

University of Pennsylvania ScholarlyCommons

Publicly Accessible Penn Dissertations

2018

Autonomous Behaviors With A Legged Robot

Berkay Deniz Ilhan University of Pennsylvania, berkaydeniz@gmail.com

Follow this and additional works at: https://repository.upenn.edu/edissertations Part of the <u>Robotics Commons</u>

Recommended Citation

Ilhan, Berkay Deniz, "Autonomous Behaviors With A Legged Robot" (2018). *Publicly Accessible Penn Dissertations*. 2713. https://repository.upenn.edu/edissertations/2713

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/edissertations/2713 For more information, please contact repository@pobox.upenn.edu.

Autonomous Behaviors With A Legged Robot

Abstract

Over the last ten years, technological advancements in sensory, motor, and computational capabilities have made it a real possibility for a legged robotic platform to traverse a diverse set of terrains and execute a variety of tasks on its own, with little to no outside intervention. However, there are still several technical challenges to be addressed in order to reach complete autonomy, where such a platform operates as an independent entity that communicates and cooperates with other intelligent systems, including humans. A central limitation for reaching this ultimate goal is modeling the world in which the robot is operating, the tasks it needs to execute, the sensors it is equipped with, and its level of mobility, all in a unified setting. This thesis presents a simple approach resulting in control strategies that are backed by a suite of formal correctness guarantees. We showcase the virtues of this approach via implementation of two behaviors on a legged mobile platform, autonomous natural terrain ascent and indoor multi-flight stairwell ascent, where we report on an extensive set of experiments demonstrating their empirical success. Lastly, we explore how to deal with violations to these models, specifically the robot's environment, where we present two possible extensions with potential performance improvements under such conditions.

Degree Type Dissertation

Degree Name Doctor of Philosophy (PhD)

Graduate Group Electrical & Systems Engineering

First Advisor Daniel E. Koditschek

Second Advisor Alejandro Ribeiro

Keywords autonomous robot, hill ascent, lyapunov stability, reactive control, stairwell ascent, unicycle

Subject Categories Robotics

AUTONOMOUS BEHAVIORS WITH A LEGGED ROBOT B. Deniz Ilhan A DISSERTATION

in

Electrical and Systems Engineering Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy 2018

> Daniel E. Koditschek, Professor Electrical and Systems Engineering Supervisor of Dissertation

> Alejandro Ribeiro, Professor Electrical and Systems Engineering Graduate Group Chairperson

> > Dissertation Committee:

Alejandro Ribeiro, Professor Electrical and Systems Engineering University of Pennsylvania Daniel E. Koditschek, Professor Electrical and Systems Engineering University of Pennsylvania

Konstantinos Daniilidis, Professor Computer and Information Science University of Pennsylvania Aaron M. Johnson, Assistant Professor Mechanical Engineering Carnegie Mellon University

AUTONOMOUS BEHAVIORS WITH A LEGGED ROBOT

© COPYRIGHT

2018

Berkay Deniz Ilhan

To my mother, Meryem Kuş, without whom I would not be the independent, questioning, learning person that I am.

To my wife, Kait, who challenges me every day, loves and supports me no matter what, even though, at times, I am a handful.

To my son, Erokay, who I watch every day with amazement.

Acknowledgments

I would like to thank all my teachers and fellow classmates of 2002 from my high school, Çanakkale Science High School, Çanakkale, Turkey. Attending a boarding school, sharing twenty four hours a day, seven days a week with complete strangers who turn into your new big family is a scary, yet, humbling experience. Being surrounded by many smart people who were dying to learn advanced new concepts in math, physics, biology, chemistry—even history—made me learn a lot about myself. Living seven hours away from home prepared me for my later move across the Atlantic.

I was lucky to be admitted into the best university in my country, Boğaziçi University, Istanbul, Turkey. I would like to thank all of my professors in the Electrical and Electronics Engineering department for teaching me not only the state-of-the-art in electronics, signal processing, and control theory but also how to learn and digest completely new concepts and ideas in an effective manner. I am forever grateful to my undergraduate advisor and Intelligent Systems Laboratory (ISL) supervisor, Işıl Bozma, for believing in me, letting me join ISL as a rising sophomore, and helping me grow as a roboticist [56].

I can not thank my advisor, Daniel E. Koditschek, enough for creating the immense learning environment in his research group, Kod*lab, where I was able to play with my favorite "toys" (i.e. expensive robotic platforms, sensors, and other equipment) and got paid for it. I learned a lot from him when it comes to control theory and sponsor relations. Also, I learned a lot from the people he brought together. I would like to thank Goran Lynch, Paul Vernaza, and Aaron Johnson for being great colleagues and great friends who helped me adapt to my new life in Philadelphia, teaching me about American food, craft beer, and sports. I would like to thank Clark Haynes and Shai Revzen for all of their support and mentorship during our time together. I would like to thank Paul Reverdy for our collaboration and for being a great listener when I needed to rant about life. I would like to thank all the other lab members throughout my time in Kod*Lab, from post-docs to undergrads to secretaries, for all the help and friendship they offered. I would also like to thank my past and present committee members, Alejandro Ribeiro, Kostas Daniilidis, Shai Revzen, and Aaron Johnson for all of their support.

I would like to thank my mother's family for being there during my formative years. I was a single child raised by a single parent, but I had the most lively and happy childhood thanks to my uncles, aunts, and cousins. I miss my grandmother every day. She was there when my mother needed her and she took great care of me all those years. My mother helped me develop important characteristics that made me an "anomaly" in many ways, good and bad. She encouraged me and supported me throughout my successes and failures. I am forever indebted to her.

Finally, I would like to thank my wife, Kait, for making it possible to get to this point. I don't know what I would do without her love, support, and occasional "reality checks". She is always there for me and I try my best to be there for her. Both of us going through advanced schooling and having a child at the same time was indeed a crazy idea. However, our son, Erokay, makes us feel every day that it is all worth it.

