_H^{-1} (x_i^+ - x_j^-)}\right)$

**for all**  $V_j$  **do**

            Add edge  $E_k(V_i, V_j)$

**end for**

**end for**

**for all**  $E_k$  **do**

    Determine  $e_{comm} = \|y_{des} - \pi(x^+)_j\|$

    where  $y_{des}$  is computed using  $\mathcal{M}_L$

    Set the weight of edge  $E_k$  to  $e_{comm}$

**end for**

---

---

**Algorithm 4** Generate geodesic sequence  $\{x_j\}$ 

---

**INPUT:** Initial and final poses,  $x_{ini}, x_{fin}$ ; Weighted connectivity graph  $G = (V, E)$  (i.e., approximation to the manifold  $M_H$ ), where  $E$  is an edge-list, with  $N$  nodes

**OUTPUT:** Sequence of state-space points,  $\{x_j\}$

Compute Inter-node distance matrix  $D[i][j]$  using Floyd Warshall algorithm:

```
for  $k = 1 : N$  do  
  for  $i = 1 : N$  do  
    for  $j = 1 : N$  do  
       $D[i][j] = \min(D[i][j], D[i][k] + D[k][j])$   
    end for  
  end for  
end for
```

Initialize  $x_0 = x_{ini}$

```
while  $x_{current} \neq x_{fin}$  do  
   $x_j = \arg \min D[current][j]$   
end while
```

---

---

**Algorithm 5** Determination of k-Nearest Neighbors

---

**INPUT:** Given a state  $x^\#$  and a set of observations  $\mathcal{X} = \{x_i\}, i = 1, \dots, N$

**OUTPUT:** Set of observations  $\mathcal{X}_{NN} = \{\hat{x}_k\}$

- (Off-line) Perform k-Means clustering on the set  $\mathcal{X}$ , choosing the number of clusters to be comparable to the cardinality of each cluster. Denote by  $c_j$ , the point in  $\mathcal{X}$  that is closest to the center of the corresponding cluster,  $\mathcal{C}_j$ , where  $j = 1, \dots, M$ .

- (On-line, step 1) Find the closest cluster point for  $x^\#$ ,  $c^\# = \arg \min \|x^\# - c_j\|, j = 1, \dots, M$ , where  $\|\cdot\|$  denotes Mahalanobis distance.

- (On-line, step 2) Find the  $k$  - nearest points in  $\mathcal{C}_k$ , to  $x^\#$ :

```
for all elements  $c_l \in \mathcal{C}_k$  do  
   $\text{Dist}[l] = \|x^\# - c_l\|$   
  Sort Dist and select first- $k$  elements  
end for
```

---

The anchor trajectory is being composed from observed segments. Asymptotically, given sufficiently dense data distribution along the desired subsets on the task manifold, the resulting trajectory will be similar to the true geodesic that might result from computations based on an analytical description of the task manifold. However, in realistic scenarios, this is not going to be the case. The generated paths will be approximations to the true geodesic. As with all planning algorithms, this generated trajectory can be smoothed in various ways, including through the use of nonparametric techniques such as multivariate splines.

A more pressing issue is that of efficiently computing nearest neighbors and shortest paths. The multi-link legged robot used in our experiment involves 8 generalized position variables and corresponding derivatives, so the task manifold is a subset of  $\mathcal{R}^{16}$ . A typical experiment utilizes observations from 15-30 minutes of exploration, yielding 1500-3000 trajectory segments. More data yields better coverage of the space and improved quality of approximation. However, this comes at a computational cost and care is required when working with larger sets of data. Such algorithmic efficiency issues have been extensively studied in the computational geometry literature and dealing with large data sets is a very active research area, see, e.g., [35].

In the following experiments, nearest neighbors are computed hierarchically using an intermediate clustering step, and shortest paths are computed using a Floyd-Warshall type algorithm based on a subsampled version of the complete set of observations (corresponding to 10 minutes of observed data). The structure of these algorithms permit numerous optimizations that we have not explored. For instance, instead of invoking a relatively expensive algorithm such as shortest path distance matrix computation at double support, one could pre-compute several manifolds and their distance matrices as an off-line operation. This would then enable real-time trajectory generation in a very tractable way - involving nothing more than matrix lookup and other elementary operations.

### 5.3 Experiments with a Simulated Multi-link Legged Robot

The experiments described here are based on a physically realistic simulated robot modeled after the Spring Flamingo robot, constructed by Dr. Jerry Pratt while at the MIT Leg Laboratory [73]. In these experiments, we collect data from this simulated robot, implemented in the *Yobotics<sup>TM</sup> Simulation Construction Toolset*. Before describing the structure of the robot and the experiments, we would like to make the following remark. Although this work is done using a simulator, special care was taken to restrict interactions between the



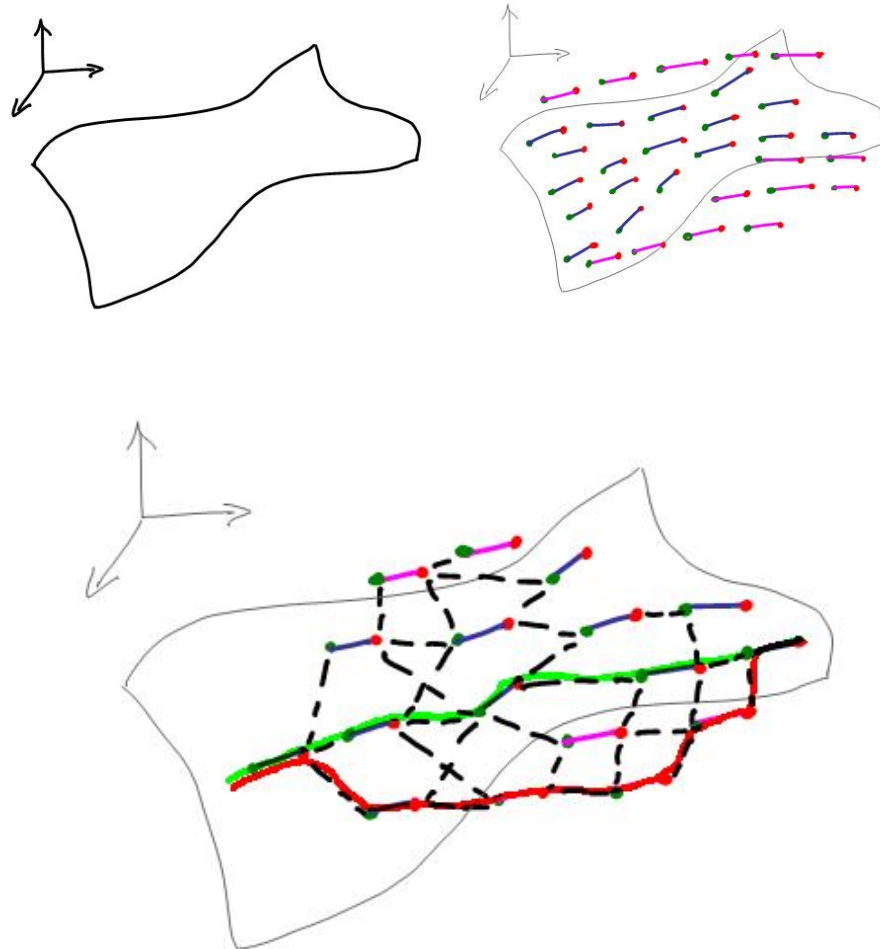


Figure 5.3: *Top*: The nature of the approximation problem. The surface on the left depicts the manifold that needs to be approximated. What we actually have are short trajectory segments, some of which lie on the task manifold (shown in blue) and some that do not (shown in magenta). *Bottom*: The approximate geodesics generated using algorithm 3. The observed trajectory segments are organized into a nearest-neighbors graph (depicted here using only a few edges). By weighting these edges with  $e_{comm}$  and computing weighted shortest paths, we are able to restrict shortest paths to the desired geodesics (shown in green) as opposed to spurious paths (shown in red).

learning program and the simulator to those that would be possible with a real robot. So, the experiment could be repeated - without modifications - on a physical robot, if one were accessible.

The structure of the robot is depicted in figure 4.7. We model this robot in terms of eight configuration variables: torso pitch,  $q_{pitch}$ , torso height from the ground,  $q_z$ , right leg joint angles at hip, knee and ankle -  $q_{rh}, q_{rk}, q_{ra}$  and similarly for the left leg,  $q_{lh}, q_{lk}, q_{la}$ . Correspondingly, there are eight state derivatives,  $\dot{q}_{pitch}, \dot{q}_z, \dot{q}_{rh}, \dot{q}_{rk}, \dot{q}_{ra}, \dot{q}_{lh}, \dot{q}_{lk}, \dot{q}_{la}$ . Let us denote the space of these variables as  $\mathcal{Q} \subset \mathbb{R}^{16}$ .

The *Simulation Construction Toolset* provides abstractions for constructing a robot dynamic simulation, including physical effects such as ground contact models. This enables the simulated robot to be used by a learning/control algorithm in exactly the same way that a real robot might be used.

Some key features of the simulation are summarized below:

- The dynamic model is simulated using an  $O(N)$  algorithm by Featherstone [110].
- Contacts between the robot foot and the ground are treated as a linear spring-mass type interaction.
- Joints between links (e.g., knee) are modeled as Pin Joints, with few exceptions to incorporate “natural elements” such as knee caps (see [73] for more details).
- The physical parameters of the robot are summarized in the table below:

| Element     | Mass(kg) | Length(m) |
|-------------|----------|-----------|
| Torso       | 12       | 0.385     |
| Leg - upper | 0.4598   | 0.42      |
| Leg - lower | 0.306    | 0.42      |
| Leg - foot  | 0.3466   | 0.04      |

Using this simulator, the first step in the experiment is to collect data, as follows:

1. Define a bounding box in phase space that includes the desired task manifold, e.g.,  $[0.5 < q_z < 0.9] \times [-0.35 < q_{pitch} < 0.35]$ . This is a very coarse specification of a region that includes the operating region of interest.
2. Define a procedure that calls the simulation from a typical initial condition and runs until the trajectory exits the defined bounding box. This simulation includes an *ad*

*hoc*, randomized, control strategy that is sufficient to enable the robot to take a few unstable steps although not sufficient to enable it to reliably walk (and certainly not enough to enable it to walk reliably on a sequence of stepping stones).

3. Collect all trajectories that begin and end inside the bounding box, in terms of 10 ms segments. This trajectory includes many state space points that will eventually enable the robot to reliably walk. But, it also includes many that are spurious - these bad points are shown in figure 5.4.
4. Subsample the set of observations for the purposes of shortest path computations. This is done by low-discrepancy sampling. This results in a smaller data set that still provides sufficient coverage. Figure 5.5 depicts the resulting data in terms of projections to the stance leg angle and swing leg angle phase spaces.

With this set of data, we are in a position to apply the approximation algorithms from section 5.2. In order to do this, the robot is given a sequence of state space points corresponding to the double support phases. These poses can be computed by first converting the foot placement goal to a configuration space point (from geometric considerations) and then finding the nearest point in the set of observed state space points to the desired configuration space point.

The result of a typical run of this algorithm is depicted in figure 5.6. This figure depicts the configurations over time (sampled at a few points along the trajectory). Again, note that the generated trajectory is a path in  $S_H \subset \mathfrak{R}^{16}$  but the depiction here only shows the configurations. These three steps correspond to three different step lengths and the algorithm can generate any commanded length from a suitable interval around which it has observations. This is an improvement over many existing machine learning based results that are limited to mimicry of specific temporally ordered observed trajectories, or to local smoothing and improvements. Instead, we are able to utilize a wide variety of noisy trajectories, representing many different gaits, to synthesize a parameterized family of gaits. Figure 5.6 depicts three specific instances from this family.

Due to technical reasons, the trajectories in figure 5.6 are depicted in a different environment from the ones used for figure 4.7, but both are equivalent.

These generated trajectories are dynamically realizable in the following sense.

- They are generated using experimentally observed trajectory segments, so that constraints such as actuator and joint limits, etc. are naturally satisfied (modulo small

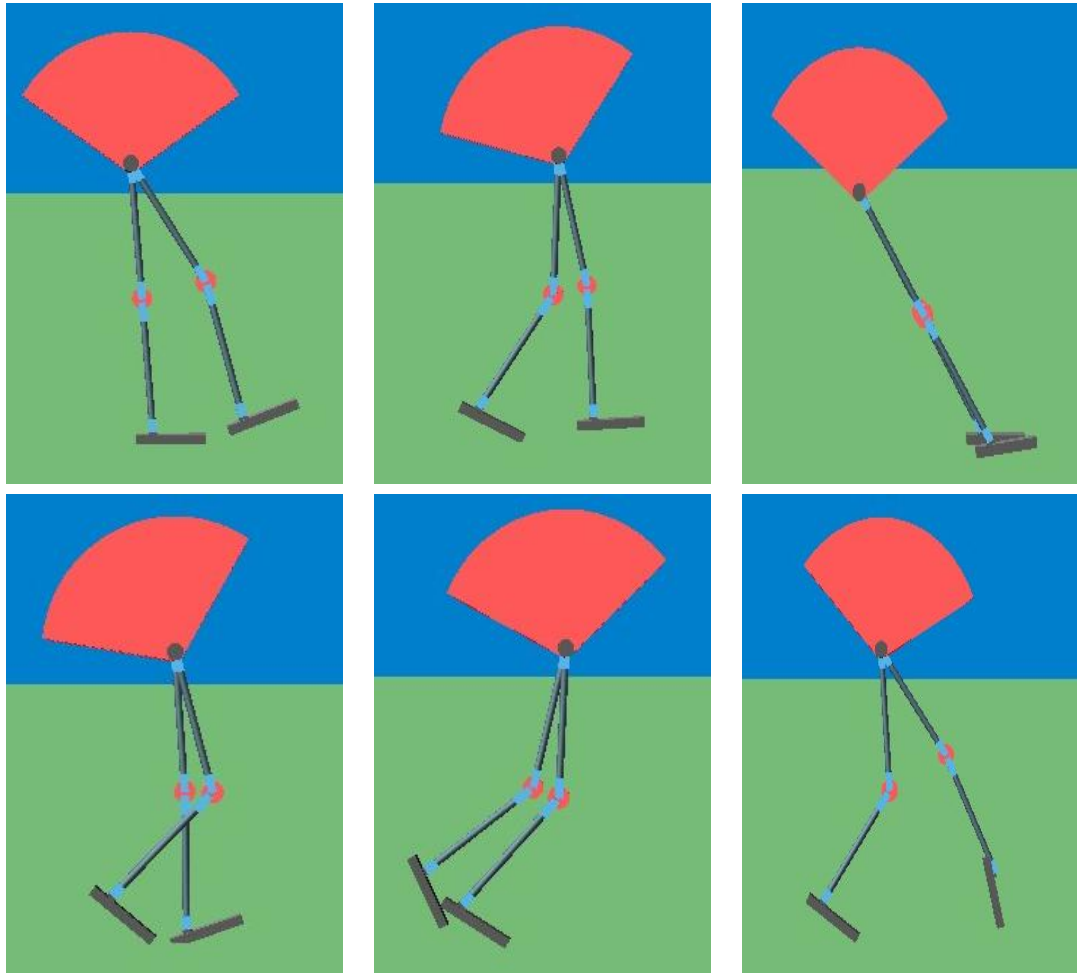


Figure 5.4: A small sampling of configurations (many of them, spurious) that are included in the observed data. Although these depictions are only in terms of the configuration space, the data set actually includes the corresponding velocities for each configuration - to enable planning in phase space.

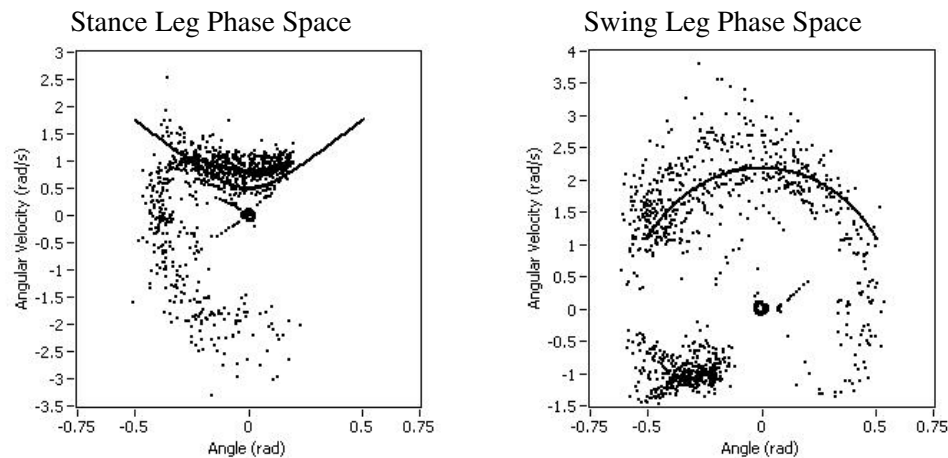


Figure 5.5: Depiction of the subsampled data in the stance leg angle and swing leg angle phase spaces. The start point of each 10 ms trajectory segment is depicted in these phase spaces. Also, a typical compass gait trajectory is overlaid to provide context. Note the many “off-pendular-trajectory” points. These are required in order to ensure dynamically realizable transitions at double support (and correspond to the dotted lines in figure 4.10).

errors dependent on the quality of approximation).

- They are regularized by a template model whose trajectories are known to satisfy all constraints expressible in that abstraction, so the high-dimensional trajectory embodies assumptions such as mostly passive/ballistic swing phases.

In order to illustrate this last point, we compute the “open-loop” torques that would be necessary to achieve trajectories of the form shown in figure 5.6. Then, we compare the result against similar torque profiles from other mostly-passive walkers (designed by Dr. Jerry Pratt, Florida Institute for Human and Machine Cognition). This will demonstrate that not only do the trajectories we generate achieve the result of intersecting desired footfall goals, but in fact this is done without expending much more energy than other minimally actuated gaits that utilize passive dynamic principles.

The resulting torque profile for 5 steps is shown in figure 5.7. Similar torque profiles were generated for three different gaits used for comparison:

- Passive - An intuitive gait, using ballistic swing and other principles, that is designed for flat, level ground walking [73].