ABSTRACT

AUTONOMOUS BEHAVIORS WITH A LEGGED ROBOT

B. Deniz Ilhan

Daniel E. Koditschek

Over the last ten years, technological advancements in sensory, motor, and computational capabilities have made it a real possibility for a legged robotic platform to traverse a diverse set of terrains and execute a variety of tasks on its own, with little to no outside intervention. However, there are still several technical challenges to be addressed in order to reach complete autonomy, where such a platform operates as an independent entity that communicates and cooperates with other intelligent systems, including humans. A central limitation for reaching this ultimate goal is modeling the *world* in which the robot is operating, the *tasks* it needs to execute, the *sensors* it is equipped with, and its level of *mobility*, all in a unified setting. This thesis presents a simple approach resulting in control strategies that are backed by a suite of formal correctness guarantees. We showcase the virtues of this approach via implementation of two behaviors on a legged mobile platform, autonomous natural terrain ascent and indoor multi-flight stairwell ascent, where we report on an extensive set of experiments demonstrating their empirical success. Lastly, we explore how to deal with violations to these models, specifically the robot's environment, where we present two possible extensions with potential performance improvements under such conditions.

Contents

Α	Acknowledgments					
A	bstra	ict		vi		
Li	st of	Table	S	xiii		
Li	st of	Figur	es	xvii		
1	Intr	oducti	ion	1		
	1.1	Motiv	ation	. 4		
	1.2	Relati	on to Published Work	. 7		
	1.3	Organ	lization and Contributions	. 8		
	1.4	Review	w of Literature	. 12		
Ι	Tas	sk End	coding for a Legged Robot	15		
2	Mat	tching-	LaSalle Functions and Stability	16		
	2.1	Basic	Definitions	. 17		
	2.2	Potent	tial Functions	. 19		
		2.2.1	Lyapunov Functions	. 19		
		2.2.2	Chetaev Functions	. 20		
		2.2.3	Matching LaSalle (ML) Functions	. 22		
	2.3	Embe	dding a System: A Special Case	. 23		
	2.4	Secon	d Order Embedding	. 26		

3 Ta	sk Ence	oding for a Legged Robot	
3.1	World	l and Task	•
	3.1.1	The World Model	
	3.1.2	Task Model	•
	3.1.3	Sensor Models	
3.2	Robot	Control	•
	3.2.1	Autonomous Point Particle Hill Ascent with Obstacle Avoidance $% \mathcal{A}$.	•
		3.2.1.1 Obstacle Model	•
		3.2.1.2 Combined Control Law	•
	3.2.2	Horizontal Unicycle Models	•
		3.2.2.1 Kinematic Unicycle	•
		3.2.2.2 Dynamic Unicycle	•
II A	Autono	mous Behaviors with a Legged Robot	
II A 4 Au	Autono	mous Behaviors with a Legged Robot ous Hill Ascent	
II A 4 Au 4.1	Autono itonom Imple	mous Behaviors with a Legged Robot ous Hill Ascent mentation	
II A 4 Au 4.1	Autonom itonom Imple 4.1.1	mous Behaviors with a Legged Robot ous Hill Ascent mentation Sensors	
II A 4 Au 4.1	Autonom itonom Imple 4.1.1	mous Behaviors with a Legged Robot ous Hill Ascent mentation	•
II A 4 Au 4.1	Autonom Itonom Imple 4.1.1	mous Behaviors with a Legged Robot ous Hill Ascent mentation	
II A 4 Au 4.1	Autonom Itonom Imple 4.1.1	mous Behaviors with a Legged Robot ous Hill Ascent mentation	• • •
II A 4 Au 4.1	Autonom Itonom Imple 4.1.1	mous Behaviors with a Legged Robot ous Hill Ascent mentation	· · · ·
II A 4 Au 4.1	Autonom Itonom Imple 4.1.1 4.1.2 4.1.3	mous Behaviors with a Legged Robot ous Hill Ascent mentation	· · · · · ·
II A 4 Au 4.1	Autonom Itonom Imple 4.1.1 4.1.2 4.1.3	mous Behaviors with a Legged Robot ous Hill Ascent mentation	· · · · · · ·
II A 4 Au 4.1	Autonom Itonom Imple 4.1.1 4.1.2 4.1.3	mous Behaviors with a Legged Robot ous Hill Ascent mentation	· · · · · · ·
II A 4 Au 4.1	Autonom Itonom Imple 4.1.1 4.1.2 4.1.3	mous Behaviors with a Legged Robot ous Hill Ascent mentation . Sensors . 4.1.1.1 Obstacle Sensor . 4.1.1.2 Hill Gradient Sensor . 4.1.1.3 Hill Incline Sensor . Task Encoder . . 4.1.3.1 Bounded Control Inputs . 4.1.3.2 Field of View . 4.1.3.3 Cyclic Body Pose Variations .	
II A 4 Au 4.1	Autonom Itonom Imple 4.1.1 4.1.2 4.1.3	mous Behaviors with a Legged Robot ous Hill Ascent mentation . Sensors . 4.1.1.1 Obstacle Sensor . 4.1.1.2 Hill Gradient Sensor . 4.1.1.3 Hill Incline Sensor . Task Encoder . . 4.1.3.1 Bounded Control Inputs . 4.1.3.2 Field of View . 4.1.3.3 Cyclic Body Pose Variations . 4.1.3.4 Dynamic Unicycle Input .	

		4.2.1	Experiment Sites	77
		4.2.2	Performance Experiments	78
			4.2.2.1 Procedure	78
			4.2.2.2 Results	80
			4.2.2.3 Common Issues	83
		4.2.3	Model Comparison Experiments	84
			4.2.3.1 Procedure	85
			4.2.3.2 Results	87
5	Aut	onomo	ous Stairwell Ascent	89
	5.1	Robot	and Task	90
		5.1.1	World Model	90
			5.1.1.1 The Stairwell Model	90
		5.1.2	Robot Model	91
		5.1.3	Task Model	91
		5.1.4	Sensor Models	91
			5.1.4.1 Depth Sensor	91
			5.1.4.2 Gap Sensor	92
			5.1.4.3 Pitch Scan Sensor	92
			5.1.4.4 Cliff Sensor	94
			5.1.4.5 Stair Sensor	94
	5.2	Auton	omous Stairwell Ascent	96
		5.2.1	The Stair Climbing Behavior	98
		5.2.2	Landing Exploration Behavior	99
	5.3	Exper	imental Results	100
Π	IV	Vorld	Model Violations 1	.03
6	Dyr	namica	l Trajectory Replanning for Uncertain Environments	L04

	6.1	Motiva	ation	105
	6.2	Backg	round Ideas	106
		6.2.1	A First Order Graph as a Second Order Attractor	107
		6.2.2	Internal Dynamical Reference Generators	107
	6.3	Contro	oller Design	109
	6.4	Applie	cation of the Construction	113
		6.4.1	Reference Generator	114
			6.4.1.1 Point attractor reference system	114
			6.4.1.2 Saturated Hopf oscillator reference system	115
		6.4.2	ISS Replanner	115
		6.4.3	Integral-ISS Tracking Error Dynamics	116
		6.4.4	Disturbance	117
	6.5	Simula	ation Studies	118
		6.5.1	Simulations and Quality Metrics	118
			6.5.1.1 Point Attractor with comb obstacle	119
			6.5.1.2 Hopf reference with two obstacles	120
	6.6	Unicy	cle Extension	121
7	A d	rift-di	ffusion model for robotic obstacle avoidance	125
	7.1	Proble	em statement	126
	7.2	Single	obstacle	128
		7.2.1	Assumptions	129
		7.2.2	Probability of escaping a single obstacle	131
		7.2.3	Escape time	133
		7.2.4	Implications for control	136
	7.3	Exper	imental results	138
		7.3.1	Implementation on RHex	139
		7.3.2	Experimental setup	140
		7.3.3	Results	141

8	Conclusions	143
Bi	bliography	147

List of Tables

3.1	Fixed relations and (unknown) geometric parameters underlying the assumed world model	37
3.2	Nomenclature and (unknown) geometric parameters underlying the task model. In the absence of obstacles, the task would be achieved by simply following	01
	the terrain gradient field (3.25) .	40
3.3	Sensor models	42
3.4	Summary of point particle control definitions, assumptions, and sufficient conditions for successful execution. Additional summaries of corresponding world, task, and sensor models are located in Table 3.1, Table 3.2, and Ta-	
	ble 3.3, respectively.	43
3.5	Summary of definitions, assumptions, and sufficient conditions for kinematic unicycle control, which is developed on top of the point particle control (Ta-	
	ble 3.4), with the difference in available measurements	57
3.6	Summary of additional definitions and assumptions for dynamic unicycle control, which is based upon the kinematic unicycle control (Table 3.5).	65
4.1	Eleven outdoor hill climbing behavior trials including 49 detectable obstacles (D.O.) successfully avoided and 58 non-detectable obstacles (N.O.) successfully mechanically traversed over around half a kilometer of climbing with only 4 faults.	81
4.2	Nine outdoor hill climbing behavior trials including 41 detectable obstacles (D.O.) successfully avoided and 44 non-detectable obstacles (N.O.) successfully mechanically traversed over around 350 meters of climbing with only 4 obstacle interaction based faults. 2 of these occurred due to robot failure over non-detectable obstacles. The other 2 occurred due to world model	
	violations (W.M.V.) where a complex set of obstacles resulted in the robot	00
4.9	control strategy failure.	82
4.3	Comparative specific resistance values for kinematic and dynamic controllers operating at two different speeds in the absence of an obstacle. The kinematic controller exhibits lower cost of transport at walking speed and the dynamic	
	controller exhibits better performance at running speed	87

- 4.4 Comparative specific resistance, SR, and moving average based specific resistance series, SR_w , mean (m) and variance (v) for kinematic and dynamic controllers operating at two different speeds in the presence of an obstacle. The kinematic controller exhibits lower cost of transport at walking speed and the dynamic controller exhibits better performance at running speed. The obstacle avoidance maneuver incurs additional cost. Mean values of SR_w agree with the trends seen in SR values. At running speed, disparity between kinematic and dynamic unicycle controllers in terms of mean and variance values of SR_w is more prevalent.
- 5.1 Ten indoor stairwell climbing behavior trials covering 731 stairs in 67 flights with a total of 12 behavioral problems. World model violations are briefly described. Rise, Run and Landing Size dimensions are given in centimeters (cm). Scans column contains two numbers; Stair Scans and Cliff Scans. Behavior faults are categorized as (S)tair Detection, (C)liff Detection, Stair (T)ransition, and (W)all Collision. Robot faults fall into 4 categories; (N)etwork Communication, (L)eg Failures, (L)I(D)AR Failures, and (I)MU Failures. 101

88

7.1 Experimental results. The noisy control strategy results in avoiding the obstacle much more quickly and with significantly higher probability. 142

List of Figures

1.1 1.2 1.3	The RHex robot on a forested hill	23
3.1	(5.50)	10
3.2	of the relative position of the obstacle center, as given in (3.28) Regions Around the Obstacle. From (3.39), $\rho_{u_i} := \left[1 - \frac{\Gamma_h}{\nu_{u_i}}\right] \rho_s$, and from	35
3.3	(3.40), $\rho_{\nu_i} := \frac{\Omega_h}{\nu_{\psi}} \rho_s$	48 55
4.1	Kinematic unicycle implementation. Measurements from the two physical sensors, IMU and LIDAR, are processed to provide the sensory inputs ex- pected by the task encoder module. In return, this module provides the combined task gradient for the kinematic unicycle control law, which is fed	
4.2	into the robot locomotion module as the velocity input Dynamic unicycle implementation. The diagram follows Figure 4.1, except the kinematic unicycle input and its derivative is utilized by the internal system representing dynamic unicycle extension, instead. This system's state is applied to the robot locomotion module as the velocity input	71 72