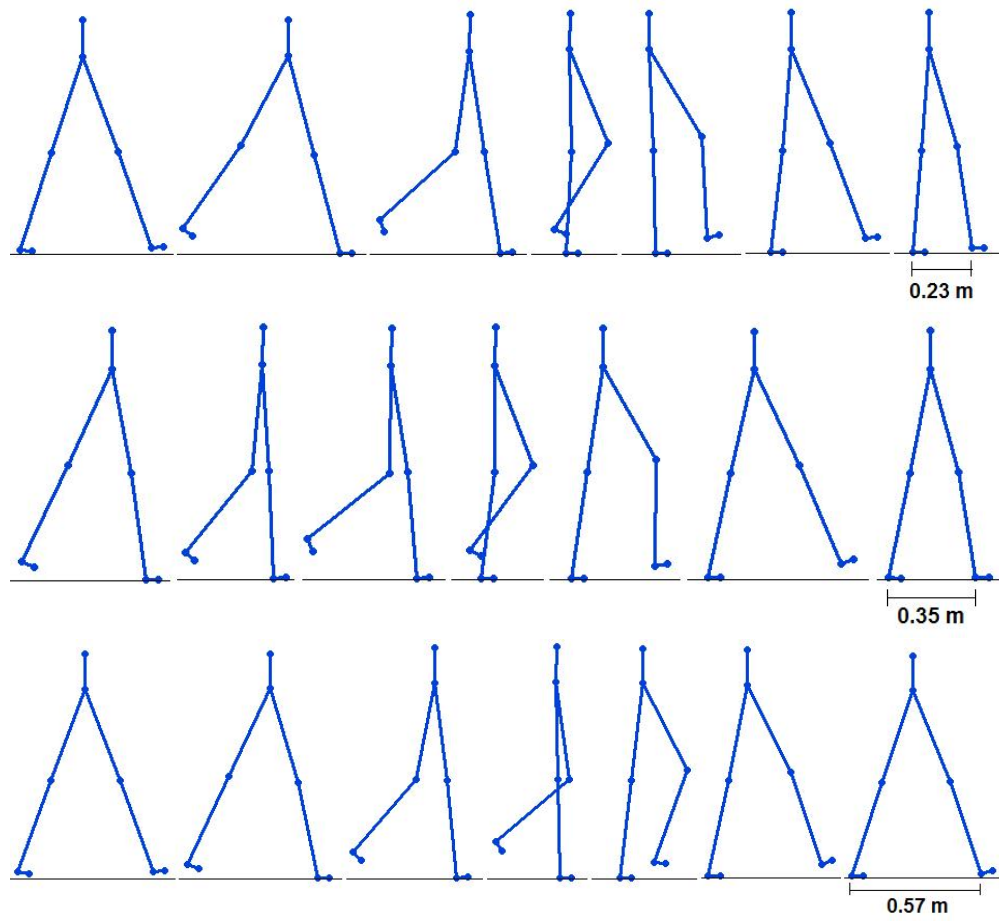


Figure 5.6: Trajectories generated using our algorithm. We have depicted captures from three different trajectories corresponding to different foot placement, and hence end pose, goals.

---

**Algorithm 6** Generate Open-Loop Torque Profile (for each joint)

---

**INPUT:** Trajectory sequence  $\{z_j\}, j = 1, \dots, N, z_j \in S_H$  and a set of observations  $\mathcal{O} = \{x_i^-, T_i, x_i^+\}, x_i \in S_H, T_i \in A_H \subset \mathbb{R}^6, i = 1, \dots, M$  (where  $T$  is a 6-dim vector of torques and  $x$  is a 16-dim vector of states)

**OUTPUT:** Sequence of torques  $\{\tau_j\}$ , time-integrated torque  $\mathcal{T}_{int}$  and energy,  $\mathcal{E}$

**for**  $j = 1 : N - 1$  **do**

Find  $K$  (e.g., = 20) points from the set  $\mathcal{O}$  defined by the distance  $\|z_j - x_i^-\|$ , including a set of  $K$  torques for the chosen joint,  $\{\hat{\tau}_k\}$

**for**  $k = 1$  to  $K$  **do**

With  $d^- = \|z_i - x_k^-\|$  and  $d^+ = \|z_{i+1} - x_k^+\|$ , define the average distance,

$$d_{Mk} = \frac{d^- + d^+}{2} \text{ (where } \|\cdot\| \text{ is the Mahalanobis distance)}$$

**end for**

$$\tau_j = \frac{\sum_1^K \exp(-d_{Mk}) \hat{\tau}_k}{\sum_1^K \exp(-d_{Mk})}$$

**end for**

To compute time-integrated torque:

$$\mathcal{T}_{int} = \sum_1^{N-1} \tau_j \cdot \delta t$$

To compute energy:

Apply Savitzky-Golay filter to de-noise velocity profile

Compute  $\mathcal{E} = \int \tau_j \cdot \dot{z}_j$  (implemented here using Simpson's 3/8 rule).

---

- Passive 2 - A perturbed version of the above gait, that executes a period-2 chaotic orbit, and in the process achieves an irregular sequence of foot placements.
- Fast - A modification of the basic passive gait that includes some active elements such as a minimum-jerk trajectory tracking scheme for the swing leg. Despite these modifications, it is still a highly efficient gait when compared against many other alternatives in the literature.

In tables 5.1 and 5.2, we present several aggregated measures computed from these trajectories.

Firstly, we list the time-integrated torques for the trajectory generated using each algorithm, over a similar period of time. Note that the trajectory generation scheme described in this chapter is the only one among the four that is capable of planned foot placements. So, passive gaits used in this comparison were only capable of smooth and level-ground walking.

A more physically meaningful metric is based on energy. We report the energy expenditure normalized by robot mass and distance traveled - a non-dimensional quantity known as *cost of transport*.

Table 5.1: Comparison of time-integrated torques,  $\mathcal{T}_{int}$  (reported in Nms) over a typical sequence of 5 steps.

|                          | Manifold Approx. | Passive  | Passive Period-2 | Fast    |
|--------------------------|------------------|----------|------------------|---------|
| $\mathcal{T}_{int_{rh}}$ | 16.7627          | 11.5357  | 13.3543          | 16.1479 |
| $\mathcal{T}_{int_{rk}}$ | 7.3352           | 2.6817   | 6.19715          | 9.2198  |
| $\mathcal{T}_{int_{ra}}$ | 18.1009          | 13.9009  | 15.0199          | 9.9699  |
| $\mathcal{T}_{int_{lh}}$ | 14.9915          | 11.0399  | 17.0174          | 15.7704 |
| $\mathcal{T}_{int_{lk}}$ | 6.00374          | 2.83644  | 5.9533           | 9.42043 |
| $\mathcal{T}_{int_{la}}$ | 10.6606          | 13.6244  | 16.4178          | 15.8559 |
| $\mathcal{T}_{intTotal}$ | 73.85464         | 55.61904 | 73.9598          | 76.3843 |

Table 5.2: Comparison of *cost of transport* - computed as total expended energy,  $\mathcal{E}$ , normalized by robot mass,  $m$ , and distance traveled,  $x$ .

|                                             | Manifold Approx. | Passive | Passive Period-2 | Fast    |
|---------------------------------------------|------------------|---------|------------------|---------|
| $cot = \left(\frac{\mathcal{E}}{mx}\right)$ | 1.63588          | 1.3672  | 1.68228          | 1.65216 |

As seen from these comparisons of energetics, the trajectories generated by the manifold approximation procedure of this chapter are on par with modulated versions of passive gaits.

This shows that we have been able to generate trajectories, beginning from imperfect knowledge of the system dynamics and a template model for trajectories, that can reliably achieve a sequence of planned footholds drawn from an interval - and this is being done in an energy efficient manner.



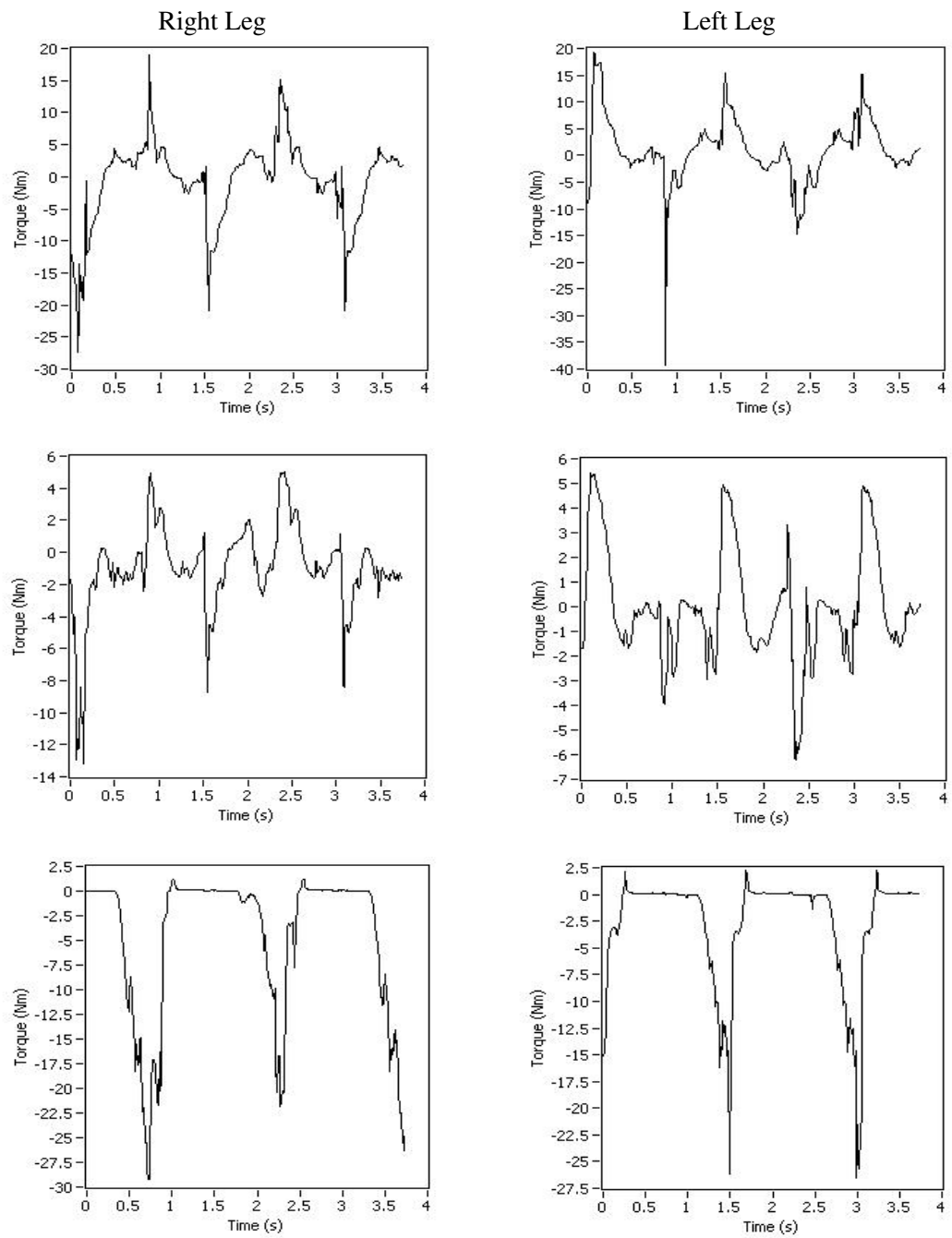


Figure 5.7: Open-loop torques for a sequence of five steps over an irregularly spaced sequence of footholds. From top to bottom, the profiles correspond to hip, knee and ankle torques.

## Chapter 6

# Task Encoding in Abstract Spaces

### 6.1 Introduction

#### 6.1.1 Abstractions and the process of functional biomimesis

In the empirical sciences, a *model* is an encapsulation of some slice of the real world within the confines of the relationships constituting a formal system. According to the eminent physicist James C. Maxwell, “*the success of any physical investigation depends upon the judicious selection of what is to be observed as of primary importance.*” To the extent that the systems of interest act in a *physical* world, the success of the modeling venture is tightly coupled with the specification of the observables and a characterization of the way in which they are linked. For our purposes as roboticists, focusing on systems with inputs *and* outputs, we would augment the set of observables to include variables that can be explicitly affected, i.e., actuated.

Consider a system,  $\mathcal{N}$ , consisting of an abstract state-space  $\Omega$ , together with a set of observables  $f_i : \Omega \mapsto \mathfrak{R}, i = 1, 2, \dots, n$ .

$$\mathcal{N} \equiv \{\Omega, f_1, f_2, \dots, f_n\} \tag{6.1}$$

We would like to have a complete characterization of the evolution of states,  $\Omega$ . However, this is almost never completely accessible. The next best thing is a characterization of the evolution of observables, perhaps in the form of equations whose solutions are

the state-space trajectories,

$$\Phi_i(f_1, f_2, \dots, f_n) = 0, \quad i = 1, 2, \dots, m \quad (6.2)$$

In cases where the observables,  $\{f_i\}$ , are sufficient to reconstruct  $\Omega$ , and when the system of equations 6.2 admits explicit solutions, the resulting scientific theory is essentially equivalent to the physical system being studied. In fact, for most systems of practical interest, the observables are not sufficient to unambiguously reconstruct the true state  $\Omega$  and the equations,  $\Phi(f) = 0$ , involve parameters and effects that cannot be explicitly characterized so that the theory is only valid under specific assumptions. The corresponding process of restricting the domain of applicability of models is one of defining abstractions. This is at the core of the scientific method. The relationship between reality and a model is summarized in the following diagram,

$$\begin{array}{ccc} \Omega & \xrightarrow{\Phi_{\mathcal{N}}} & \Omega \\ \downarrow \pi & & \downarrow \pi \\ \{f_1, f_2, \dots, f_n\} & \xrightarrow{\Phi(f)} & \{f_1, f_2, \dots, f_n\} \end{array}$$

Not knowing  $\Phi_{\mathcal{N}}$  and  $\pi$  explicitly, scientific theories attempt to construct approximations to reality,  $\hat{\Phi}(f)$ , according to several criteria. In the descriptive sciences, the typical goal has been to achieve exact prediction under a very wide set of conditions:  $\hat{\Phi}(f) \equiv \Phi(f)$ .

We have used several abstractions in this dissertation as well. However, our goal has been quite different - for the following reasons:

- Our primary interest is in constructing autonomous agents that can operate in an unstructured environment, with limited knowledge of system and environmental dynamics. So, the observables  $\{f_i\}$  are necessarily incomplete and incapable of completely characterizing the system and its environment.
- The tasks that need to be performed by these autonomous agents are ill-posed (e.g., consider the numerous ways to walk, run or swim). This brings into question the need for quantitatively exact equivalence  $\hat{\Phi}(f) \equiv \Phi(f)$ . Instead, what is often required is a *qualitative* (e.g., topological) equivalence over a very large region of the state spaces.
- The success of the agent is not defined by any single well defined cost function, but instead by a multi-objective notion of safety and reliability.

The essential distinction is that although the robot is a physical entity, operating in a physical world and subject to all the laws that are the focus of the natural sciences, the task assigned to this physical entity and the corresponding strategies it adopts are artificial [19]. Physical laws do not sufficiently constrain the space of such tasks to make the task encoding problem well posed.

In many engineering systems, particularly in structured environments, this has not been an issue because there are numerous additional requirements - time/speed, energy consumption, etc. that can be optimized to yield concrete solutions to specific design problems. On the other hand, in unstructured environments and with ill-posed tasks (such as the bipedal walking problem considered earlier) there is a need for alternative paradigms that yield *least restrictive* encodings of the task. With the emergence of autonomy and *robust intelligence* as important driving forces in the long-term research agenda for robotics, the problem of task encoding takes on greater importance than in the past.

The approach of this dissertation has been that of functional biomimesis, i.e., we borrow concepts from biology at the qualitative level - in terms of low-dimensional template models - which has the effect of regularizing computations involving the more complex robotic system. To the extent that the biological systems we mimic are much more capable than any robot we have been able to build, this provides a much better basis for encoding than traditional cost functions.

The way we have implemented this scheme in the design of the bipedal walking algorithm can still be understood in terms of the above commutative diagram. However,

- The choices of  $\{f_1, f_2, \dots, f_n\}$  and  $\hat{\Phi}(f)$  are driven by descriptive biological theories that are based on a family of equivalent biological systems (i.e., animals) that perform the behavior of interest.
- The equivalence  $\hat{\Phi}(f) \equiv \Phi(f)$  is treated in the sense of an approximation that preserves certain phase-space geometrical structures, crucial for task achievement. In particular, we are less interested in quantitatively exact mimicry, and far more interested in qualitative similarity over a broad operating regime. This is the most important contribution of the biologically inspired “template” model.
- The task specification is naturally achieved in a phase space that is well studied in dynamical systems theory. This then enables the use of conservation laws and invariance properties, e.g., of total system energy, to be the basis for simplification yielding concise representations.

In general, one may not be so fortunate. Many tasks, even when they involve physical robotic systems, are described in terms of requirements that do not have such a direct mapping to dynamical principles. However, it may still be possible to impose a natural dynamical structure by describing the task in terms of manifolds on which trajectories may be constructed. The fact that we do not have readily available parameterizations, e.g., by energy invariance, necessitates tractable algorithmic techniques for trajectory generation on the manifold. In general, this may be nontrivial even for a low-dimensional template. In this chapter, we present approximation algorithms that solve this problem in two stages - (a) cover the manifold with a well-distributed point set or continuous curve that can be generated incrementally and (b) generate trajectories by approximating geodesic paths in terms of this finite set of samples on the manifold. This is a simple conceptual scheme and the latter step was already addressed in chapter 5. However, even when a manifold in configuration or phase-space is specified explicitly, the problem of generating well-distributed points and curves that are restricted to this manifold can be nontrivial. We present novel algorithms to solve this problem.

### 6.1.2 Some useful notions from differential geometry

In the foregoing discussion, we have made good use of the fact that the phase-space trajectories of a robot performing specific dynamical tasks, e.g., stable walking, lie on a manifold. Empirical results with specific robots [53, 54] provide one motivation for utilizing this structure as the primary representation of the task. In fact, there are deeper reasons for tasks to be manifold-structured, briefly discussed in the following subsection. Prior to that, we will quickly summarize some important notions from differential geometry [14] that will be useful throughout the rest of this chapter. Readers conversant with the language of manifold theory may skip this subsection, without loss of continuity.

#### **Definition [Topological Space]**

A set  $X$  is said to be equipped with a topology if there is a family of subsets, specifically *open* sets, satisfying the following requirements:

1. the union of any family of open sets is open,
2. the intersection of finitely many open sets is open,
3. the set  $X$  and its empty subset are open.

Such a set is called a *topological space*.

A set  $V \subset X$  is *closed* if its complement,  $U = X \setminus V$ , is open. A topological space  $X$  is said to be *compact* if each covering  $\{U_\alpha\}$  of it by open sets,  $X = \bigcup_\alpha U_\alpha$ , contains a finite subcover,  $X = \bigcup_{i=1}^N U_{\alpha_i}$ .

Topological spaces that have the structure of a Euclidean space in the neighborhood of each point are called manifolds.

**Definition [Smooth Manifold]**

Let  $M$  be a set of points. We say that  $M$  carries a smooth *atlas* if there is a finite or countable family  $\{U_\alpha\}$  of subsets of  $M$  with the following properties:

1. These sets form a covering, i.e.,  $M \subset \bigcup_\alpha U_\alpha$ .
2. The points of each set  $U_\alpha$  are in one-to-one correspondence with the points of a domain  $V \subset \mathbb{R}^n$ . Therefore, one can introduce local coordinates  $(x_\alpha^1, \dots, x_\alpha^n)$  in  $U_\alpha$  by assigning to a point  $x \in U_\alpha$  the coordinates of the corresponding point in  $V_\alpha$ .
3. In the intersection  $U_\alpha \cap U_\beta$ , the local coordinates  $(x_\alpha^1, \dots, x_\alpha^n)$  and  $(x_\beta^1, \dots, x_\beta^n)$  are related to each other by mutually inverse smooth changes of coordinates,

$$x_\alpha^i = x_\alpha^i(x_\beta^1, \dots, x_\beta^n), x_\beta^j = x_\beta^j(x_\alpha^1, \dots, x_\alpha^n), \quad i, j = 1, \dots, n \quad (6.3)$$

with nonzero Jacobians,

$$\det \left( \frac{\partial x_\alpha^i}{\partial x_\beta^j} \right) \neq 0, \det \left( \frac{\partial x_\beta^i}{\partial x_\alpha^j} \right) \neq 0, \quad (6.4)$$

Figure 6.1 depicts a typical manifold and two overlapping charts.

One of the most useful manifolds in robotics is that of *tangent bundles*.

**Definition [Tangent Bundles]**

Let  $M^n$  be a smooth manifold. Then, the pairs  $(x, v)$ , where  $x \in M^n$  and  $v$  is a tangent vector at the point  $x$ , form a smooth manifold  $TM^n$  of dimension  $2n$ . This manifold is called the *tangent bundle* to  $M^n$ .

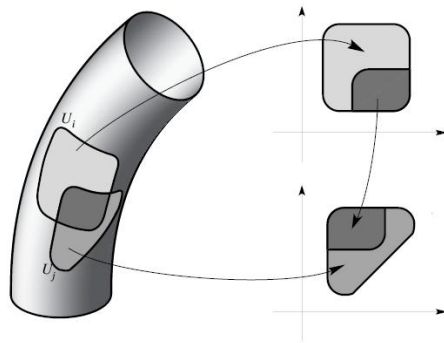


Figure 6.1: Overlapping charts on a nonlinear manifold.

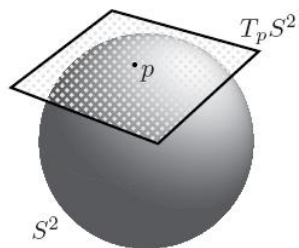


Figure 6.2: Tangent bundle for the sphere,  $S^2$ .

The concept of the tangent bundle, using the example of the sphere, i.e., the manifold  $S^2$ , is depicted in figure 6.2.

One of the major benefits of the definition of constructions such as the tangent bundle is that it enables the definition of intrinsic properties of systems, without regard to the particular parametrization using which it is being describes in any given context.

For instance, the Lagrangian of a physical system (e.g., equation 4.1 that describes the double pendulum) can be written as a function  $L : TM \mapsto \mathfrak{R}$ , that holds true for all parameterizations of the configuration space of the system. In this sense, such maps encode the behavior of the system in a way that is not subject to change with implementation details such as sensor coordinate frames or calibration.

**Definition [Riemannian Manifold]**

If  $M$  is a smooth manifold then by a Riemannian metric  $g$  on  $M$  we mean a smooth assignment of an inner-product to each tangent space of  $M$ . This means that, for each  $p \in M$ ,  $g_p : T_pM \times T_pM \mapsto \mathfrak{R}$  is a symmetric, positive definite, bilinear map, and furthermore the assignment  $p \mapsto g_p$  is smooth, i.e., for any smooth vector fields  $X$  and  $Y$  on  $M$ ,  $p \mapsto g_p(X_p, Y_p)$  is a smooth function.

The pair  $(M, g)$  then will be called a *Riemannian manifold*.

We say that a diffeomorphism  $f : M \mapsto N$  between a pair of Riemannian manifolds  $(M, g)$  and  $(N, h)$  is an isometry provided that  $g_p(X, Y) = h_{f(p)}(d_{f(p)}(X), d_{f(p)}(Y))$  for all  $p \in M$  and  $X, Y \in T_pM$ .

**Definition [Manifold with Boundary]**

A manifold with boundary is a space containing both interior points and boundary points.

Every interior point has a neighborhood homeomorphic to the open  $n$ -ball

$$\{(x_1, x_2, \dots, x_n) \mid \sum x_i^2 < 1\}.$$

Every boundary point has a neighborhood homeomorphic to the “half”  $n$ -ball

$$\{(x_1, x_2, \dots, x_n) \mid \sum x_i^2 < 1, x_1 \geq 0\}.$$

The homeomorphism must send the boundary point to a point with  $x_1 = 0$ .

An important feature of manifolds defined as arbitrary abstract surfaces is that they can be realized as surfaces in a corresponding Euclidean space, stated formally as follows (the proofs may be found in [14]).



**Theorem 6.1.1** *Let  $M^n$  be a compact smooth manifold without boundary. Then there exists an imbedding*

$$f : M^n \mapsto \mathfrak{R}^N$$

*into a Euclidean space of sufficiently large dimension*

**Theorem 6.1.2** *Any compact smooth  $n$ -dimensional manifold  $M^n$  can be immersed into  $\mathfrak{R}^{2n}$  and imbedded (realized as an  $n$ -dimensional surface) in  $\mathfrak{R}^{2n+1}$ .*

In the rest of this section, we will outline some further properties of manifolds - using the case of surfaces in 3-dim. There exist natural generalizations of all these concepts in higher dimensions. For a concrete example of a Riemannian metric, consider a curve lying on a surface in the 3-dim Euclidean space  $\mathfrak{R}^3$ . We might represent the curve in terms of the simple coordinate system  $(x, y, z)$ . The length of this curve  $(u(t), v(t))$  on a surface  $r(u, v)$  may be defined to be equivalent to its length in  $\mathfrak{R}^3$ . So, the metric on this surface is *induced* by the metric of the space  $\mathfrak{R}^3$  containing the surface. The velocity vector for this curve may we written as,

$$(\dot{x}, \dot{y}, \dot{z}) = r_u \dot{u} + r_v \dot{v} \tag{6.5}$$

where  $\dot{x} = x_u \dot{u} + x_v \dot{v}$ , and similarly for  $y, z$ .

We can write this length as,

$$l = \int \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} dt \tag{6.6}$$

Substituting the expressions for  $\dot{x}, \dot{y}, \dot{z}$ ,

$$\dot{x}^2 + \dot{y}^2 + \dot{z}^2 = E\dot{u}^2 + 2F\dot{u}\dot{v} + G\dot{v}^2 \tag{6.7}$$

$$E = \langle r_u, r_u \rangle = x_u^2 + y_u^2 + z_u^2$$

$$F = \langle r_u, r_v \rangle = x_u x_v + y_u y_v + z_u z_v$$

$$G = \langle r_v, r_v \rangle = x_v^2 + y_v^2 + z_v^2$$

These equations can be concisely written in tensor notation as,

$$g_{ij} = \left\langle \frac{\partial r}{\partial x^i}, \frac{\partial r}{\partial x^j} \right\rangle$$

using which the length of the curve can be concisely expressed as,

$$l = \int \sqrt{g_{ij}\dot{x}^i\dot{x}^j} dt \quad (6.8)$$

The above construction yields the *first fundamental form* of the Riemannian metric on a surface.

$$dl^2 = g_{ij}dx^i dx^j = Edu^2 + 2Fdudv + Gdv^2 \quad (6.9)$$

where  $E$ ,  $F$  and  $G$  are functions of the coordinates  $u$ ,  $v$ .

A *geodesic* on the Riemannian manifold is defined as the curve which minimizes the above length. From the calculus of variations [38] we know that a geodesic curve  $\gamma(t)$  is the solution of the Euler-Lagrange equations,

$$\frac{d}{dt} \left( \frac{\partial \gamma}{\partial \dot{x}_j} \right) = \frac{\partial \gamma}{\partial x_j}$$

There exist natural generalizations of these concepts to higher dimensions. For instance, the area of a domain  $U$  on a surface  $r = r(u, v)$  is defined as,

$$\sigma(U) = \iint_U \sqrt{g} du dv$$

Similarly, in higher dimensions,

$$dl^2 = \sum_{i,j=1}^d g_{ij}(u_1, \dots, u_d) du_i^2 du_j^2 \quad (6.10)$$

One of the most important reasons for utilizing this notion of a Riemannian manifold, at least for our purposes as roboticists, is that we are interested in studying curved surfaces in the configuration and phase spaces. So, the notion of *curvature* is of fundamental importance.

The idea of the curvature of a surface is most transparent when the surface is the graph of a function and the point where we wish to determine curvature is a critical point of this function. Consider the Euclidean coordinates in  $\mathbb{R}^3$  such that the point  $x = y = z = 0$  lies on the surface, the surface is the graph of a function  $z = f(x, y)$  in a neighborhood of this point, and the  $z$ -axis is orthogonal to the surface at this point. So,  $x = y = z = 0$  is

a critical point of  $f$  and  $f_x = f_y = 0$  for  $x = y = 0$ . At this point, the first fundamental form is  $dx^2 + dy^2$  (i.e.,  $g_{ij} = \delta_{ij}$ ), and the *second fundamental form* is defined to be the quadratic term of the Taylor expansion of  $f$ ,

$$b_{ij}dx^i dx^j = f_{xx}dx^2 + 2f_{xy}dxdy + f_{yy}dy^2$$

Using this, we define two forms of the curvature. The *Gaussian curvature* is the determinant,

$$K = f_{xx}f_{yy} - f_{xy}^2 \quad (6.11)$$

and the *mean curvature* is the half-trace,

$$H = \frac{1}{2}(f_{xx} + f_{yy}) \quad (6.12)$$

With these definitions, we can characterize various regions on an abstract surface,

- If  $K > 0$ , then the point is a local minimum/maximum and in a neighborhood of this point, the surface lies entirely on one side of the tangent plane. In other words, the surface is either convex or concave.
- If  $K < 0$ , the point is a saddle-point and in any arbitrarily small neighborhood of this point there are points of the surface lying on either side of the tangent plane.

It is important to realize that the above characterization holds for the particular point where  $K$  is evaluated. So, for a surface with  $K < 0$  everywhere, *every* point behaves as a saddle point and the behavior of trajectories may be somewhat non-intuitive to someone used to thinking in terms of Euclidean geometry.

If one were given two different scalar products  $G = (g_{ij})$  and  $B = (b_{ij})$ , corresponding to the first and second fundamental forms of the abstract surface at a point  $r(u_0, v_0)$ , then the following result can be established.

**Theorem 6.1.3** *For any point on a surface there exist coordinates  $x^1$  and  $x^2$  in a neighborhood of the point such that the first and second fundamental forms become*

$$\begin{aligned} g_{ij}dx^i dx^j &= (dx^1)^2 + (dx^2)^2 \\ b_{ij}dx^i dx^j &= k_1(dx^1)^2 + k_2(dx^2)^2 \end{aligned}$$

where  $k_1, k_2$  are the eigenvalues of the pair of scalar products, obtained from the equation  $\det(B - \lambda G) = 0$ , and represent the principal curvatures of the surface at a given point.

**Proof** See [14].

With this, the Gaussian curvature is the product  $K = k_1 k_2$  and the mean curvature is the half-sum  $H = \frac{k_1 + k_2}{2}$ .

For any point of the surface  $r = r(x^1, x^2)$ , the vectors  $r_1 = \frac{\partial r}{\partial x^1}$ ,  $r_2 = \frac{\partial r}{\partial x^2}$  and the normal  $n$  form a basis of the three-dimensional space. The partial derivatives can be written in terms of this basis as  $r_{jk} = \Gamma_{jk}^i r_i + b_{jk} n$ . Since the basis vectors are orthogonal, we obtain that  $b_{ij} = \langle r_{ij}, n \rangle$  - the coefficients of the second fundamental form.

The quantities  $\Gamma_{jk}^i = \Gamma_{kj}^i$  are called the *Christoffel symbols* (which is an example of a *connection*). For this construct, the following equalities hold:

$$\Gamma_{ij}^k = \frac{1}{2} g^{kl} \left( \frac{\partial g_{il}}{\partial x^j} + \frac{\partial g_{jl}}{\partial x^i} - \frac{\partial g_{ij}}{\partial x^l} \right)$$

where  $(g^{ij})$  is the inverse of the Gram matrix  $(g_{ij})$ .

The benefit of defining this construct is that it provides a succinct description of geodesic trajectories on manifolds.

**Lemma 6.1.4** *Given a connection  $\Gamma_{jk}^i$ , a point  $x$  and a vector  $v$  at this point there exists a unique geodesic of this connection starting from  $x$  with initial velocity  $v$ . The geodesic is described by the differential equation,*

$$\frac{d^2 x^i}{dt^2} + \Gamma_{jk}^i \frac{dx^j}{dt} \frac{dx^k}{dt} = 0, i = 1, \dots, n \quad (6.13)$$

**Proof** See [14].

Some further properties of geodesics are:

- On a closed Riemannian manifold, each geodesic  $x(t)$  can be extended unboundedly (i.e., defined for all  $t \in \mathfrak{R}$ ). Manifolds on which all geodesics are extended unboundedly are complete.
- On a complete Riemannian manifold, any two points can be connected by a geodesic.

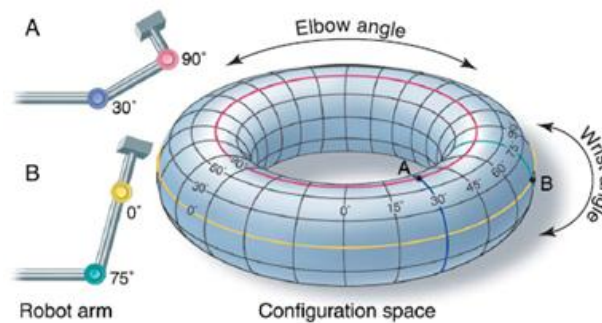


Figure 6.3: The configuration manifold of a typical two-link manipulator.

- Let  $x(t)$  be a geodesic and  $P$  a point on it. Then for any point of  $x(t)$  sufficiently close to  $P$ , the geodesic between  $P$  and this point has the smallest length among all curves joining these points.
- If a curve  $x(T)$  joining two points  $P$  and  $Q$  has the shortest length among the curves joining these points, then this curve is a geodesic.

### 6.1.3 Why are dynamical tasks manifold structured?

With the background provided by the previous subsection, the question of why tasks are manifold structured can be answered in terms of the following high level themes.

#### Structure in the configuration space:

Most robots are composed of translational and rotational elements, and their resultant motion in the environment can be described in terms of these elements. This implies that the configuration spaces of most robotic systems are naturally described in terms of products of spatial motion and rotation groups [14, 13], e.g., the special orthogonal group  $SO(3)$  and the special Euclidean group  $SE(3)$ . Figure 6.3 depicts a simple configuration space arising from rotary motions - for a two-link manipulator. In general, the dynamics of most robots evolve on *Lie groups* and have the structure of both a group and a differentiable manifold. In particular, due to limits on the motion of actuators, the true configuration space is invariably a *manifold with boundary*.

In addition to this, configuration spaces for systems such as bipedal robots have

stratified structure. For a typical biped, the fact that one or the other foot is in contact with the ground splits the manifold into codimension one submanifolds (i.e., contact constraints eliminate a dimension). When both feet are in contact with the ground, the system is on a codimension 2 submanifold of the free configuration space formed by the intersection of the single contact submanifolds. This was abstractly illustrated in Figure 4.8.

### Structure derived from conservation principles:

Often, when *natural (or ballistic) dynamics* are being utilized as an element of the motion plan - the system behavior is *conservative*, i.e., there is an invariant of the motion that is being preserved. When this invariant is the energy, then the system is *Hamiltonian* and the possible set of behaviors of such a system is rich.

The fact that geodesics on manifolds have something to do with conservation principles can be understood by noting that the quantity  $E = \langle \dot{x}, \dot{x} \rangle = g_{ij} \dot{x}^i \dot{x}^j$  is preserved along the geodesic flow of the metric  $g_{ij}$ . However, there are also deeper reasons for such invariance in mechanical systems. The following result (stated here without proof, see [14] for details) is presented to give an idea of how manifolds arise from conservation principles.

Define a Hamiltonian system as a system defined in the position-momentum phase space  $(x, p)$  by the equations,

$$\dot{x} = \frac{\partial H}{\partial p}, \dot{p} = -\frac{\partial H}{\partial x} \quad (6.14)$$

**Theorem 6.1.5 (Maupertuis-Fermat-Jacobi principle)** *If  $(x(t), p(t))$  is a solution to a Hamiltonian system with the Hamiltonian  $H = \frac{p^2}{2m} + U(x) = E$  with the fixed energy  $E$ , then the curve  $x(t)$  is a geodesic of the new metric  $g_{ij} = 2m(E - U(x))\delta_{ij}$ , but the parameter  $t$  on this curve is not natural relative to this metric.*

Much of the work described in chapters 4 and 5 utilized manifolds deriving from this principle. However, realistic tasks require jumps between elements of a parameterized family of such manifolds and the resulting trajectory tends to be complex. Moreover, not having an explicit analytical characterization of these manifolds, e.g., functional forms of  $g_{ij}$ , we developed an indirect approach to constructing geodesics on these manifolds.

### Abstract surfaces:

While the existence of manifold structures corresponding to mechanical systems enjoys the basis of a very deep physical theory, there are numerous other abstract manifolds that are

nonetheless relevant to applications in robotics. To pick an example abstract manifold that is entirely removed from the world of classical mechanics, consider the space of probability distributions [16].

Specifically, consider a submanifold  $P$  of the space of all probability measures on some probability space  $\Omega$ , parameterized by coordinates  $\theta$ , i.e.,  $\{p(\cdot, \theta)\mu \mid \theta \in U \subset \mathbb{R}^n\}$ , where  $\mu$  is a measure on  $\Omega$  and  $p(\cdot, \theta)$  is a probability density function on  $\Omega$  for all  $\theta \in U$ . Then, the Fisher information metric is a natural Riemannian metric on this space:

$$g_{ij} = \int_{\Omega} \frac{1}{p} \frac{\partial p}{\partial \theta^i} \frac{\partial p}{\partial \theta^j} \mu$$

Such a metric is useful in the definition of estimation and stochastic control strategies. For instance, this enables efficient updates to a policy or a model in an unknown environment.