4.3 4.4 4.5	Procedure for performance experiments. The steps taken to generate reported results are categorized as <i>on-site</i> and <i>post-processing</i>	78 83
4.6	detectable obstacle and its effects on the steepest ascent direction during trial 1 Procedure for model comparison experiments with no obstacle. The steps	84
4.7	taken to generate reported results are categorized as <i>on-site</i> and <i>post-processing</i> . Procedure for model comparison experiments with a single obstacle. The steps taken to generate reported results are categorized as <i>on-site</i> and <i>post-processing</i>	85 86
5.1 5.2	The pitch wiggle behavior for up and down scans, with inactive legs removed for clarity	93
5.3	in degrees. The top two graphs contain the raw readings and the output of a simple filter	97 98
6.1	Plant evolution for the point attractor reference meeting a comb obstacle with different transient to reference (c_{sr}) coupling gains((red) plant, (black) reference). (a) For a small value of c_{sr} the particle remains blocked by the obstacle, (b) For a moderate c_{sr} value plant escapes with very low costs, (c),(d) For higher values of c_{sr} energy cost grows again with resonance peaks when the replanner induces escape maneuvers whose spatial frequencies cou- ple strongly to the geometric features of the particular obstacle.	119
6.2	Energy consumed over the course of the point attractor reference with comb obstacle depicted in Fig. 1 as a function of the transient to reference coupling gain, c_{sr} (of (6.16)). (a) Magnified view of small values of c_{sr} ; (b) Larger values of c_{sr} showing optimum, an approximately linear increase in cost with increased c_{sr} , and occasional resonance peaks where cost is larger over a	
6.3	narrow range	120

6.4	Plant evolution for the Hopf cycle attractor reference meeting an elaborate obstacle with different transient to reference coupling gains ((red) plant, (black) reference). (a) The classical trajectory tracker (small or zero $c_{\rm sr}$) gets trapped in the cul-de-sac as expected, and unlike the previous case, even though the reference trajectory gets behind the plant at each period, the plant can not leave the trap. (b),(c) At a sufficiently large $c_{\rm sr}$ a qualitative change appears whereby the initial hit excites a successful escape recovery trajectory which returns along the unblocked portion of the cycle to repeat	
6.5	the same pattern, cycle after cycle. \ldots Energy consumed over the course of the Hopf cycle attractor reference with simple obstacle depicted in Fig. 3 as a function of the transient to reference coupling gain. (a) Magnified view of small values of $c_{\rm sr}$; (b) Larger values of $c_{\rm sr}$ showing optimum, an approximately linear increase in cost with increased	122
6.6	Energy consumed over the course of the Hopf cycle attractor reference with elaborate obstacle depicted in Fig. 4 as a function of the transient to reference coupling gain. (a) Magnified view of small values of $c_{\rm sr}$; (b) Larger values of $c_{\rm sr}$ showing optimum, an approximately linear increase in cost with increased	120
6.7	c_{sr} , and occasional resonance peaks where cost is larger over a narrow range. Contributions to total Lyapunov function η_{total} for one cycle of the Hopf system. The tracking error Lyapunov η_{e} (red) comprising potential (cyan) and kinetic terms grows rapidly when the obstacle is hit, causing a growth of the transient ϕ_{s} ($\eta_{e} + \phi_{s}$ in green). The ISS Lyapunov function η_{total} (blue) continues to grow until the transient becomes sufficiently small, and then it too decays exponentially	124 124
7.1	Example obstacles placed in the flow described by the dynamics (7.5). Panel A: an obstacle that is convex with respect to the flow, and will not trap a particle with $D > 0$. Panel B: An obstacle that is concave with respect to the flow and may trap a particle regardless of the value of D .	121
7.2	The geometry of interaction with a concave obstacle. There are three charac- teristic lengths involved: d , $D\sqrt{t_d}$, and ℓ . The particle starts at location \mathbf{x}_0 , which is at a distance d from the obstacle, and travels at a constant speed u . This defines the time to interaction t_d through the relationship $d = ut_d$. At the interaction time, the effects of diffusion mean the particle distribu- tion has characteristic width $D\sqrt{t_d}$. When the particle interacts interacts with the obstacle, it will get trapped if its location falls inside the concave	101
7.3	footprint, which has width ℓ	134 135

- 7.4 Probability of escape (line with circles, right scale) and expected time to escape conditional on escaping (solid line, left scale) a circular obstacle of radius R = 5, as a function of diffusion coefficient D. For D = 0, the particle gets trapped with probability one, while for D greater than 10^{-3} , the probability of escape is effectively one. The drop in probability of escape for Dgreater than 1 is due to the finite time of simulation. The obstacle was centered at $\mathbf{x} = (0,0)$ and the initial location of the particle was $\mathbf{x}_0 = (0, -20)$. The drift speed u was equal to 10. The dashed lines indicate one standard deviation above and below the mean expected time to escape. All quantities were computed based on 1,000 simulations for each set of parameter values.

137

Chapter 1

Introduction

Over the last ten years, technological advancements in sensory, motor, and computational capabilities have made it a real possibility for a legged robotic platform to traverse a diverse set of terrains [47, 62] and execute a variety of tasks [62, 88] on its own, with little to no outside intervention. However, there are still several technical challenges to be addressed in order to reach complete autonomy, where such a platform operates as an independent entity that communicates and cooperates with other intelligent systems, including humans. A central limitation for reaching this ultimate goal is modeling the *world* in which the robot is operating, the *tasks* it needs to execute, the *sensors* it is equipped with, and its level of *mobility*, all in a unified setting. This thesis presents a simple approach resulting in control strategies that are backed by a suite of formal correctness guarantees, allowing successful task execution on the target legged mobile platform, RHex [42, 121].

Many of the design considerations guiding the body of this work stem from the development of the new generation RHex platform in 2010 [42]. The first generation RHex platform had been almost a decade old. Its superior locomotion capabilities demonstrated over the years [25, 99, 121] had not been matched with adequate sensory and processing power because the platform could not support substantial improvements without adding more



Figure 1.1: The RHex robot on a forested hill.

weight to the robot or reducing battery runtime [42]. Many decisions made for the new platform, such as the body shape, battery chemistry, motors and motor drivers, power regulation and distribution, intra-robot communication interface, software infrastructure, and sensory and computational payload support, were shaped by the variety of tasks the robot would execute autonomously and the environments in which these tasks would take place.

With its versatile locomotion capabilities, RHex can be deployed in both indoor and outdoor settings. The modes of locomotion the platform needs to operate in and the sensory capabilities it needs to possess differ significantly from one setting to the other. In either case, one major challenge is to model the evolution of robot position and develop provably correct control strategies executing various exploration and navigation tasks.

One outdoor setting that the platform has been deployed in several times over the years is forested hills (Figure 1.1). Even in its early days, the robot was capable of adjusting its



Figure 1.2: The X-RHex robot climbing a stairwell.

locomotion pattern to adapt to inclines [80] to avoid diverging from the uphill direction, which could otherwise lead to robot failure due to flipping, potentially damaging its sensory equipment. In [62], the authors demonstrated an alternative to this locomotion pattern based approach, relying instead on autonomous steering towards the incline direction. The simplicity of this approach was intriguing, and it was intuitively clear why it was successful. However, a formal explanation was not nearly straightforward. This motivated us to model the environment, the task, sensory capabilities required, and the level of mobility in order to provide a correctness analysis and expand the range of locomotion speeds and inclines in which this behavior can be deployed [57].

Thanks to its stair ascent [99] and descent [25] gaits, the robot is capable of traversing multiple floors inside a building (Figure 1.2). Thus, as it was demonstrated in [62] and [58], many indoor exploration and navigation tasks can be implemented in a hybrid manner via transitions between floor traversal and stair climbing.

One of the virtues, and yet also a limitation, of the behaviors we have developed during

this thesis is the simplicity in the modeling decisions. Specifically, what happens when our assumptions regarding the world are violated? It is clear that our guarantees regarding the performance of the robot would not be viable anymore. The question is, how can we modify our strategy without dramatically altering our bottom-up approach to executing tasks on a legged platform? In [115], we consider a point particle agent governed by unconstrained second order dynamics and present a control law for interacting with more complex obstacle shapes while avoiding *entrapment* by an undesired fixed point. The formal extensions of this construction following our autonomous behavior design strategy is beyond the scope of this thesis. However, we do have an implementation on the RHex platform. In addition, [114] presents an alternative approach to the same entrapment problem.

The body of work that forms this thesis focuses on developing behaviors executable on the RHex legged platform [42, 121]. However, the lessons learned and methods developed can be applied to any mobile robotic platform that can afford the point particle, or horizontal unicycle motion model abstractions. Even when this assumption is not achievable, we speculate it is possible to expand the bottom-up approach presented in our work and find the sufficient lifting into the next simple motion model that can work as the gross simplification.

As an important note, various portions of this dissertation, including related text and figures, have been published in [57, 58, 114, 115]. All of these entries were written in collaboration with different co-authors. Even though we have included a complete account of all these efforts in the proceeding chapters, we specify their relation to this thesis in Section 1.2.

1.1 Motivation

In [62], which is the preliminary presentation of the two behaviors we focus on in this thesis, the authors emphasized the intrinsic value of these behaviors for intelligence, surveillance, and reconnaissance (ISR) as well as search and rescue operations [15, 104]. The increased frequency and severity of natural disasters, such as wildfires, due to climate change [13, 39, 141], makes it ever more important to channel advances in robotics for such purposes. As a versatile platform with ever growing locomotive capabilities [65], we believe RHex is a natural starting point for a new generation of robots utilized in disaster relief. In addition, our expanding work with geoscience researchers further reinforces the potential value of autonomous ascent of natural geological formations for many field science applications [112].

Despite its nearly self-evident value, the task of unassisted natural terrain ascent has long been thought to be challenging. Prior to [62], the literature on the autonomous hill ascent was limited to either simulation studies [6] or reports of empirical work at extremely slow speeds due to safety concerns [123, 151], with detailed terrain identification and mapping to avoid failures due to entrapment by small obstacles [81, 139]. Similarly, the only reports we have found documenting empirical work on autonomy over multiple flights of stairs prior to [62] mention a few anecdotal successes [152] or assume a very specific, simple landing geometry [134].

In contrast, the results of [62] suggested that both of these behaviors can be readily achieved if properly decomposed into an appropriately layered architecture. For this setup, the *mechanical intelligence* of the platform takes care of all the minor insults and small obstacle perturbations through intrinsic gait stability, while a model-based *planner* deals with the more serious obstacles.

In [62], the planner for the autonomous terrain ascent took the form of an ad hoc reactive scheme equipped with the simplest possible non-trivial world model—a smooth, diskpunctured surface (i.e., a sphere world [79])—and similarly ad hoc and stripped-down body frame sensor suite: an IMU and LIDAR. Startled by the very high empirical success rate over a variety of seemingly challenging natural landscapes in [62], we resolved to isolate the role of the world model by replacing the original ad hoc reactive layer with a provably correct sensorimotor scheme, i.e., one guaranteed to achieve successful ascent assuming an accurately modeled environment. Accordingly, [57] describes and demonstrates correctness of a sensorimotor scheme for a unicycle driving on a (sufficiently sparsely punctured) surface whose perceptual apparatus is limited to the same purely body frame (IMU and LIDAR) sensors. We both recover and extend the empirical trials of the precursor paper using the same legged platform, RHex [42, 121].

It is clear that no real forested environment will present the simplified geometry (sparse, convex obstacles) we formally posit. The value in carefully establishing its sufficiency for correctness of our simple, greedy, reactive navigation scheme reflects the interest in joining this work to a decades long tradition of multi-level [46] mobile robot architecture. The framework of a deliberative layer deploying reactive subsystems is well established in the field of robot navigation [97] as well as in the more general AI literature [59]. General consensus notwithstanding, the specifics of how to design and interface abstraction barriers has taken a long time to sort out for computational systems [1]. We believe that building sound and soundly inter-operative mechanical, reactive, and deliberative layers for robots will require a similarly delicate interplay between their formal and empirical properties. Reviewing the specifically related literature in Section 1.4, we will suggest the place that our empirically capable and formally well characterized architecture might occupy in the full navigation stack of an autonomous outdoor robot.