In fact, as we alluded to in subsection 6.1.2, many algebraic surfaces defined by equality constraints are manifolds. The simplest example is that of linear manifolds and equality constraints. Let  $X = \mathbb{R}^n$  and  $M \subset X$ ,

$$M = \left\{ y \in \mathbb{R}^n \mid \sum_i \alpha_i y_i = 0 \right\}$$

$M$  is an  $n - 1$  dimensional manifold. Often, the effect of kinematic constraints results in nonlinear versions of such algebraic surfaces.

The objective of the foregoing discussion is to make the following points:

- The geometric structure of a manifold provides an expressive representation for encoding tasks in robots and autonomous agents - not just for applications that directly involve mechanics, but also for abstract/artificial applications.
- With this expressive language, it becomes possible to develop common algorithmic techniques that can be used in a variety of different domains.

In this scheme, once the task has been specified as a manifold, such that all trajectories on this differentiable structure are *sufficient* for achieving the task at hand - it becomes possible to define algorithms that search for trajectories satisfying various other criteria arising from run-time issues. In the next section, we will present a concrete, illustrative example of an *artificial* task specification and its encoding as a manifold.

Given a manifold that encodes the task and its constraints, generation of feasible trajectories becomes a two step process:

1. **Manifold sampling:** Cover the manifold with a well-distributed point set or curve, that can be generated incrementally according to a parameter, such that the task specification is captured in a finite set of representative elements.
2. **Manifold approximation:** Using the finite set generated above, approximate geodesic paths using shortest path and other graph algorithms, akin to the procedures described in chapter 5.

It turns out that the manifold approximation step is relatively well understood. In small dimensions, the trajectories can be obtained by solving the differential equations 6.13. In general, this may become cumbersome and approximation algorithms such as those for high-dimensional computational geometry [35] prove to be more tractable. However, these techniques first require the manifold sampling problem to be solved. In incremental settings and for general manifolds, this problem is relatively open and there are few good algorithmic solutions available. In section 6.3.1, we will cite recent results regarding the generation of well-distributed point sets. Then, in section 6.4 we will present a novel generalization to the incremental generation of continuous curves, which can be used - among other things, to generate segments such as the experimental observations in figure 5.3.

## **6.2 Encoding the Constrained Evolution of a Polygonal Shape**

Consider the problem of motion planning for metamorphic robots, i.e., robots that can change their shape during the performance of a task. Often, such robots involve a large number of individual units that work together and the problem is really that of decentralized control of the shape of a robot collective. Some typical examples of where this problem arises include formation control [111] and locomotion in fluids, e.g., the geometric swimming problem [112], [113] and its robotic analog [114], [115].

Often, motion planning with such collectives is achieved in one of two ways - (a) design of a set of local rules whose global evolution is expected to achieve a desired result and (b) optimization-based methods working in the high-dimensional configuration space. The former suffers from the disadvantage of lacking explicit control and expressive power to enable use in a precisely programmable (i.e., parameterizable) manner. The latter can often lead to excessive computational complexity.



As with all of the applications described in this dissertation, often the essential task-level goal can be described in a significantly lower dimensional setting and the additional degrees of freedom handled appropriately. Works such as [116] demonstrate the utility of this decomposition - where the large number of units in the collective is not a computational cost but, in fact, a desirable way to drive down an approximation error.

The goal of this section is to provide a simple but nontrivial demonstration of how the task of controlled evolution of shape can be broken down in terms of a template-anchor decomposition, with the template task definition being encoded as a manifold. Then, in section 6.3.2, we will use this parametrization to generate geodesics that represent the template plan for this task.

*Shape spaces* have been studied extensively in various contexts in the statistics and applied mathematics literature. One common definition [15] is based on a normalization that eliminates size, orientation and other effects. While this is useful in recognition and analysis tasks, this is a very inconvenient definition in applications involving robot motion planning. Instead, we wish to deal with the space of curves as they relate to the physical locations of the corresponding modules. In addition, we are interested in constrained shapes, such as the space of curves defined by a constant length or enclosing a constant area, so that we can work the task constraints into the manifold structure.

Concretely, we consider the task of coordinating multiple modules to move in a triangle formation. The constraint is that they must preserve the enclosed area or the outside perimeter. The basic scheme is illustrated in figure 6.4. We have defined the triangle in terms of a normalized base (the line from  $(-1, 0)$  to  $(1, 0)$  - whose length acts as the equivalent of energy in that we can define multiple manifolds using this parameter). The free parameters are  $(u, v)$  as shown. This is Bookstein's parametrization [15] of triangle shapes (except for the length/energy parameter).

Given two triangles we may represent the parameters as complex numbers of the form  $u_j + iv_j$ , and there exists a simple Euclidean distance between the triangles (represented as the complex numbers  $A$  and  $B$ ),

$$d(A, B) = \sqrt{1 - \|A^*B\|^2} \quad (6.15)$$

This distance is a minimizer of the quantity  $\|e^{i\phi}A - B\|$ . Using this, we can write down a globally defined metric that makes the manifold Riemannian.

If a triangle is perturbed slightly through the movement of  $u + iv$  then we may

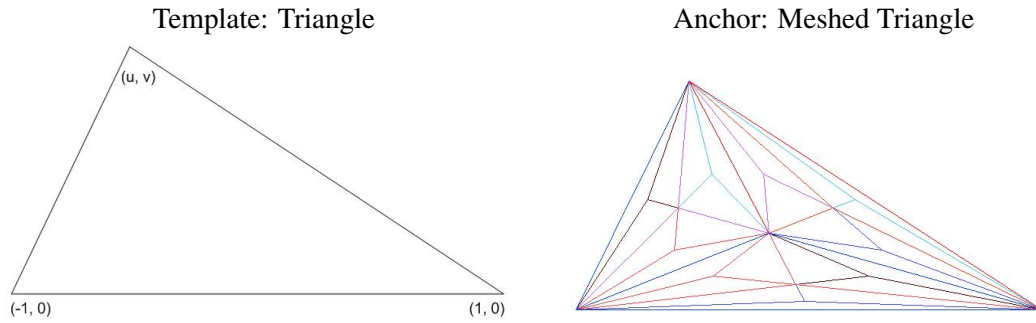


Figure 6.4: The template and the anchor in shape spaces. A module may be located at each vertex in the meshed triangle and the location of each of these vertices may be expressed entirely in terms of the template parameters  $(u, v)$ . In this way, given a sequence of points  $\{u_i, v_i\}$ , it is possible to induce the corresponding higher-dimensional sequence of spatial locations of the modules. The task is encoded in terms of a manifold representing this template parametrization - whose geodesics yield  $\{u_i, v_i\}$ .

compute the distance between the triangle and its perturbed neighbor using the normed vector distance. Define the coordinates of a normalized unit-area triangle and a slightly perturbed version as  $\tau_1$  and  $\tau_2$  respectively, where,

$$\tau_1 = \begin{bmatrix} (1 - \frac{u}{3}) - i\frac{v}{3} \\ (-1 - \frac{u}{3}) - i\frac{v}{3} \\ \frac{2}{3}u + i\frac{2}{3}v \end{bmatrix}, \quad \tau_2 = \begin{bmatrix} (1 - \frac{u+du}{3}) - i\frac{v+dv}{3} \\ (-1 - \frac{u+du}{3}) - i\frac{v+dv}{3} \\ \frac{2}{3}(u + du) + i\frac{2}{3}(v + dv) \end{bmatrix} \quad (6.16)$$

These coordinates are normalized by the area, i.e.,  $\sqrt{v}$  to yield unit-area triangles (this is where we may introduce the length/energy parameter to define a family of manifolds). Using this, the distance may be expressed as a function of  $(u, v, du, dv)$  and this function may be expanded as a Taylor series.

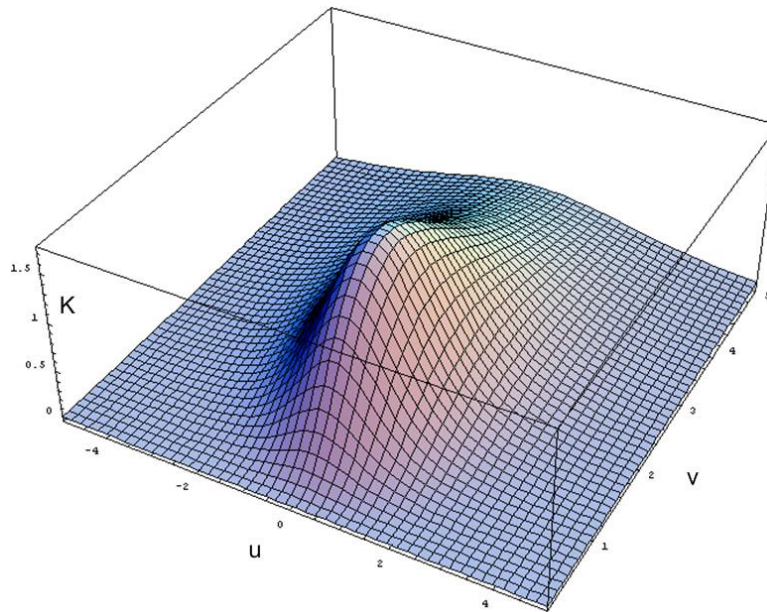


Figure 6.5: Gauss curvature function for the space of unit area triangles.

This expansion provides the following global metric.

$$\begin{aligned}
 ds^2 &= E(u, v)du^2 + 2F(u, v)dudv + G(u, v)dv^2 & (6.17) \\
 E(u, v) &= \frac{2(3 + u^2)}{3v(3 + u^2 + v^2)} \\
 F(u, v) &= \frac{u(-3 - u^2 + v^2)}{3v^2(3 + u^2 + v^2)} \\
 G(u, v) &= \frac{u^4 - 2u^2(-3 + v^2) + (3 + v^2)^2}{6v^3(3 + u^2 + v^2)} & (6.18)
 \end{aligned}$$

To show that even for this simple example, the behavior of geodesics has a nontrivial structure with respect to the chosen parametrization, we compute the Gaussian curvature function,

$$\kappa = \frac{72v^3}{(3 + u^2 + v^2)^3} \quad (6.19)$$

The shape of this function is visualized in figure 6.5.

This shows that the space is distinctly non-Euclidean. So, for instance, the distance

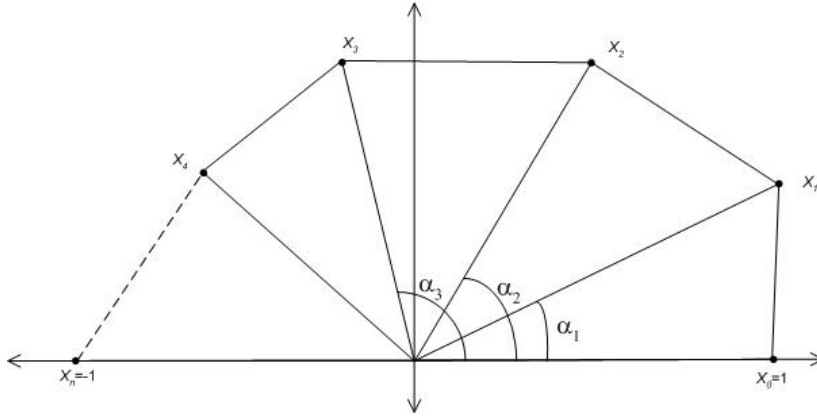


Figure 6.6: Parametrization of  $n$ -gons within a connected component containing star-convex polygons - the polygon is defined as a sequence of vertex-angle pairs  $\{(x_i, \alpha_i)\}$ .

between a triangle and its perturbed neighbor (measured in terms of the metric) varies much more rapidly near  $(0, 2)$  than it does at other far away points. So, the shortest path between two shapes is not a straight line in the sense of the parametrization of Bookstein's coordinates. Instead, it is a curved path that minimizes length according on the Riemannian manifold. Such a path will be computed in subsection 6.3.2.

The above computation may also be repeated for the case of unit-perimeter triangles. For this, the normalization in equation 6.16 would be the the perimeter,  $2 + \sqrt{(u-1)^2 + v^2} + \sqrt{(u+1)^2 + v^2}$ . Again, as with the case of unit-area triangles, it becomes possible to define a family of constant-perimeter manifolds that may be used in computations.

An extension to the case of  $n$ -gons takes us into a space with more complex topology [15]. A parametrization within a single connected component of the space of  $n$ -gons is shown in figure 6.6. This connected component consists of star-convex shapes that can be defined in terms of the vertex-angle pairs  $(x_i, \alpha_i)$ . Then, the area of this polygon can we written as,

$$A = \frac{1}{2} \sum_{n=0}^{N-2} x_n x_{n+1} \sin(\alpha_{n+1}) \quad (6.20)$$

Using this expression, it becomes possible to repeat the calculations shown above in order to derive a global metric for a Riemannian manifold.

The broad drift of the above argument is that it is possible to encode tasks, with no *natural* definition of dynamics, in terms of a manifold that then imparts useful structure

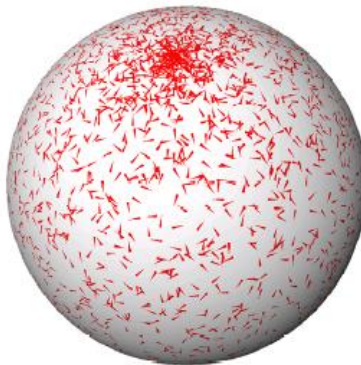


Figure 6.7: A finite set of vectors, generated by uniform sampling in the natural parameterization, depicted as they are distributed on the surface of a sphere. Notice that despite the significant number of points sampled, there are still fairly large pockets uncovered, and correspondingly there are several places where samples are wasted by repetition.

to enable efficient planning and trajectory generation. In the following sections, we will discuss how to compute trajectories on such manifolds.

### 6.3 Approximation of Geodesics on Task Manifolds

Given a manifold that encodes the task and corresponding constraints, the main technical issue is that of generating trajectories on these manifolds. The standard approach represented by equation 6.13 can be cumbersome in high-dimensions and with a fairly general class of manifolds. Instead, for the purposes of robotics applications, one could adopt a numerical approach based on first sampling the manifold - to yield a point set of representative states, and then approximating geodesics using this finite set. A naive sampling on such manifolds, in terms of the standard parameterizations used in typical applications, will yield a point set that is not entirely representative. For instance, figure 6.7 shows a finite set of vectors that are generated by uniform sampling the parametrization of a sphere. This is just a simple 2-manifold and the distribution is already far from uniform. In general, in high dimensions, this problem of non-uniformity becomes much more severe. Often, even more sophisticated parameterizations (e.g., the Mercator projection for the sphere) may be conformal but not area-preserving - which is the real requirement for uniformity.

This suggests a need for better methods for sampling on manifolds, to enable subse-

quent operations such as trajectory generation. In this section, we will present results from recent work on incremental generation of well-distributed (in the sense of *low-discrepancy*) point sets on manifolds [117], which will be used to generate approximate geodesics. In the next section, we will present a generalization of the results of [117] to the case of continuous curves.

### 6.3.1 Well-distributed point sets

The approach we take to defining “well-distributed” point sets is to utilize low discrepancy sequences. Low discrepancy point sets and sequences [118] have a successful history within robotics. They have been used in sampling based motion planning and area coverage applications [119]. In recent work [120], techniques have been proposed for generating sequences in an incremental fashion, which is often a very important requirement. However, the generation of these sequences is based on a computationally expensive search for an optimal ordering.

The essence of the approach of this section is as follows. In order to optimally sample a manifold, we first define a scheme to optimally sample a much simpler object, such as a rectangle in Euclidean space, and then we define a diffeomorphism that maps this onto the manifold. In fact, in order for the resulting scheme to work, the diffeomorphism will need to be *fair* according to certain criteria. The procedure for achieving this, and the definition of fairness are discussed in the following subsections.

#### On the notion of low-discrepancy point sets and sequences

The definition of discrepancy of a finite set  $X$  was introduced to quantify the homogeneity of finite-dimensional point sets [121]:

$$D(X) = \sup_R |m(R) - p(R)| \quad (6.21)$$

In equation (6.21),  $R$  runs over all  $d$ -dimensional rectangles  $[0, r]^d$  with  $0 \leq r \leq 1$ ,  $m(R)$  stands for the Lebesgue measure of  $R$  and  $p(R)$  is the ratio of the number of points of  $X$  in  $R$  and the number of all points of  $X$ . The lower the discrepancy the better or more homogeneous is the distribution of the point set. The discrepancy of an infinite sequence  $X = \{x_1, x_2, x_3, \dots, x_n, \dots\}$  is a new sequence of positive real numbers  $D(X_n)$ , where  $X_n$  stands for the first  $n$  elements of  $X$ .

There exists a point set of given length that realizes the lowest discrepancy. It is known (the Roth bound [122]) that the following inequality holds true for all finite sequences  $X_n$  of length  $n$  in the  $d$ -dimensional unit cube.

$$D(X_n) \geq B_d \frac{(\log n)^{\frac{d-1}{2}}}{n} \quad (6.22)$$

$B_d$  depends only on  $d$ . Except for the trivial case  $d = 1$ , it is unknown whether the theoretical lower bound is attainable. Many schemes to build finite sequences  $X_n$  of length  $n$  do exist that deliver a slightly worse limit,

$$D(X_n) \geq B_d \frac{(\log n)^d}{n} \quad (6.23)$$

There are also infinite sequences  $X$  with the above lower bound, equation (6.23), for all subsequences consisting of the first  $n$  elements. The latter result leads to the definition of low-discrepancy infinite sequences  $X$ . The inequality (6.23) must be valid for all subsequences of the first  $n$  elements, where  $B_d$  is an appropriate constant.

Many low-discrepancy sequences in  $d$ -dimensional unit cubes can be constructed as combinations of 1-dimensional low-discrepancy sequences. Popular low-discrepancy sequences are based on schemes introduced by Corput [123], Halton [124], Sobol [125] and Niederreiter [121].

One of the primary motivations for investigations into these sequences arises from high-dimensional function approximation and Monte-Carlo integration. In this setting, there is a well-known [121] relationship between integrals  $I$ , approximations  $I_n$ , and an infinite sequence  $X = \{x_1, x_2, \dots, x_n, \dots\}$  in  $d$ -dimensions, known as the Koksma-Hlawka inequality:

$$|I(f) - I_n(f)| \leq V(f)D(X_n) \quad (6.24)$$

$$I(f) = \int_0^1 f(x)dx \quad (6.25)$$

$$I_n(f) = \frac{1}{n} \sum_{i=1}^n f(x_i) \quad (6.26)$$

where  $V(f)$  is the variation of the function in the sense of Hardy and Krause.