Finally, the deeper research question motivating this thesis is how much planning responsibility can be assigned to any purely reactive layer. Although we are only able to furnish conditions sufficient for gradient ascent of a particularly equipped robot, we are increasingly persuaded that they are also very close to be necessary for any uninformed greedy agent. By greedy, we mean that the agent's state ascends a Morse function (i.e., there is a smooth scalar valued map that is non-decreasing along any of its motions). It is well established that a perfectly informed gradient ascent is always possible (up to a set of zero measure initial conditions) [79]. By uninformed, we mean that the agent knows nothing in advance about the shape and location of the obstacles which must be encountered in real time and sensed in body-centric coordinates along the way. Two other very different recent treatments of uninformed greedy navigation to be mentioned below [8, 109], have arrived at sets of sufficient conditions quite similar to those we impose here: a topological sphere world [79] populated by sufficiently sparse and convex obstacles. We will return in the conclusion to a more speculative discussion of what our present results suggest about how to better construe the notion of a *reactive* agent and, thereby, its interface to a deliberative executive.

1.2 Relation to Published Work

The body of work forming this dissertation previously appeared in [57, 58, 114, 115]. In this section, I would like to describe my involvement in each of these publications.

- "Autonomous Legged Hill Ascent" [57]: I was the first author. I developed the theoretical work for point particle control law and its extensions to kinematic and dynamic unicycle agents. I implemented both control laws on the robot and conducted all the experimentation. In addition, I implemented the software framework and tuned a jogging-speed gait for the robot to be used for fast-pace locomotion. I also designed, implemented, and tested a battery monitoring solution that provided the power data in the specific resistance comparison experiments.
- "Autonomous Stairwell Ascent" [58]: I was the first author. Building on top of [62],
 I improved the perceptual capabilities of the robot, performed modifications and updates on the implemented behavior, and conducted a new set of experiments.
- "Dynamical Trajectory Replanning For Uncertain environments" [115]: I was the second author. I worked closely with the first author in the theoretical development phase and developed some of the theoretical proofs. In addition, with the assist of the first author, I developed the simulation environment, conducted extensive simulation studies for tuning the desired behaviors and investigating the performance.

• "A Drift-Diffusion Model For Robotic Obstacle Avoidance" [114]: I was the second author. I worked with the first author in experimental setup, implementation, and experimentation, where we utilized [57] as the base implementation to compare with.

1.3 Organization and Contributions

This thesis is composed of three main parts. Part I focuses on encoding tasks for a legged robot and covers Chapter 2 and Chapter 3. The main motivation behind the theoretical developments¹ established in Chapter 2 is to provide proper tools for the analysis of the control laws presented in Chapter 3. We start with some basic definitions on the stability of compact sets in Section 2.1. In Section 2.2, we proceed with a first order autonomous system described in (2.1) and provide definitions for Lyapunov (Section 2.2.1) and Chetaev (Section 2.2.2) functions accordingly. Then, we introduce the *Matching LaSalle* (ML) functions and utilize them for stability analysis of (2.1) in Section 2.2.3. We further investigate two special cases: embedding these systems into higher dimensional spaces (Section 2.3) and second order systems (Section 2.4).

Chapter 3 introduces an encoding strategy for a family of tasks where the task in hand can be reformulated as *autonomous hill ascent* with the *goal* of reaching a compact subset of the work space². More specifically, we provide a formal model yielding rigorous conditions on the geometric features of the environment under which our family of controllers can be guaranteed to succeed without relying on a more deliberative higher control layer. We accomplish this by incorporating knowledge of certain assumed parametric bounds that encode the mitigating features of the (otherwise unknown) putatively simplistic environment that afford success for our reactive (greedy) real-time motion controller. The nature of these parametric bounds lends insight into the essential problem constraints, enabling improved robot capabilities in comparison with [62] by affording operation on steeper hills and at

¹This chapter, as well as related text and figures, previously appeared in [57] as part of the appendices. ²This chapter, as well as related text and figures, previously appeared in [57].

higher speeds.³ Our controllers are based on a gradient vector field suitable for a fully actuated point particle ((3.34) in Section 3.2.1) that combines the vestibular perception of steepest ascent with avoidance of impassable obstacles as they come into exteroceptive view along the way. Their guarantees of convergence and obstacle avoidance follow from the properties of their associated ML function (Definition 2.2.8 in Section 2.2.3) that plays the role of a global Lyapunov function for the resulting closed loop systems.

In order to apply this idealized climbing *template* [41] control to a mechanically realistic robot model, we embed the point particle gradient field in the wrench space of the kinematic unicycle ((3.62) in Section 3.2.2.1) for slow paced climbing (Table 4.1 in Section 4.2.2.2) and, in turn, embed that first order vector field in a second order dynamical unicycle extension ((3.72) in Section 3.2.2.2) for fast paced climbing (Table 4.2 in Section 4.2.2.2). These models inherit the convergence properties. However, the specific subset of the free space that is kept positive invariant (i.e., the exact extent of the resulting safe states) proves very hard to characterize, so obstacle avoidance cannot be formally guaranteed.⁴

In Part II, we present two behaviors implemented on a legged robot, autonomous hill ascent (Chapter 4) and autonomous stairwell ascent (Chapter 5). In Chapter 4, we present the implementation details of the Autonomous Hill Ascent⁵ behavior, an application of task level autonomy wherein a legged robot achieves unassisted ascent of outdoor forested terrain in a variety of challenging settings, as depicted in Figure 1.3. Our work (in concert with the initial implementations reported in [62]) offers the first documented account of completely autonomous ascent over naturally populated hillsides by a robotic platform at speeds comparable to human uphill hiking and flat surface walking⁶. Our implementation on the RHex platform is tested in various challenging settings to showcase this empirical

³Experiments reported in [62] are run only at walking speed. In addition, they are limited to up to 17° slopes, whereas the slopes reported here as navigated by our upgraded controller include terrain up to 36° .

 $^{^{4}}$ In practice, none of our extensive experiments have witnessed an algorithmically generated collision and we conjecture that the positive invariant subset of these extended state spaces have a projection that is almost coincident with the obstacle free configuration space—see Section 3.2.2 for a more detailed discussion.