### Mapping low-discrepancy point sets in Euclidean spaces to Riemannian spaces

The question arises as to whether an area-preserving map up to a constant factor from  $[0, 1]^2$  onto an abstract surface  $S$  can be found. This provides an approach to sampling over  $S$ , using low-discrepancy sequences generated in  $[0, 1]^2$ .

For sufficiently smooth bijective functions  $f(u, v)$  and  $g(u, v)$  the determinant of the Jacobian represents the ratio between the two area elements. Based on such mappings, a set of points defined in  $[0, 1]^2$  could be transformed onto the abstract surface, preserving low discrepancy in the sense of the metric defined for the surface. The following result from [117] summarizes this.

**Theorem 6.3.1** *Let  $\Psi(u, v)$  be a nonnegative function on  $[0, 1]^2$  where  $\Psi^2$  is continuously differentiable. Let  $\Psi(u, v)$  be positive with exception of a set  $L$  of points  $(u, v)$  of Lebesgue-measure 0. For  $(u, v) \notin L$ , let*

$$f(u, v) = \frac{1}{\int_0^1 \Psi(u, v) du} \int_0^u \Psi(u, v) du \quad (6.27)$$

$$g(u, v) = g(v) = \frac{1}{\int_0^1 \int_0^1 \Psi(u, v) dudv} \int_0^1 \int_0^v \Psi(u, v) dudv \quad (6.28)$$

Furthermore, let the functions  $f$  and  $g$  be extendable to continuously differentiable mappings defined on  $[0, 1]^2$ . Then the extension functions  $(f, g)$  define a diffeomorphism of  $[0, 1]^2 \setminus L$  where the following equality is valid for a constant  $c$ ,

$$\frac{\partial f(u, v)}{\partial u} \frac{\partial g(u, v)}{\partial v} - \frac{\partial f(u, v)}{\partial v} \frac{\partial g(u, v)}{\partial u} = c\Psi(u, v) \quad (6.29)$$

Note that  $\Psi(u, v)$  is the volume element.

**Proof** See [117].

### 6.3.2 Algorithm for manifold sampling-based approximation of geodesics

Figure 6.8 depicts a geodesic in the space of unit-area triangles computed by directly solving the differential equations 6.13.

Although in this case we have been able to compute such a geodesic, in general this can be cumbersome. The alternate numerical approach is to approximate the path from a sample point set on the manifold. Such an approximate geodesic is depicted in figure 6.9.



---

**Algorithm 7** Algorithm for sampling on a manifold  $S$ 

---

**INPUT:** Abstract surface  $S$ ,  $\Psi(u, v)$  (satisfying requirements of theorem 6.3.1,  $x(u, v)$ , a known embedding of  $S$  in  $\mathbb{R}^m$  and  $D$ , a well-distributed set in  $[0, 1]^2$ .

**OUTPUT:** Well-distributed point set on  $S$ .

Construct a diffeomorphism  $(f(u, v), g(u, v))$  of  $[0, 1]^2 \setminus L$  according to the above theorem.

Compute the inverse transform  $(f^{-1}(u, v), g^{-1}(u, v))$ .

Output the image of  $D$  under the transform  $x(f^{-1}(u, v), g^{-1}(u, v))$

---

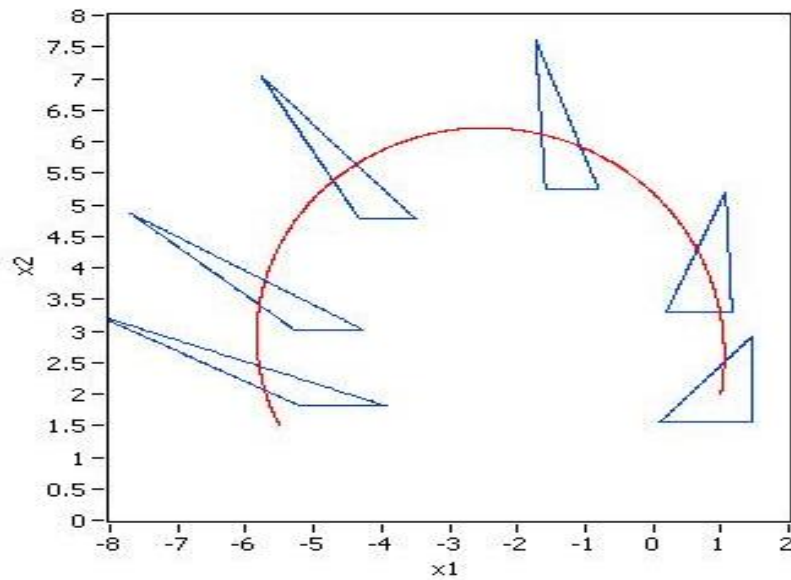


Figure 6.8: A typical geodesic, indicating a path for continuous shape evolution, in the space of unit-area triangles. The generalized variables are  $x^1 = u, x^2 = v$ .

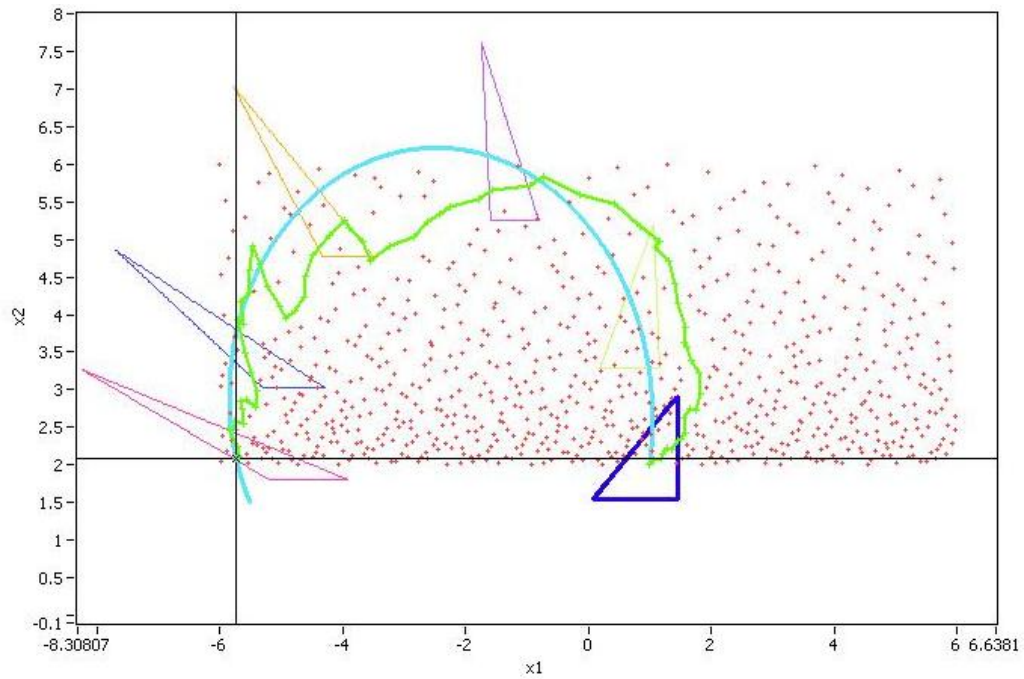


Figure 6.9: Approximate geodesics computed as shortest paths using a well-distributed landmark points in the shape space. Depicted on this figure are low-discrepancy sample points distributed according to a Riemannian metric, the true geodesic (in blue) and the approximate geodesic (in green). Also, some points on the true geodesic include the triangle shape overlaid on the plot.

For this problem, the inverse transformations (as per the conditions of theorem 6.3.1 are,

$$f^{-1}(u, v) = u, g^{-1}(u, v) = \frac{v_0 v_1}{v_0 + v_1 - v}$$

where  $v \in [v_0, v_1]$ .

---

**Algorithm 8** Algorithm for approximating geodesics on a manifold  $S$

---

**INPUT:** A finite set  $\mathcal{X}$  of well-distributed sample points lying on the abstract surface  $S$ , a Riemannian metric on  $S$ .

**OUTPUT:** Sequence of points  $\{x_i\}, x \in \mathcal{X}$ .

Form a nearest-neighbors graph (weighted by distance according to the Riemannian metric) using the same procedure as in algorithm 3.

Compute geodesics using the procedure of algorithm 4.

---

## 6.4 Well-distributed Curves on Riemannian Manifolds

The approximation procedure for computing geodesics shown in the previous section is generic and can be applied to a variety of different manifolds and domains. However, there are applications where we wish to approximate in terms of curves, or curve segments, rather than point sets. The procedure outlined in chapter 5 is one example of this. In this section, the results of [117] are extended to the case of continuous curves. Although the resulting curves have many similarities to those in figure 5.3, it is important to remember that here we explicitly know the template manifold we are computing with. We are adopting this procedure because, in abstract spaces, even this computation may not be feasible analytically. However, the algorithm is agnostic to whether we are dealing with template or anchor models in the sense of the earlier discussions.

### 6.4.1 Low-discrepancy curves in the unit square

One of the earliest known quasi-random sequences is the Richtmyer sequence [126, 127], which illustrates a simple but general result in ergodic dynamics [128, 129]. Let  $x_n = \{n\alpha\}$  (i.e.,  $[n\alpha] \bmod 1$ ) and  $X = \{x_1, x_2, \dots, x_n, \dots\}$ , where  $\alpha = (\alpha_1, \dots, \alpha_d)$  is irrational and  $\alpha_1, \dots, \alpha_d$  are linearly independent over the rational numbers. Then for almost all  $\alpha$  in  $\mathbb{R}^d$  and for positive  $\epsilon$ , with the exception of a set of points that has zero Lebesgue measure,

$$D(X_n) = O\left(\frac{\log^{d+1+\epsilon} n}{n}\right) \quad (6.30)$$

The Richtmeyer sequence is probably the only quasi-random sequence based on a linear congruential algorithm [130]. This is useful because it suggests a natural extension to the generation of curves. We will now provide such an extension.

Let  $C$  be a given piecewise smooth and finite curve in the unit square  $S$ . Furthermore, let  $R$  be an arbitrary aligned rectangle in  $S$  with lower left corner  $(0, 0)$ . Let  $L$  be the length of the given curve in  $S$  and  $l$  be the length of the sub-curve of  $C$  that lies in  $R$ . In case of well-distributed curves, the ratio  $l/L$  should represent the area  $A(R)$  of  $R$  reasonably well. This gives rise to the following definition of discrepancy of a given finite piecewise smooth curve in  $S$ :

$$D(C) = \sup_R \left| \frac{l}{L} - \frac{A(R)}{A(S)} \right| \quad (6.31)$$

It would be desirable to construct curves  $C$  with the property that the discrepancy is always small. More precisely, we will call an infinite and piecewise sufficiently smooth curve  $C : \mathfrak{R}^+ \mapsto S$ , in natural parametrization, a low-discrepancy curve if for all positive arc lengths  $L$  the curves  $C_L = C/[0, L]$  satisfy the inequalities (the function  $F$  must be defined appropriately):

$$D(C_L) \leq F(L) \quad (6.32)$$

In fact, a piecewise smooth curve in natural parametrization generates sequences  $\{x_1, x_2, \dots, x_n, \dots\}$  by setting  $x_n = C_n(n\Delta)$  where  $\Delta$  is a fixed positive number. The inequality 6.32 lets us hope for a similar formula for the derived sequence  $\{x_1, x_2, \dots, x_n, \dots\}$ . Because of equation 6.30, a realistic goal is:

$$F(L) = O\left(\frac{\log^{3+\epsilon} L}{L}\right), d = 2 \quad (6.33)$$

We have to show that this goal is attainable.

To this end, for  $\alpha = (\alpha_1, \alpha_2)$ , let  $C_{\mathcal{A}}(\alpha)$  be the piecewise linear curve  $(t\alpha_1 \bmod 1, t\alpha_2 \bmod 1) = (\{t\alpha_1\}, \{t\alpha_2\})$  where  $t$  is in  $\mathfrak{R}^+$ .

In fact, we can define three classes of curves in the unit square, as shown in figure 6.10. For the simplest type of curves, let us call this class  $C_{\mathcal{A}}$ , the right-left and top-bottom edges are identified so that the curve jumps from one edge to the other upon hitting it. In

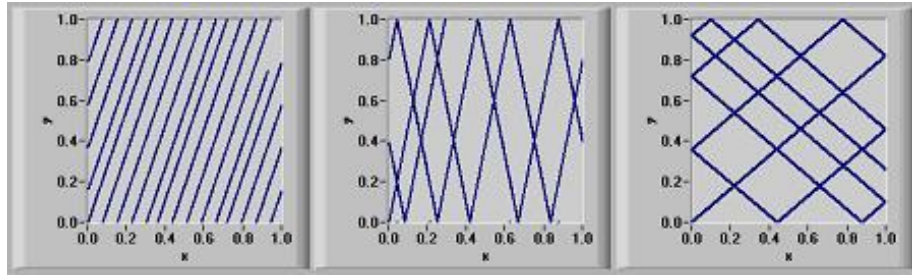


Figure 6.10: Various low-discrepancy curves for the unit square:  $C_A, C_B, C_C$  from left to right.

this scheme, all curves are parallel and continue indefinitely when the square tiles the plane. In the second type of curves, class  $C_B$ , we introduce reflections at the top and bottom edges but preserve the same identification between right and left edges. The third type of curves, class  $C_C$ , involves reflections on all edges. The latter curve is continuous.

**Theorem 6.4.1** For almost all numbers  $\alpha$  in  $\mathbb{R}^2$ ,  $C_A(\alpha)$  is a low-discrepancy curve in the sense of equations 6.32 and 6.33.

**Proof** In order to prove this statement, we will establish that the ratio of the length of a curve segment to the total length is commensurate with the corresponding ratio of the area of an axis-aligned rectangle to the unit square that contains it, as suggested in equation 6.31.

Without loss of generality, we assume  $\alpha_1, \alpha_2 > 0$ .  $C_A(\alpha)$  intersects the axes at  $(x = 0, y_n = \{\frac{n\alpha_2}{\alpha_1}\})$  and  $(x_n = \{\frac{n\alpha_1}{\alpha_2}\}, y = 0)$ , where  $n$  is an arbitrary natural number. For almost all  $\alpha_1, \alpha_2$  all three of the quantities  $(\alpha_1, \alpha_2)$ ,  $\frac{\alpha_1}{\alpha_2}$  and  $\frac{\alpha_2}{\alpha_1}$  generate low-discrepancy sequences in the sense of equation 6.30, in  $\mathbb{R}^2$ ,  $\mathbb{R}$  and  $\mathbb{R}$  respectively. In other words, the aforementioned sequences  $x_n, y_n$  form low-discrepancy sequences in  $[0, 1]$ .

Now, for the class of curves  $C_A$ , all curve segments between points of intersection with the edges of the square are parallel to each other. So, by reasoning about the distribution of these points of intersection, we may arrive at conclusions about the distribution of the curves themselves. With this in mind, we will define the average curve length, i.e.,  $A(R)$ , in the form of integrals.

Let  $\tan \phi = \frac{\alpha_2}{\alpha_1}$  and  $[0, a] \times [0, b]$  be a rectangle with  $0 < a, b < 1$ . Depending on the relative values of  $(\alpha_1, \alpha_2)$ ,  $a, b$  the integrals take on specific forms. We will explain the case when  $\frac{b}{a} \leq \tan \phi$ ,  $b < 1 - \tan \phi$  (see Figure 6.11) in some detail, the other cases being similar.

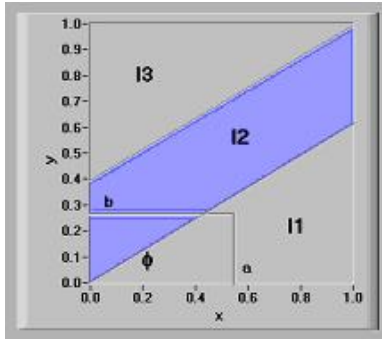


Figure 6.11: Definition of the integrals  $I_1$ ,  $I_2$  and  $I_3$

We divide the unit square into three parts, as shown in figure 6.11. Then,  $I_1$ ,  $I_2$  and  $I_3$  are real numbers that represent the average length that the  $(\alpha_1, \alpha_2)$  lines corresponding to these regions have in common with the rectangle  $[0, a] \times [0, b]$ . Asymptotically, with curve length  $L \rightarrow \infty$ , these quantities may be represented as follows (based on simple geometric considerations),

$$I_1 = \int_0^{a - \frac{b}{\tan \phi}} \frac{b}{\sin \phi} dx + \int_{a - \frac{b}{\tan \phi}}^a \frac{a - x}{\cos \phi} dx = \frac{ab}{\sin \phi} - \frac{b^2 \cos \phi}{2 \sin^2 \phi} \quad (6.34)$$

$$I_2 = \int_0^b \frac{b - y}{\sin \phi} dy = \frac{b^2}{2 \sin \phi} \quad (6.35)$$

$$I_3 = 0 \quad (6.36)$$

$$I_1 \sin \phi + (I_2 + I_3) \cos \phi = ab \quad (6.37)$$

The final term stands for the average length that  $(\alpha_1, \alpha_2)$  lines in  $[0, 1] \times [0, 1]$  with slope  $\tan \phi$  have in common with the rectangle  $[0, a] \times [0, b]$ . As expected, it is exactly the area of the rectangle.

In practice, with a finite length curve, what is the discrepancy? We can estimate this using the Koksma-Hlawka inequality 6.24. For instance,  $I_1$  is approximated using finite length curve segments of the form,

$$l_1(x_i) = \begin{cases} \frac{b}{\sin \phi} & : 0 \leq x_i \leq a - \frac{b}{\tan \phi} \\ \frac{a - x_i}{\cos \phi} & : a - \frac{b}{\tan \phi} \leq x_i \leq a \\ 0 & : a \leq x_i \leq 1 \end{cases}$$

Using a finite number,  $n$ , of such segments, we have the discrepancy,

$$\left| I_1 - \sum_{i=1}^n l_1(x_i) \right| = O\left(\frac{\log^{2+\epsilon} n}{n}\right) \quad (6.38)$$

where  $x_i = \left\{ \frac{i\alpha_1}{\alpha_2} \right\}$

The sum stands for the length of that part of the given curve that lies in  $[0, a] \times [0, b]$ . The exact same argument may be made for the integrals  $I_2$  and  $I_3$ . This means that a finitely generated curve segment of type  $C_{\mathcal{A}}$  also has a very low discrepancy. Moreover, note that we are reasoning about a 1-dimensional sequence of points of intersection. So, the constant 3 (for  $d = 2$ ) in equation 6.33 is now replaced with 2 ( $d = 1$ ). ■

We can now develop two variations of Theorem 6.4.1.