⁵This chapter, as well as related text and figures, previously appeared in [57].

⁶Based on an uphill hiking speed for a 10° hill of 0.56m/sec [83] and a walking speed on flat terrain of 1.46m/sec [72].



Figure 1.3: An illustration of the hill climbing controller implemented on RHex. The left image is a sample scene containing a single obstacle. The right image is a representation of the sensory inputs and the aggregate control law, all in body coordinates. The black point cloud is the LIDAR reading corresponding to the tree located on the robot's left side. The vectors represent (clockwise from -45°): (green) the hill gradient extracted from the IMU reading (4.1), (blue) the combined negative gradient (3.34), (black) resulting kinematic unicycle control input (3.62), (red) the component from the detected obstacle (3.30).

success, summarized in Table 4.1 in Section 4.2.2.2 and Table 4.2 in Section 4.2.2.2. These experiments constitute 20 long runs with direct distances anywhere from 12.5 meters to 96.8 meters, spanning almost a kilometer. The runtimes of these experiments vary from several seconds (19 seconds) to a few minutes (7 minutes 31 seconds), during which we report 90 instances of our methods enabling the robot to successfully avoid obstacles while maintaining autonomous hill ascent. In addition, we report 98 instances at which the robot's mechanical intelligence took care of circumstances that could otherwise hinder or even stall the robot's progress. In total we report 11 instances of failure, 6 of which were due to the robot's mechanical capabilities not being able to overcome the entrapment posed by the complex nature of the terrain, and an additional 2 due to obstacle shapes that violate our world model.

Chapter 5 focuses on a behavior that is generally acknowledged to hold great importance, yet still considerably difficult for existing man-portable mobile robots: the autonomous

climbing of multi-flight stairwells in indoor settings [113] (Figure 1.2)⁷. To accomplish this task, we replicate Chapter 3 and posit a very simple, deterministic world model and an equally simple deterministic perceptual model, along with a family of feedback controllers selected using (a sometimes slightly relaxed form of) sequential composition [24] in a manner that seems intuitively sufficient to achieve the specified navigation task. To the best of our knowledge, no previous authors have documented the completely autonomous ascent of general multi-floor stairwells. Combined with [62], the primary contribution we report in this chapter is our success in doing so on a variety of building interior styles, documented in the data tables of Section 5.3.

In Part III, we present two methods that could be incorporated into the behaviors from Part II to address world model violations, specifically regarding the obstacle shape assumption. Chapter 6 introduces a novel reference generator and tracking control architecture that enjoys appropriate stability properties and we present a handful of simulations demonstrating its ability to dislodge a simple point mass particle from cul-de-sac traps that block a naive tracking controller⁸. The energy costs calculated over a range of controller gains exhibit similar features for all three systems: a minimal threshold for escaping the trap, followed by a small range over which energy cost fluctuates, then a *sweet spot* exhibiting qualitatively *best* behavior that extends over a significant interval. This is followed by a roughly linear increase, and finally a mostly linear increase in cost, with many irregular cost fluctuations. A key feature of this architecture lies in its ability to isolate task specification, the reference subsystem (6.9), from the *replanner* (6.7), the encoding of how to handle unanticipated but structured obstacles to its execution. We end the chapter with a loose interpretation of the dynamical replanner for a unicycle agent with limited perceptual capabilities as described in Chapter 3.

In Chapter 7, we present a stochastic framework for modeling and analysis of robot nav-

⁷This chapter, as well as related text and figures, previously appeared in [58].

⁸This chapter, as well as related text and figures, previously appeared in [115].

igation in the presence of obstacles⁹. We show that, with appropriate assumptions, the probability of a robot avoiding a given obstacle can be reduced to a function of a single dimensionless parameter which captures all relevant quantities of the problem. This parameter is analogous to the Péclet number considered in the literature on mass transport in advection-diffusion fluid flows. Using the framework we also compute statistics of the time required to escape an obstacle in an informative case. The results of the computation show that adding noise to the navigation strategy can improve performance. Finally, we present experimental results on the RHex robotic platform, illustrating how this approach could result in performance improvements. For this, we start with Chapter 3, but with a parameter set that does not guarantee instability of the undesired equilibria as a demonstration of an obstacle that could entrap the robot. Instead, we utilize the presented approach to drive the robot from this spurious fixed point.

1.4 Review of Literature

Unicycle models—underactuated planar rigid bodies endowed with fore-aft and rotational control affordances—are widely used as templates [41] for unmanned ground vehicles. The unicycle control literature divides roughly into three families of problems: convergence to a fixed goal set—often a designated set of rigid placements [2, 33, 61, 88, 107, 120] or a path on the plane [2, 35, 44, 90, 125, 129], trajectory tracking with the aim of seeking and maintaining convergence to a time varying reference signal [28, 67, 84, 120, 153], and the generalization of these problem settings to multi-robot formations [37, 38, 86, 94, 120]. Our work takes its place within the first family concerning stabilization to a fixed set. However, unlike the work where the robot position and heading in relation to the goal is assumed to be available [2, 61, 88, 107, 120], our sensor model posits merely the availability of the instantaneous gradient vector (in body coordinates) of a fixed planar potential field to whose local maxima we seek, along with a stand-off sensor that can *see* planar obstacles along the

⁹This chapter, as well as related text and figures, previously appeared in [114].

way.

The problem of hill climbing (planar potential function ascent) with altitude-only sensory information is the focus of a large literature on *extremum seeking* [137], which has been applied as well in the reduced control affordance setting of unicycle-like vehicles [31, 87, 96, 154]. However, respecting the gravitational potential presented by a physical hill, our vestibular local gradient sensing model seems much more natural (readily instantiated by a standard commercial inertial measurement unit (IMU)) than the presumption of a device adequately sensitive to the small relative height variations afforded by forested hills and sloping parks. Moreover, the high control authority *dithering motions*, typically required to extract gradients from concentrations [96], turn out to be particularly undesirable for underactuated legged robotic platforms like RHex on physical hills. This is because the rapidly shifting cross-gradient motion threatens robot failure due to flipping [62].