**Theorem 6.4.2** *For almost all numbers  $\alpha$  in  $\mathfrak{R}^2$ ,  $C_{\mathcal{B}}(\alpha)$  and  $C_{\mathcal{C}}(\alpha)$  are low-discrepancy curves in the sense of equations 6.32 and 6.33.*

**Proof** We begin with a remark. One can show that Theorem 6.4.1 is still valid when in the definition of low-discrepancy curves a broader class of rectangles  $R$  is considered, i.e., rectangles where we replace  $[0, a] \times [0, b]$  with the more generic  $[c, a] \times [d, b]$ . The literature on geometric discrepancy [118] includes several related proofs and expanded discussion on such ideas.

*Curves of type  $\mathcal{B}$ :*

Such a curve can be translated into an equivalent version acting in  $[0, 1] \times [0, 2]$ , by simply mirroring the square. To this end, reflections at the upper edge (see Figure 6.10) are ignored. What results is an equivalent scheme as type  $\mathcal{A}$  in  $[0, 1] \times [0, 2]$ . For almost all choices of  $\alpha$ , the resulting curve in  $[0, 1] \times [0, 2]$  is low-discrepancy. The relation between the original space and the new one is straightforward. The original curve goes through a rectangle  $R = [0, a] \times [0, b]$  if and only if the derived curve in  $[0, 1] \times [0, 2]$  goes through  $[0, a] \times [0, b]$  or through  $[0, a] \times [2 - b, 2]$  (see the remark at the beginning of this proof). The latter implies that  $C_{\mathcal{B}}(\alpha)$  satisfies equations 6.32 and 6.33.

*Curves of type C:*

We essentially repeat the arguments from type B. For almost all  $\alpha$ , curves of type B in  $[0, 2] \times [0, 1]$  are low-discrepancy. Such curves can be generated when reflections at the right edge are ignored. The mirrored version of this curve goes through a rectangle  $R = [0, a] \times [0, b]$  if and only if the original curve in  $[0, 2] \times [0, 1]$  goes through  $[0, a] \times [0, b]$  or  $[2 - a, 2] \times [0, b]$  (see the remark at the beginning of this proof). The latter implies that  $C_C(\alpha)$  satisfies 6.32 and 6.33. ■

Curves  $C_C(\alpha)$  can be regarded as first examples of continuous trajectories in a unit square that offer low-discrepancy behavior. In real area coverage scenarios they are highly efficient compared to alternate techniques, as we will demonstrate later in this section.

Finally, these results can be generalized to higher dimensions, using the same style of argument. We state this theorem without proof.

**Theorem 6.4.3** *For almost all numbers  $\alpha$  in  $\mathfrak{R}^d$ ,  $C_A(\alpha)$ ,  $C_B(\alpha)$  and  $C_C(\alpha)$  are low-discrepancy curves in the generalized sense of equations 6.32 and 6.33 in  $d$ -dimensional unit cubes.*

In practice, the question arises as to how these curves can be realized on digital computers. For this purpose, it is quite reasonable to assume that we have the ability to generate rational numbers of user-specified arbitrary precision. In this case, the error due to the rational approximation of  $\alpha$  in theorem 6.4.1, 6.4.2 and 6.4.3 is correspondingly small so that finite lengths of the resulting curves generate discrepancies that are very low.

**Theorem 6.4.4** *If  $\{n\alpha\}$  generates a low-discrepancy sequence, for some irrational number  $\alpha$ , then  $\exists N \gg 1$  such that  $\{n\frac{p}{q}\}$ ,  $n = 1, \dots, N$ , also generates a low-discrepancy sequence, where  $p, q \in \mathfrak{N}$ .*

**Proof** The Hurwitz theorem in number theory [131] states that there are infinitely many  $\frac{p}{q}$  with the property  $|\alpha - \frac{p}{q}| < \frac{1}{\sqrt{5}q^2}$ .

Given any irrational number, it is possible to find a sufficiently large  $q \in \mathfrak{N}$  such that the error of the rational approximation is small. Let  $q_1$  be such a number, with  $q_1 > N \gg 1$ . Then,

$$\left| n \left( \alpha - \frac{p_1}{q_1} \right) \right| < q_1 \cdot \frac{1}{q_1^2} = \frac{1}{q_1} < \frac{1}{N} \quad (6.39)$$

Now, assume that  $\{(n\alpha) \bmod 1\}$  generates a low-discrepancy point set, of discrepancy  $D(\alpha)$  for  $n = 1, \dots, N$ . When  $\alpha$  is irrational,  $\{(n\alpha) \bmod 1\}$  does not intersect



the vertices of the unit square. Then, there always exists a neighborhood where elements of  $\{(nx) \bmod 1\}$ ,  $n = 1, \dots, N$ , are continuous functions of  $x$ . This implies that  $D(x)$  is a continuous function of  $x$  in this neighborhood.

By selecting suitable  $p_1, q_1$  that approximate  $\alpha$  well, to within  $\epsilon = \frac{1}{q_1}$ , we arrive at the bound,  $|D(x) - D(\alpha)| < \delta$ , where  $\delta$  is a suitably small constant. This implies that a curve constructed using these sequences, according to the procedure shown in theorem 6.4.1, and using rational approximations to  $\alpha$ , is still low-discrepancy. ■

## 6.4.2 Fair mappings between Euclidean and Riemannian spaces

The primary objective of the above discussion is to define a scheme for generating well-distributed continuous curves that may be lifted to abstract surfaces. In general, the parametrization of a given surface as it appears in applications will only rarely respect the stated requirements of fairness and low-discrepancy. In situations where the parametrization does not preserve low-discrepancy, we may suitably modify it via the use of a diffeomorphism.

Given an abstract surface  $S$  with a Riemannian metric defined for  $(u, v)$  in  $[0, 1]^2$ ,

$$ds^2 = E(u, v)du^2 + F(u, v)dudv + G(u, v)dv^2 \quad (6.40)$$

where  $E(u, v), F(u, v), G(u, v)$  are differentiable functions in  $u$  and  $v$  and  $EG - F^2$  is positive. The area element  $dA$  is defined by,

$$dA = \sqrt{E(u, v)G(u, v) - F^2(u, v)}du \wedge dv \quad (6.41)$$

The function,

$$\Psi(u, v) = \sqrt{E(u, v)G(u, v) - F^2(u, v)} \quad (6.42)$$

is nonnegative in  $[0, 1]^2$  and  $\Psi^2(u, v)$  is differentiable.

Let  $\alpha = (\alpha_1, \alpha_2)$  be a given irrational vector (direction) in  $\mathfrak{R}^2$ . According to the definitions 6.40 and 6.41, line and area elements of  $S$  for a specific direction  $(du, dv) = (\alpha_1 du, \alpha_2 du)$  (i.e., with  $dv = \frac{\alpha_2}{\alpha_1} du$ ) satisfy,

$$\frac{ds}{du} = \sqrt{E(u, v)\alpha_1^2 + F(u, v)\alpha_1\alpha_2 + G(u, v)\alpha_2^2} \quad (6.43)$$

$$\frac{dA}{du^2} = \sqrt{E(u, v)G(u, v) - F^2(u, v)}\alpha_1\alpha_2 \quad (6.44)$$

From this, we define the quantity  $Q$  which describes our notion of fairness,

$$Q = \frac{\frac{ds}{du}}{\frac{dA}{du^2}} = \frac{\sqrt{E(u, v)\alpha_1^2 + F(u, v)\alpha_1\alpha_2 + G(u, v)\alpha_2^2}}{\sqrt{E(u, v)G(u, v) - F^2(u, v)}\alpha_1\alpha_2} \quad (6.45)$$

**Definition: Low-Discrepancy Curve** A piecewise smooth curve  $C : \mathbb{R}^+ \mapsto S$  lying on an abstract surface  $S$  is called a low-discrepancy curve based on a vector  $\alpha = (\alpha_1, \alpha_2)$ , iff

1.  $C$  is  $S$ -filling, i.e.  $C$  comes arbitrarily close to any point of  $S$ .
2. There is a parametrization of  $S$  where  $Q$  in equation 6.45 is constant for all  $(u, v)$ .
3. In any regular point of  $C$ , the tangent vector is parallel to  $\alpha = (\alpha_1, \alpha_2)$ .

---

**Algorithm 9** Generation of low-discrepancy curves on a manifold  $S$

---

**INPUT:** A finite set  $\mathcal{X}$  of well-distributed sample points lying on the abstract surface  $S$ , a Riemannian metric on  $S$ .

**OUTPUT:** Sequence of points  $\{x_i\}, x \in \mathcal{X}$ .

Find a parametrization of  $S$  that satisfies conditions 2 and 3 in the definition of low-discrepancy curves. Also see the remark below.

Generate a curve in  $S$  based on the image of a low-discrepancy curve in the unit square according to Theorem 6.4.2.

---

**Remark** The parametrization in step 2 is not unique. In all examples an originally given natural parametrization is modified using replacements  $u \mapsto h(u)$  or  $v \mapsto h(v)$  where  $h$  is a smooth diffeomorphism.

An abstract  $d$ -dimensional surface is defined by,

$$ds^2 = \sum_{i,j=1}^d g_{ij}(u_1, u_2, \dots, u_d) du_i^2 du_j^2 \quad (6.46)$$

where the matrix consisting of  $g_{ij} : [0, d]^d \mapsto \mathfrak{R}$  is always symmetric, differentiable, and positive semi-definite. An embedding of an abstract space as in equation 6.46 in an  $m$ -dimensional Euclidean space is a diffeomorphism  $f$  of the hypercube  $[0, 1]^d$  with  $f_1(u_1, u_2, \dots, u_d), f_2(u_1, u_2, \dots, u_d), \dots, f_m(u_1, u_2, \dots, u_d) : \mathfrak{R}^d \mapsto \mathfrak{R}^m$  where the Riemannian metric of this embedding is described by equation 6.46. Usually, this definition is too restrictive. Instead, local diffeomorphisms, i.e., coordinate patches, should be used where these patches cover the whole space under consideration.

Let  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)$  be a given vector (direction) in  $\mathfrak{R}^d$ . According to equation 6.46, the line and volume/content elements for a specific direction  $(du_1, du_2, \dots, du_d) = (\alpha_1 du, \alpha_2 du, \dots, \alpha_d du)$  are:

$$\frac{ds}{du} = \sqrt{\sum_{i,j=1}^d g_{ij}(u_1, u_2, \dots, u_d) \alpha_i \alpha_j} \quad (6.47)$$

$$\frac{dV}{du^n} = \sqrt{\det(g_{ij}(u_1, u_2, \dots, u_d)) \alpha_1 \alpha_2 \dots \alpha_d} \quad (6.48)$$

From this,

$$Q = \frac{\frac{ds}{du}}{\frac{dV}{du^n}} = \frac{\sqrt{\sum_{i,j=1}^d g_{ij}(u_1, u_2, \dots, u_d) \alpha_i \alpha_j}}{\sqrt{\det(g_{ij}(u_1, u_2, \dots, u_d)) \alpha_1 \alpha_2 \dots \alpha_d}} \quad (6.49)$$

**Definition** A piecewise smooth curve  $C : \mathfrak{R}^+ \mapsto S$  in the given Riemannian space  $S$  is called low-discrepancy curve based on a vector direction  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)$  iff

1.  $C$  is  $S$ -filling, i.e.  $C$  comes arbitrarily close to any point of  $S$ .
2. There is a parametrization of  $S$  where  $Q$  in equation 6.49 is constant for all  $(u_1, u_2, \dots, u_d)$ .
3. In any regular point of  $C$ , the tangent vector is parallel to  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_d)$ .

**Remark** In higher dimensions, i.e.,  $d \geq 3$ , the above definition may yield a very large class of solutions which may need to be further specialized through the use of additional requirements, e.g., see section 6.4.4.

### 6.4.3 Examples of the coverage of various surfaces

We will demonstrate the use of the proposed technique to cover various surfaces, chosen to correspond to geometrical shapes and surfaces that are commonly encountered in robotics. We will present visualizations in the Euclidean space.

#### Covering the surface of the unit cube

Consider the surface of a unit cube. This is the most natural generalization of the unit square in section 6.4.1. Figure 6.12 depicts an “opened out” version in the plane of the paper, a transformed version of the which can be used to tile the plane and a low-discrepancy curve drawn on this tiling.

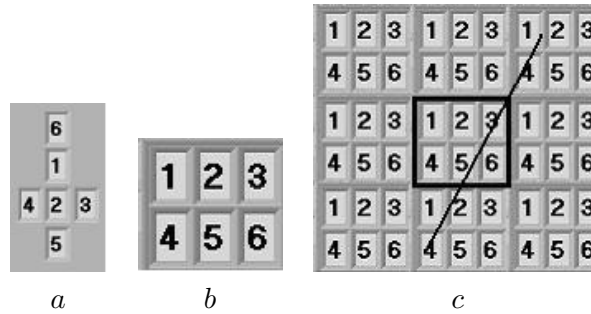


Figure 6.12: Tiling of the plane and relation to the surface of the unit cube. Part *a* depicts the faces of an opened out cube (imagine a typical packing box). Part *b* depicts an equivalent version, rearranged in such a way that it is easier to tile the plane. Part *c* shows the low-discrepancy curve traversing the tiled and flattened cubes. The key point to note is that whenever two edges of a face in part *c* touch, they will also touch in the 3-dimensional cube. The depicted curve is a prescription of the sequence of faces, and points on that face, to visit.

#### Covering the annulus

Consider an annulus, i.e., a ring, that has a standard parametrization given by  $x(u, v) = (u \cos v, u \sin v)$ , where  $0 \leq u_0 \leq u \leq u_1$  and  $0 \leq v \leq 2\pi$ .

Let us consider a diffeomorphism such that  $u \mapsto g(u)$  and  $v \mapsto v$ . Then the ring can be re-parameterized by  $x(u, v) = (g(u) \cos v, g(u) \sin v)$ . Now, we can apply the method developed earlier. Let  $g$  be sufficiently smooth, where  $g$  maps  $[u_0, u_1]$  onto  $[u_0, u_1]$ . Then,

according to our proposed method,  $g(u)$  must satisfy an ordinary differential equation (for  $\alpha_1, \alpha_2 > 0$ ),

$$g'(u) = \frac{\alpha_2 g(u)}{\sqrt{c^2 g^2(u) \alpha_1^2 \alpha_2^2 - \alpha_1^2}} \quad (6.50)$$

where  $g(u_0) = u_0$  and  $g(u_1) = u_1$ , and,

$$\frac{ds}{du} = \sqrt{g'^2(u) \alpha_1^2 + g^2(u) \alpha_2^2} \quad (6.51)$$

$$\frac{dA}{du^2} = g(u) g'(u) \alpha_1 \alpha_2 \quad (6.52)$$

Equation 6.50 has the closed form solution,

$$\frac{\alpha_1}{\alpha_2} \left\{ \sqrt{c^2 g^2(u) \alpha_2^2 - 1} + \arctan \left( \frac{1}{\sqrt{c^2 g^2(u) \alpha_2^2 - 1}} \right) \right\} = u + D \quad (6.53)$$

where  $D$  is the unknown constant of integration. Using boundary conditions that arise from the domain and image constraints of the mapping,  $g(u_0) = u_0$  and  $g(u_1) = u_1$ , it follows that,

$$\frac{\alpha_1}{\alpha_2} \left\{ \sqrt{c^2 u_0^2 \alpha_2^2 - 1} + \arctan \left( \frac{1}{\sqrt{c^2 u_0^2 \alpha_2^2 - 1}} \right) \right\} = u_0 + D \quad (6.54)$$

$$\frac{\alpha_1}{\alpha_2} \left\{ \sqrt{c^2 u_1^2 \alpha_2^2 - 1} + \arctan \left( \frac{1}{\sqrt{c^2 u_1^2 \alpha_2^2 - 1}} \right) \right\} = u_1 + D \quad (6.55)$$

We solve for  $c, D$  and then use equation 6.53 as an implicit definition of the required diffeomorphism. In general, obtaining such explicit solutions could become tedious and difficult. In such cases, one may solve the nonlinear differential equations numerically using shooting methods [132].

Figure 6.13 shows a resulting low-discrepancy curve filling the given ring for the specific case where  $u_0 = 1, u_1 = 2$ . The parameters  $\alpha_1, \alpha_2$  and  $c$  were chosen appropriately by numerical experimentation.

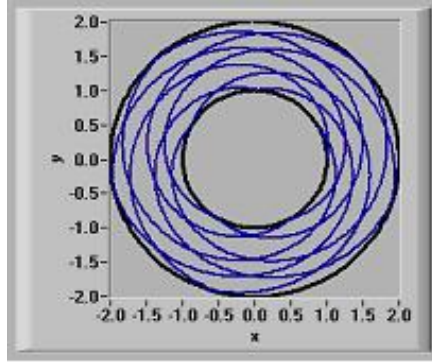


Figure 6.13: A low-discrepancy curve in the annulus.

### Covering the surface of a torus

The torus is one of the most important non-trivial surfaces that appear in robotics, especially when working with configuration spaces of mechanical systems, e.g., figure 6.3. We will use our algorithm to generate a low-discrepancy curve for this surface. We use the diffeomorphism  $u \mapsto u$  and  $v \mapsto g(v)$ . Given the  $\mathfrak{R}^3$  embedding of a torus ( $b < a$ ),

$$x(u, v) = ((a + b \cos(2\pi g(v))) \cos(2\pi u), (a + b \cos(2\pi g(v))) \sin(2\pi u), b \sin(2\pi g(v))) \quad (6.56)$$

$$ds^2 = 4\pi^2(a + b \cos(2\pi g(v)))^2 du^2 + 4\pi^2 b^2 g'^2(v) dv^2 \quad (6.57)$$

$$dA^2 = 16\pi^4 b^2 (a + b \cos(2\pi g(v)))^2 g'^2(v) du^2 dv^2 \quad (6.58)$$

The function  $g$  maps  $[0, 1]$  onto  $[0, 1]$  and is sufficiently smooth. Constant ratio of  $\frac{ds}{du}$  and  $\frac{dA}{du^2}$  in  $\alpha$ -direction can be achieved if the following equation holds true ( $c$  is a constant,  $\alpha_1 > 0$ ):

$$\frac{(a + b \cos(2\pi g(v)))^2 \alpha_1^2 + b^2 g'^2(v) \alpha_2^2}{4\pi^2 b^2 g'^2(v) (a + b \cos(2\pi g(v)))^2 \alpha_1^2 \alpha_2^2} = c \Rightarrow g'(v) = \frac{(a + b \cos(2\pi g(v))) \alpha_1}{\sqrt{4c\pi^2 b^2 (a + b \cos(2\pi g(v)))^2 \alpha_1^2 \alpha_2^2 - b^2 \alpha_2^2}} \quad (6.59)$$

The boundary conditions are  $g(0) = 0$  and  $g(1) = 1$ . A solution of equation 6.59

guarantees  $g'(0) = g'(1)$ . As before, this equation also admits a closed form solution. However, the resulting expressions tend to be somewhat involved. For the purposes of this example, the parameter  $c$  is chosen numerically with the aid of a shooting method. Figure 6.14 depicts part of the resulting low-discrepancy curve lying on the surface of a torus. Because of  $g'(0) = g'(1)$ , the curve is smooth.

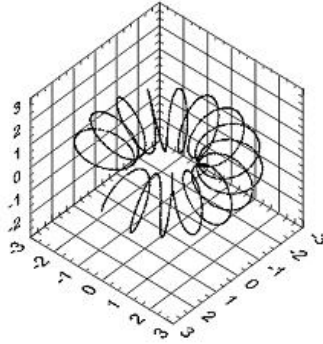


Figure 6.14: Low-discrepancy curve filling the surface of a torus.

### Covering the surface of a sphere

A more sophisticated example is a part of a sphere given as an abstract surface by  $ds^2 = 4\pi^2 \sin^2(\pi g(v)) du^2 + \pi^2 dv^2$  where  $(u, v)$  is in  $[0, 1] \times [v_0, v_1]$  with  $v_0 < v_1$  in  $(0, 1)$ . A Euclidean embedding of this surface is given by,

$$x(u, v) = (\sin(2\pi u) \sin(\pi g(v)), \cos(2\pi u) \sin(\pi g(v)), \cos(\pi g(v))) \quad (6.60)$$

The function  $g(v)$  is smooth and maps  $[v_0, v_1]$  onto  $[v_0, v_1]$ . According to Definition 1 and equation 6.45, we have,

$$g'(v) = \frac{2 \sin(\pi g(v)) \alpha_1}{\sqrt{4c^2 \pi^2 \sin^2(\pi g(v)) \alpha_1^2 \alpha_2^2 - \alpha_2^2}} \quad (6.61)$$

The boundary conditions are  $g(v_0) = v_0$  and  $g(v_1) = v_1$ . Figure 6.15 depicts the resulting low-discrepancy curve.

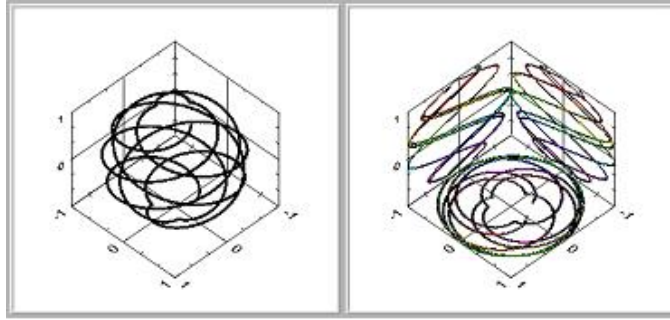


Figure 6.15: A low-discrepancy curve on the surface of a sphere. The figure on the right shows the three 2-dimensional projections along the axis planes. Contrast this against figure 6.7 which achieves much poorer coverage despite using significantly more curvelength. Moreover, when both position and orientation are taken into account, it becomes clear that naive schemes, such as in figure 6.7, are simply not sufficient.

#### 6.4.4 Discussion: On the definition of fair diffeomorphisms

Broadly speaking, there are three major steps in the procedure outlined in the previous subsections:

1. Define a criterion according to which the curve optimally covers space. This is a *fairness* requirement.
2. Define a diffeomorphism that carries the unit square to an abstract surface.
3. Based on the previous two steps, solve a differential equation whose solution yields the diffeomorphism that respects our stated criterion for fairness.

We defined fairness according to the requirement that any two arbitrary segments of the curve, if they have the same length, must cover the same amounts of area or volume. This is an intuitively obvious definition and it is easy to visualize this requirement for a curve in the plane. However, in higher dimensions, this requirement alone may not be sufficient in the sense that the resulting family of solutions may be large. One may wish to utilize additional definitions of fairness.

For instance, with a 3-manifold, in addition to the basic fairness requirement in equation 6.49, we would require that the volume is not affected,  $\sqrt{\det(g)}dx_1^2dx_2^2dx_3^2 = \text{const.}$  and the curve is fair in the sense of being distributed in accordance with the surface



element, i.e.,  $\frac{dl}{L} = \frac{da}{A(S)}$  where  $da = \sqrt{(g_{11}g_{22} - g_{12}^2)dx_1^2dx_2^2} + \sqrt{(g_{11}g_{33} - g_{13}^2)dx_1^2dx_3^2} + \sqrt{(g_{22}g_{33} - g_{23}^2)dx_2^2dx_3^2}$  is the area element.

In general, this sort of engineering of fairness requirements would be difficult to factor into traditional techniques for generating low discrepancy sequences. On the other hand, due to the geometric nature of our problem formulation, such requirements fit very naturally into our algorithmic framework.

## Chapter 7

# Conclusions and Prospects for the Future

### 7.1 Key Ideas

At the highest level, this dissertation represents an attempt to merge the benefits of model-based control design, which has been used with spectacular success in many structured engineering applications, with those of machine learning, emphasizing data-driven synthesis and adaptation of strategies. As robots leave the factory floors and move into unstructured environments, the needs of reliable and efficient operation in the face of imprecise information makes such a merger of ideas appealing. Correspondingly, the complementary strengths and weaknesses of these two disciplines makes such a merger necessary, if not inevitable.

In our approach to this high level problem, three major themes have emerged to become useful design principles:

1. *Functional Biomimesis*: Even as engineers struggle to build efficient, reliable and yet flexible machines, natural counterparts of exactly such machines are all around us. Even the humble invertebrates that crawl around in the undergrowth represent marvels of natural engineering that we are yet to replicate. Utilizing principles from the study of such biological solutions is a promising and fertile direction for robotics research. While the basic notion of abstraction has a long tradition and defines the essence of the scientific method, specific approaches to defining useful abstractions such as the template-anchor decomposition [6, 4, 100] are instructive to the planning

and control researcher. Our intent in performing such abstractions is not only to reduce a system to its constituent elements, the standard reductionist paradigm, but instead to properly delineate functional components so that it becomes possible to infer concise relationships in an integrative scientific setting.

2. *Qualitative Models*: Separating the concerns of qualitative correctness from those of quantitative optimality is an important step towards managing complexity. The use of qualitative models that support structurally stable phase space orbits makes it possible to reason about global behaviors without committing to the quantitative details of a changing task and environment.
3. *Using Qualitative Models to Regularize Learning*: The need for data-driven adaptation has been understood since the inception of planning and control as a domain of inquiry. However, specifying “prior knowledge” in a learning algorithm, without unduly constraining the space of possible solutions, has been a difficult task - particularly when dealing with high-dimensional nonlinear systems. We find that when the problem is suitably formulated - in terms of a template model that utilizes the *natural* dynamics, the task takes on the more organized structure of a manifold which can be learned in a data-driven manner. Moreover, this paradigm is a fairly general one and the manifold sampling and approximation algorithms developed in this dissertation can be gainfully applied in many other domains as well.

## 7.2 Future Work and Open Questions

### 7.2.1 Scaling up to physical bipeds

The eventual goal of the work described in chapters 4 and 5 is to endow humanoid robots with the ability to efficiently and robustly execute spatiotemporally extended tasks. This needs a few components in addition to the ones addressed in this dissertation. For instance, one would need to also implement high-level deliberative planning and low-level feedback control for disturbance rejection to go with the trajectory generation algorithms. Although these problems have been extensively addressed in the literature, some care and effort is required to implement them on physical systems.

There is also an algorithmic challenge. The algorithms and experiments described in chapter 5 were intended to explore and demonstrate a variety of conceptual issues. As

such, each algorithm can be improved through the incorporation of ideas from current research in computational statistics, high-dimensional computational geometry, etc. The main issue is that of dealing with large sets of observations which is necessary for at least two reasons. Firstly, a large well-distributed data set is required to capture all the subtleties in the dynamics of a physical robot. Secondly, the ideal goal is to have a robot that learns on-line, continuously adding to the set of observations.

The single improvement that will best address this issue is the incorporation of multi-resolution algorithms. Many operations in the algorithms developed here were implemented in the most direct way possible, but each of them can be improved in terms of memory and run-time requirements. For operations such as computation of shortest paths, one could organize the data in such a way that the problem is solved by iterative improvement - by first computing a coarse path which is then refined by appropriate relaxation procedures. Some care is required when designing such hierarchical procedures to ensure that they converges to similar solutions paths as the naive procedure using all the data. Similarly, nonparametric statistical approximations can be done in a “multipole” fashion such that the effect of far away points is clumped together and the effective dimensionality of the computations is kept manageable. Yet another improvement that is likely to be significant, especially for computing actions, is to use more sophisticated regression techniques than the one in algorithm 6.

Unfortunately, we do not own or have easy access to a humanoid robot on which we can try our ideas, but this area has seen a burst of activity in recent years and we hope to be able to collaborate with owners of such robots to carry this work forward.

## **7.2.2 Understanding the proper role of template models**

### **What are the properties of a good template model?**

As described in chapters 4 and 5, one of the principal benefits of the use of template models in our work is that it provides guidance on phase space topology. In general, typical sets of observations deduced from high-dimensional nonlinear systems may not be sufficient to deduce, and elucidate to the level required by subsequent algorithmic components, the existence of structure such as the stratified manifold depicted in figure 4.8. However, knowing that the template model possesses these characteristics such as intersections and jumps makes it possible to disambiguate data coming from adjoining manifolds, and also to discard spurious data. This is a powerful addition to the standard machine learning approach.

In general, given a collection of simple models, how can we know which ones will yield similar benefits? In an unknown domain, how are we to rank the power of these abstractions to shape learning? These are important questions that will need to be answered as we apply this approach to other novel domains.

### **Can template models be learned autonomously?**

A very related question to the previous one is whether we can induce suitable abstractions directly from data. This is the current focus of a very vibrant research community interested in nonlinear dimensionality reduction. However, most current approaches are focused primarily on specific metric structure such as inter-point distances of a cloud. There is a need for algorithms that can be targeted differently, e.g., to preserve topological structure even at the cost of some significant local distortions. There have been some attempts in this direction, e.g., [133, 51], however these works were not concerned with the end goal of planning and control as we are. We believe that the answer to the question of whether templates can be learned autonomously is perhaps closely tied to that of information requirements for various tasks.

### **7.2.3 Towards scientific discovery**

Despite the long history of artificial intelligence research, it is only recently that nontrivial dynamical systems entered the purview of AI researchers. In some part, this has been stimulated by increasing interest in embodied cognition and intelligent robotics. However, although the development of computational tools for discovering dynamics may often be motivated by robotics, their domain of applicability is much broader. One promising area that can benefit from such tools is that of scientific discovery.

To understand this connection, consider the simple description of our approach to solving the dynamic bipedal walking problem, depicted in figure 7.1. If in this data flow diagram, we were to replace template model with the hypotheses and structural assumptions being made about a set of experimental data, which takes the place of the robot, then the output of this process is a predictive model of control and decision making as it occurs in the natural system being studied.

Several leading scientists and engineers are currently advocating such an approach to scientific discovery [134, 135]. Particularly in the biological sciences, where spatiotemporally extended decision processes play an important role in organizing the operation of

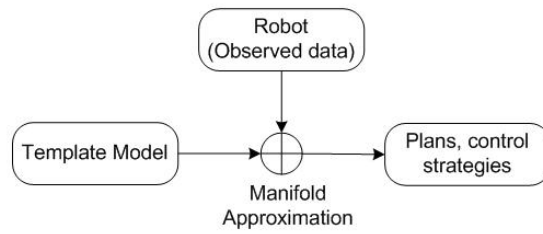


Figure 7.1: High-level conceptual illustration of the approach taken to solve the dynamic bipedal walking problem.

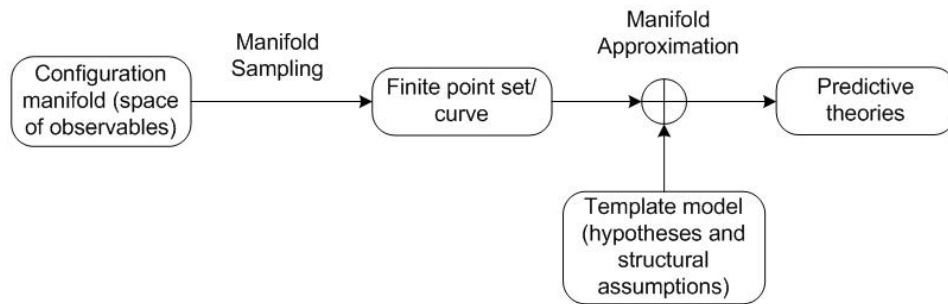


Figure 7.2: A possible architecture for organizing the tools developed in this dissertation to address the needs of scientific discovery - to autonomously discover dynamical structures.

the system, tools such as those developed in this dissertation are likely to be useful. In fact, in this setting, one may consider a more refined process outlined in figure 7.2. This is an important direction for future inquiry - one we plan to actively pursue.

# Bibliography

- [1] H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki and S. Thrun, *Principles of Robot Motion*, MIT Press, 2005.
- [2] M. Raibert, *Legged Robots that Balance*, The MIT Press, Cambridge, MA, 1986.
- [3] D.M. Wolpert, Z. Ghahramani and J.R. Flanagan, Perspectives and problems in motor learning, *Trends in Cognitive Sciences*, Vol. 5(11), pp. 487-494, Nov. 2001.
- [4] M.H. Dickinson, C.T. Farley, R.J. Full, M.A.R. Koehl, R. Kram and S. Lehman, How animals move: An integrative view, *Science*, Vol. 288, pp. 100-106, April 2000.
- [5] R.M. Alexander, *Principles of Animal Locomotion*, Princeton University Press, 2002.
- [6] R.J. Full and D.E. Koditschek, Templates and anchors: neuromechanical hypotheses of legged locomotion on land, *Journal of Experimental Biology*, Vol. 202(23), pp. 3325-3332, 1999.
- [7] R. Murray-Smith and T.A. Johansen, Eds., *Multiple Model Approaches to Nonlinear Modeling and Control*, Taylor & Francis, London, UK, 1997.
- [8] S.H. Collins, A. Ruina, R. Tedrake and M. Wisse, Efficient bipedal robots based on passive-dynamic walkers, *Science*, Vol. 307, pp. 1082-1085, Feb. 2005.
- [9] T. McGeer, Dynamics and control of bipedal locomotion, *Journal of Theoretical Biology*, Vol. 163, pp. 277-314, 1993.
- [10] S. Mochon and T.A. McMahon, Ballistic walking: An improved model, *Mathematical Biosciences*, Vol. 52, pp. 241-260, 1980.

- [11] G.A. Cavagna, P.A. Willems, M.A. Legramandi and N.C. Heglund, Pendular energy transduction within the step in human walking, *Journal of Experimental Biology*, Vol. 205, pp. 3413-3422, 2002.
- [12] M. Wisse, Three additions to passive dynamic walking: actuation, an upper body, and 3D stability, In *Proc. IEEE Intl. Conf. Humanoid Robots*, Los Angeles, CA, 2004.
- [13] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, Springer-Verlag, 1989.
- [14] S. P. Novikov and I. A. Taimanov, *Modern Geometric Structures and Fields*, AMS Graduate Studies in Mathematics, vol. 71, 2006.
- [15] D. G. Kendall, D. Barden, T. K. Carne and H. Le, *Shape and Shape Theory*, 1999.
- [16] S-I. Amari and H. Nagaoka, *Methods of Information Geometry*, AMS, 2001.
- [17] S. M. LaValle, P. Finn, L.E. Kavraki and J.-C. Latombe, A randomized kinematics-based approach to pharmacophore-constrained conformational search and database screening, *Journal of Computational Chemistry*, Vol. 21(9), pp. 731-747, 2000.
- [18] M.Moll, D. Schwarz and L.E. Kavraki, Roadmap methods for protein folding and docking, In *Protein Structure Prediction: Methods and Protocols*, Methods In Molecular Biology, 2nd ed., Humana Press, 2007.
- [19] H. Simon, *The Sciences of the Artificial*, 3rd ed., MIT Press, 1996.
- [20] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- [21] B.J. Kuipers, *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*, Cambridge, MA: MIT Press, 1994.
- [22] E. Sacks, Automatic analysis of one-parameter planar ordinary differential equations by intelligent numeric simulation, *Artificial Intelligence*, Vol. 48(1), pp. 27-56, 1991.
- [23] P. Struss, Model-based and qualitative reasoning: An introduction, *Annals of Mathematics and Artificial Intelligence*, Vol. 19(3-4), pp. 355-381, 1997.
- [24] T. Nishida, Grammatical description of behaviors of ordinary differential equations in two-dimensional phase space, *Artificial Intelligence*, Vol. 91, pp. 3-32, 1997.



- [25] F. Zhao, Extracting and representing qualitative behaviors of complex systems in phase spaces, *Artificial Intelligence*, Vol. 69(1-2), pp. 51-92, 1994.
- [26] B. Shults and B.J. Kuipers, Proving properties of continuous systems: qualitative simulation and temporal logic, *Artificial Intelligence*, Vol. 92, pp. 91-129, 1997.
- [27] K.M-K. Yip, *KAM: A System for Intelligently Guiding Numerical Experimentation by Computer*, MIT Press, 1991.
- [28] E. Bradley, Autonomous exploration and control of chaotic systems, *Cybernetics and Systems*, Vol. 26, pp. 299-319, 1995.
- [29] E. Bradley, M. Easley and R. Stolle, Reasoning about nonlinear system identification, *Artificial Intelligence*, Vol. 133, pp. 139-188, 2001.
- [30] F. Zhao, S.C. Loh and J.A. May, Phase-space nonlinear control toolbox: The maglev experience, in P.J. Antsaklis, W. Kohn, M. Lemmon, A. Nerode and S. Sastry (Eds.), *Hybrid Systems V*, LNCS 1567, pp. 429-444, Springer, 1999.
- [31] P. Holmes, Poincaré, celestial mechanics, dynamical systems theory and “chaos”, *Physics Reports*, Vol. 193, pp. 137, 1990.
- [32] A.G. Hofmann, *Robust Execution of Bipedal Walking Tasks from Biomechanical Principles*, Ph.D. Thesis, Massachusetts Institute of Technology, 2005.
- [33] J. B. Tenenbaum, V. de Silva and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science*, Vol. 290(5500), pp. 2319-2323, 2000.
- [34] Y. Bengio, M. Monperrus, and H. Larochelle, Nonlocal estimation of manifold structure, *Neural Computation*, Vol. 18, pp. 2509-2528, 2006.
- [35] P. Indyk, *High-dimensional Computational Geometry*, Ph.D. Thesis, Stanford University, 2000.
- [36] M. Bern et al., Emerging challenges in computational topology, Report on an NSF-funded Workshop, arXiv:cs/9909001v1, 1999.
- [37] V. Robins, J. Meiss and E. Bradley, Computing connectedness: disconnectedness and discreteness, *Physica D*, Vol. 139, pp. 276-300, 2000.

- [38] I.M. Gelfand and S.V. Fomin, *Calculus of Variations*, Dover, 2000.
- [39] K. D. Mombaur, *Stability Optimization of Open-loop Controlled Walking Robots*, Ph.D. Thesis, Universität Heidelberg, 2001.
- [40] M.B. Milam, *Real-time Optimal Trajectory Generation for Constrained Dynamical Systems*, Ph.D. Thesis, California Institute of Technology, 2003.
- [41] S.M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.
- [42] E. Frazzoli. *Robust Hybrid Control for Autonomous Vehicle Motion Planning*. Ph.D. Thesis, Massachusetts Institute of Technology, 2001.
- [43] C. Dever, B. Mettler, E. Feron, J. Popovic and M. McConley, Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes, *Journal of Guidance, Control and Dynamics*, Vol.29(2), pp. 289-302, 2006.
- [44] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.
- [45] M.T. Rosenstein and A.G. Barto, Reinforcement learning with supervision by a stable controller, In *Proc. American Control Conference*, pp. 4517-4522, 2004.
- [46] T. J. Perkins and A. G. Barto, Lyapunov design for safe reinforcement learning, *Journal of Machine Learning Research*, Vol. 3, pp. 803-823, 2002.
- [47] A.Y. Ng and H.J. Kim, Stable adaptive control with online learning, In *Proc. Neural Information Processing Systems*, Vol. 17, 2005.
- [48] R.L. Tedrake. *Applied Optimal Control for Dynamically Stable Legged Locomotion*. Ph.D. Thesis, Massachusetts Institute of Technology, 2004.
- [49] D.H. Wolpert and W.G. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation*, Vol. 1, pp. 67, 1997.
- [50] S. Schaal, Is imitation learning the route to humanoid robots?, *Trends in Cognitive Sciences*, Vol. 3, pp. 233-242, 1999.
- [51] D. Suč and I. Bratko, Skill modeling through symbolic reconstruction of operator's trajectories. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, Vol. 30(6), pp. 617-624, 2000.

- [52] C. Atkeson, A.W. Moore and S. Schaal, Locally weighted learning for control, *Artificial Intelligence Review*, Vol. 11, pp. 75-113, 1997.
- [53] R.A. Peters and O.C. Jenkins, Uncovering manifold structures in robonaut's sensory-data state space. In *Proc. IEEE Intl. Conf. Humanoid Robotics*, 2005.
- [54] O.C. Jenkins and M.J. Mataric, A spatio-temporal extension to Isomap nonlinear dimension reduction. In *Proc. Intl. Conf. Machine Learning*, pp. 441-448, 2004.
- [55] K. Tatani and Y. Nakamura, Reductive mapping for sequential patterns of humanoid body motion, In *Proc. Intl. Symp. Adaptive Motion of Animals and Machines*, 2003.
- [56] R. Chalodhorn, D.B. Grimes, G. Maganis, R.P.N. Rao, M. Asada, Learning humanoid motion dynamics through sensory-motor mapping in reduced dimensional spaces, In *Proc. IEEE Intl. Conf. Robotics and Automation*, 2006.
- [57] R. F. Stengel, *Optimal Control and Estimation*. Dover Publications, Mineola, NY. 1994.
- [58] G. E. Dullerud and F. Paganini, *A Course in Robust Control Theory: A Convex Approach*, Texts in Applied Mathematics, Vol 36, Springer - New York, 2000.
- [59] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed., Addison-Wesley, 1994.
- [60] J.J.E. Slotine, W. Li, *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs NJ, 1991.
- [61] D. Bernstein, What makes some control problems hard?, *IEEE Control Systems Magazine*, 2002.
- [62] V. Blondel and J. Tsitsiklis, Complexity of stability and controllability of elementary hybrid systems, *Automatica*, Vol. 35(3), pp. 479-489, 1999.
- [63] R.R. Burridge, A.A. Rizzi and D.E. Koditschek, Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research*, Vol. 18, pp. 534-555, 1999.
- [64] P. Faloutsos, M. van de Panne and D. Terzopoulos, Composable controllers for physics-based character animation. *Proc. SIGGRAPH*, pp. 251-260, 2002.

- [65] A.D. Kuo, J.M. Donelan and A. Ruina, Energetic consequences of walking like an inverted pendulum: step-to-step transitions, *Exercise and Sport Sciences Review*, Vol. 33(2), pp.88-97, 2005.
- [66] A.D. Kuo, Energetics of actively powered locomotion using the simplest walking model, *Journal of Biomechanical Engineering*, Vol. 124, pp. 113-120, 2002.
- [67] E.R. Kandel, J.H. Schwartz, and T.M. Jessell (Eds.), *Principles of Neural Science*, 3rd edition. Norwalk, CT: Appleton & Lange, 1991.
- [68] M.R. Dimitrijevic, Y. Gerasimenko and M.M. Pinter, Evidence for a spinal central pattern generator in humans, *Annals of the New York Academy of Sciences*, Vol. 860, pp. 360-376, 1998.
- [69] P. Holmes, R.J. Full, D. Koditschek and J. Guckenheimer, Dynamics of legged locomotion: Models, analysis, and challenges, *SIAM Review*, Vol. 48(2), pp. 207-304, 2006.
- [70] A. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model, *Science*, Vol. 315(5817), pp. 1416-1420, 2007.
- [71] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, Hybrid zero dynamics of planar biped walkers, *IEEE Transactions on Automatic Control*, Vol. 48(1), pp. 42-56, 2003.
- [72] E. R. Westervelt, G. Buche, and J. W. Grizzle, Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds, *International Journal of Robotics Research*, Vol. 24(6), pp. 559-582, 2004.
- [73] J. Pratt, C-M. Chew, A. Torres, P. Dilworth, and G. Pratt, Virtual model control: An intuitive approach for bipedal locomotion, *International Journal of Robotics Research*, Vol. 20(2), pp.129-143, 2001.
- [74] C. E. Bauby and A. D. Kuo. Active control of lateral balance in human walking. *Journal of Biomechanics*, Vol. 33, pp. 1433-1440, 2000.
- [75] B. Krauskopf and H.M. Osinga, Computing geodesic level sets on global (un)stable manifolds of vector fields, *SIAM Journal on Applied Dynamical Systems*, Vol. 2(4), pp. 546-560, 2003.

- [76] K. J. Åström and K. Furuta, Swinging up a pendulum by energy control, *Automatica*, Vol. 36, pp. 287-295, 2000.
- [77] J. Zhao and M.W. Spong, Hybrid control for global stabilization of the cart-pendulum system, *Automatica*, Vol. 37, pp. 1941-1951, 2001.
- [78] A. Shiriaev, A. Pogromsky, H. Ludvigsen and O. Egeland, On global properties of passivity-based control of an inverted pendulum, *International Journal of Robust and Nonlinear Control*, Vol. 10(4), pp. 283-300, 2000.
- [79] Q. Wei, W.P. Dayawansa and W.S. Levine, Nonlinear controller for an inverted pendulum having restricted travel, *Automatica*, Vol. 31(6), pp. 841 - 850, 1995.
- [80] Z. Lin, A. Saberi, M. Gutmann and YA Shamash, Linear controller for an inverted pendulum having restricted travel: a high-and-low gain approach. *Automatica*, Vol. 32, pp. 933-937, 1996.
- [81] F.H. Clarke, Yu.S. Ledyaev, R.J. Stern and P.R. Wolenski, Qualitative properties of trajectories of control systems: A survey, *Journal of Dynamical and Control Systems*, Vol. 1(1), pp. 1-48, 1995.
- [82] B. J. Kuipers and K. J. Åström, The composition and validation of heterogeneous control laws, *Automatica*, Vol. 30(2), pp. 233-249, 1994.
- [83] C.J. Tomlin, J. Lygeros and S.S. Sastry, A game theoretic approach to controller design for hybrid systems, *Proceedings of the IEEE*, Vol. 88(7), pp. 949-970, 2000.
- [84] R. Ghosh and C. J. Tomlin, Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modeling: Delta-Notch protein signaling, *IEE Transactions on Systems Biology*, Vol. 1(1), pp. 170-183, 2004.
- [85] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Springer-Verlag, Berlin, 1983.
- [86] D.W. Jordan and P. Smith, *Nonlinear Ordinary Differential Equations. An Introduction to Dynamical Systems*, 3rd ed., Oxford University Press, Oxford, 1999.
- [87] K.M. Passino and S. Yurkovich, *Fuzzy Control*, Addison Wesley, 1997.
- [88] S. Haykin, *Neural Networks*, 2nd ed., Prentice-Hall, NJ, 1999.

- [89] F. Bullo and A.D. Lewis, *Geometric Control of Mechanical Systems*, No. 49, Texts in Applied Mathematics, Springer-Verlag, 2004.
- [90] T. Parisini, M. Sanguineti and R. Zoppoli, Nonlinear stabilization by receding-horizon neural regulators, *International Journal of Control*, Vol. 70(3), pp. 341-362, 1998.
- [91] A. Draeger, S. Engell and H. Ranke, Model predictive control using neural networks, *IEEE Control Systems Magazine*, Vol. 15(5), pp. 61-66, 1995.
- [92] R. M. Ghigliazza, R. Altendorfer, P. Holmes and D. Koditschek, A simply stabilized running model, *SIAM Review*, Vol. 47(3), pp. 519-549, 2005.
- [93] J.K. Hodgins and M. Raibert, Biped gymnastics, *International Journal of Robotics Research*, Vol. 9(2), pp. 115-132, 1990.
- [94] M. Garcia, A. Ruina, A. Chatterjee and M. Coleman, The simplest walking model: stability, complexity, and scaling, *ASME Journal of Biomechanical Engineering*, Vol. 120(2), pp. 281-288, Apr 1998.
- [95] M. Vukobratovic, B. Borovac, D. Surla and D. Stokic, *Biped Locomotion: Dynamics, Stability, Control and Application*, Scientific Fundamentals of Robotics, Vol. 7, Springer-Verlag, 1989.
- [96] P. Sardain and G. Bessonnet, Forces acting on a biped robot, center of pressure - zero moment point. *IEEE Transactions on Systems, Man, and Cybernetics Part A*, Vol. 34(5), pp. 630-637, 2004.
- [97] J.J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, Motion planning for humanoid robots under obstacle and dynamic balance constraints, In *Proc. IEEE Intl Conf. Robotics and Automation*, pp. 692-698, 2001.
- [98] C. Canudas-de-Wit, On the concept of virtual constraints as a tool for walking robot control and balancing, *Annual Reviews in Control*, Vol. 28, pp. 157-166, 2004.
- [99] K. J. Åström, Modeling of complex systems, In *Modeling, Control and Optimization of Complex Systems - Symposium in Honor of Professor Yu-Chi Ho*, Kluwer, 2002.

- [100] Y. Gutfreund, T. Flash, Y. Yarom, G. Fiorito, I. Segev and B. Hochner, Organization of octopus arm movements, *Journal of Neuroscience*, Vol. 16(22), pp. 7297-7307, 1996.
- [101] Y.P. Ivanenko, N. Dominici, G. Cappellini, B. Dan, G. Cheron and F. Lacquaniti, Development of pendulum mechanism and kinematic coordination from the first unsupported steps in toddlers, *Journal of Experimental Biology*, Vol. 207, pp. 3797-3810, 2004.
- [102] M. Coleman and A. Ruina, An uncontrolled toy that can walk but cannot stand still, *Physical Review Letters*, Vol. 80(16), pp. 3658-3661, 1998.
- [103] A.D. Kuo, Harvesting energy by improving the economy of human walking, *Science*, Vol. 309, pp. 1686-1687, 2005.
- [104] M. Srinivasan and A. Ruina, Computer optimization of a minimal biped model discovers walking and running, *Nature*, Vol. 439, pp. 72-75, 2006.
- [105] B. Goodwine and J. Burdick. Trajectory generation for kinematic legged robots. In *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 2689-2696, 1997.
- [106] J.B. Dingwell and J.P. Cusumano, Nonlinear time series analysis of normal and pathological human walking, *Chaos*, Vol. 10(4), pp. 848-863, 2000.
- [107] G.J. Sussman and J. Wisdom, *Structure and Interpretation of Classical Mechanics*, The MIT Press, 2001.
- [108] M. Balasubramanian, E. L. Schwartz, J. B. Tenenbaum, V. de Silva, and J. C. Langford, The Isomap algorithm and topological stability, *Science*, vol. 295, pp. 7a, January 2002.
- [109] R. E. Poppele, G. Bosco and A. M. Rankin, Independent representations of limb axis length and orientation in spinocerebellar response components, *Journal of Neurophysiology*, Vol. 87(1), pp. 409-422, 2002.
- [110] R. Featherstone, *Robot Dynamics Algorithms*, Kluwer Academic Publishers, 1987.
- [111] N. Michael, C. Belta and V. Kumar, Controlling three dimensional swarms of robots, In *Proc. Intl. Conf. Robotics and Automation*, 2006.

- [112] E.M. Purcell, Life at low Reynolds number, *American Journal of Physics*, 45(1):3, 1977.
- [113] A. Shapere and F. Wilczek, Self-propulsion at low Reynolds number. *Physical Review Letters*, 58(20):2051, 1987.
- [114] G. Kosa, M. Shoham and M. Zaaroor, Propulsion of a swimming micro medical robot, In *Proc. Intl. Conf. Robotics and Automation*, 2005.
- [115] I-Ming Chen, Hsi-Shang Li and A. Cathala, Design and simulation of Amoebot - a metamorphic underwater vehicle, In *Proc. Intl. Conf. Robotics and Automation*, 1999.
- [116] A. Hayashi and B.J. Kuipers, A continuous approach to robot motion planning with many degrees of freedom, In *Proc. IEEE Intl. Conf. Intelligent Robots and Systems*, 1992.
- [117] L. Wenzel, R. Rajagopal and D. Nair, Induced well-distributed sets in Riemannian spaces, *ACM Transactions on Mathematical Software*, Vol. 29(1), pp. 8294, 2003.
- [118] J. Matoušek, *Geometric Discrepancy: An Illustrated Guide*, Springer, 1999.
- [119] S.M. LaValle, M.S. Branicky, and S.R. Lindemann, On the relationship between classical grid search and probabilistic roadmaps. *International Journal of Robotics Research*, Vol. 23(7-8), pp. 673-692, 2004.
- [120] S.R. Lindemann, and S.M. LaValle, Incremental low-discrepancy lattice methods for motion planning, In *Proc. IEEE International Conference on Robotics and Automation*, pp. 2920-2927, 2003.
- [121] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, CBMS-NSF Regional Conference Series in Applied Math., No.63, SIAM, 1992.
- [122] L. Kocis and W.J. Whiten, Computational investigations of low-discrepancy sequences, *ACM Transactions on Mathematical Software*, Vol. 23(2), pp. 266-294, 1997.
- [123] J.G. van der Corput, Vereilungsfunktionen I,II, *Nederl. Akad. Wetensch. Proc. Ser. B*, Vol. 38, pp. 813-821,1058-1066, 1935.



- [124] J.H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numerical Mathematics*, Vol. 2, pp. 84-90, 1960.
- [125] I.M. Sobol', The distribution of points in a cube and the approximate evaluation of integrals, *Zh. Vychisl. Mat. i Mat. Fiz. (USSR Computational Mathematics and Mathematical Physics)*, Vol. 7, pp. 784-802, 1967.
- [126] R.D. Richtmyer, The evaluation of definite integrals, and quasi-Monte Carlo method based on the properties of algebraic numbers. Report LA-1342, Los Alamos Scientific Laboratory, 1951.
- [127] R.D. Richtmyer, A non-random sampling method, based on congruences for monte carlo problems, Report NYO-8674, Institute of Mathematical Sciences, New York University, 1958.
- [128] J.M. Hammersly and D.C. Handscomb, *Monte Carlo Methods*, London: Methuen and Company, 1964.
- [129] W. Morokoff and R.E. Caflisch, Quasi-random sequences and their discrepancies, *SIAM Journal on Scientific and Statistical Computing*, Vol. 15, pp. 1251-1279, 1994.
- [130] F. James, J. Hoogland and R. Kleiss, Multidimensional sampling for simulation and integration: measures, discrepancies, and quasi-random numbers, *Computer Physics Communications*, Vol. 99, pp. 180-220, 1997.
- [131] G.H. Hardy and E.M. Wright, *An Introduction to the Theory of Numbers*, 5th ed., Oxford University Press, 1980.
- [132] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, 1992.
- [133] S. V. Chakravarthy and J. Ghosh, Function emulation using radial basis function networks, *Neural Networks*, Vol. 10(3), pp. 459-478, 1997.
- [134] C.J. Tomlin and J.D. Axelrod, Understanding biology by reverse engineering the control, *Proceedings of the National Academy of Sciences*, Vol. 102(12), pp. 4219-4220, 2005.
- [135] M.E. Csete and J.C. Doyle, Reverse engineering of biological complexity, *Science*, Vol. 295(5560), pp. 1664-1669, 2002.

# Vita

Subramanian Ramamoorthy was born in Tiruchirapalli, Tamilnadu, India on 31 October 1975, the son of Lakshmi and Ramamurthy Subramanian. He received the Bachelor of Engineering degree from Bangalore University, India in 1996. During 1995-96, he worked on a senior thesis at the National Aerospace Laboratories, Bangalore, India. He spent October 1996 - July 1997 as a Project Associate at the Inter-University Consortium for Department of Atomic Energy Facilities, Indore, India.

In August 1997, he moved to the United States to pursue graduate studies. He received the Master of Engineering degree from the University of Virginia in 1999. Since October 1999, he has been employed with National Instruments Corp., Austin, TX, where he has been a member of multiple research and development groups including motion control, computer vision, control design and dynamic simulation, making significant contributions to five commercial software products.

He began doctoral studies at The University of Texas at Austin in January 2001. His doctoral research work is conducted within the Intelligent Robotics Laboratory, under the supervision of Prof. Benjamin J. Kuipers.

Permanent Address: 3356 Lake Austin Blvd. #C  
Austin, TX 78703  
USA

This dissertation was typeset with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub><sup>1</sup> by the author.

---

<sup>1</sup>L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> is an extension of L<sup>A</sup>T<sub>E</sub>X. L<sup>A</sup>T<sub>E</sub>X is a collection of macros for T<sub>E</sub>X. T<sub>E</sub>X is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.