The majority of the work on the problem of autonomous stair ascent is limited to detection of the stairs themselves [30, 36, 110, 124, 148, 150], climbing a single flight of stairs with very few steps [14, 91, 101, 102, 140], and autonomous transitions between flat surface walking and stair ascent under the control of an operator [14, 51, 101, 148].

Over the last two decades, there has been a growing interest in developing autonomy for offroad vehicles [69], [93]. These efforts benefited a great deal from Defense Advanced Research Projects Agency (DARPA) initiatives such as the Grand Challenge in 2005 [23], [27], and Learning Applied to Ground Vehicles (LAGR) program between 2004 and 2008 [60], [12], [81]. Both of these initiatives targeted large scale vehicles, where resulting research focused on deliberative navigation, terrain classification, mapping and learning. In contrast, we are interested in still less structured environments (natural forest rather than steep, sharply winding, unpaved roads or prepared terrain panels) and in exploring the capabilities of an intermediate, formally well-characterized reactive layer in between the mechanical plant and deliberative planner. Moreover, in place of the terrain-learning [111], environmentclassifying [82], and map-building [139] components traditionally associated with navigation in unstructured environments, we substitute a simple greedy strategy: a set-attractor basin generating an analytic vector field computed from local sensor-based measurements. The mechanical competence of the platform abstracts away the need to represent and reason about the details of the terrain. Encoding the task as a form of (punctured) hill climbing postpones the need for classification and maps at the reactive layer on which we focus with this work. We emphasize that it is the very simple nature of the encoded task—the very narrow assumptions the robot makes about the presences of only convex and well separated obstacles—that affords the greedy approach its success and its formal correctness.

Returning to the question of abstraction barriers in the navigation stack, parallel theoretical work [142], which integrates a different reactive motion planner [8] into a new task planner for indoor mobile manipulation [143] based upon angelic hierarchical search [95], suggests the importance of supporting abstract task deliberation with narrowly competent greedy motion controllers even in far more structured settings than the forested hills we explore here. In that work, sufficient conditions for correct local manipulation of known objects in a partially known environment are predicated upon a similarly naïve model of the unknowns. There, simulations show that the reactive motion planner relieves the abstract task planner of myriad geometric details (as here, the problem of circumventing simple but unanticipated obstacles) that would otherwise abort its execution, while typically completing its assigned subgoals even absent its conservative preconditions (as here, the assumption that the unanticipated obstacles are convex and sufficiently sparse). In Chapter 8, we discuss analogous next steps in developing a more complete navigation stack for the completely unstructured outdoor environments addressed by the naïvely competent (i.e., provably correct relative to narrowly conservative assumptions about the environment) reactive motion planner we present here.

Part I

Task Encoding for a Legged Robot

Chapter 2

Matching-LaSalle Functions and Stability

In this chapter, we present the theoretical developments enabling the analysis of the family of control laws introduced in Chapter 3. We start the chapter with some definitions on stability of compact sets in Section 2.1. We focus on compact sets with connected components and their stability to allow the *world* presented in Section 3.1.1 to be more complex in nature than a potential function with a sparse set of isolated equilibria. This increased complexity requires us to define a weaker notion of stability. Thus, we conclude Section 2.1 with the definition of *Almost Global Asymptotic Stability*.

Our goal in Section 2.2, is to develop a new type of potential function for the stability analysis of the autonomous system, (2.1), whose set of fixed points contains a compact subset with connected components. We first provide definitions of Lyapunov and Chetaev functions compatible with the problem setting, and then, we introduce the Matching LaSalle (ML) functions. These functions, by definition, can be utilized to generate local Lyapunov or Chetaev functions around fixed points. Thus, this system admitting an ML function becomes an important tool for investigating their stability. We analyze the control law for an unconstrained point particle presented in Section 3.2.1.2 via this approach.

Lastly, we turn our attention to embedding the system given in (2.1) into higher dimensional spaces (Section 2.3) and second order systems (Section 2.4). Under both scenarios, we investigate whether the stability results derived for the base system survives the corresponding operation. We utilize these findings in the analysis of the horizontal unicycle control laws presented in Section 3.2.2.

2.1 Basic Definitions

Consider a positive-invariant compact set, $\mathcal{X} \subset \mathbb{R}^m$, with the state variable, $\mathbf{x} \in \mathcal{X}$.

Definition 2.1.1 (point-set distance). For the compact set $\mathcal{G} \subset \mathcal{X}$,

$$|\mathbf{x}|_{\mathcal{G}} := \inf_{\bar{\mathbf{x}} \in \mathcal{G}} |\mathbf{x} - \bar{\mathbf{x}}|$$
 .

Definition 2.1.2 (local stability [4]). The compact set $\mathcal{G} \subset \mathcal{X}$ is called locally stable if

$$\forall \epsilon > 0, \quad \exists \beta > 0 : |\mathbf{x}_0|_{\mathcal{G}} < \beta \implies |\mathbf{x}(t, \mathbf{x}_0)|_{\mathcal{G}} \le \epsilon, \quad \forall t \ge 0.$$

Corollary 2.1.3. Let a compact set $\mathcal{G} \subset \mathcal{X}$ be composed of compact connected components, \mathcal{G}_j . If \mathcal{G}_j are all locally stable, then \mathcal{G} is locally stable. This result simply follows from that, $|\mathbf{x}|_{\mathcal{G}} = \min_j \{|\mathbf{x}|_{\mathcal{G}_j}\}.$

Definition 2.1.4 (set instability). The compact set $\mathcal{G} \subset \mathcal{X}$ is called unstable if it is NOT locally stable.

Definition 2.1.5 (local attractiveness). The compact set $\mathcal{G} \subset \mathcal{X}$ is called locally attractive if there exists a nonempty open set \mathcal{U} , satisfying $\mathcal{G} \subset \mathcal{U} \subset \mathcal{X}$, such that,

$$\forall \mathbf{x}_0 \in \mathcal{U}, \quad \lim_{t \to \infty} |\mathbf{x}(t, \mathbf{x}_0)|_{\mathcal{G}} = 0.$$
Lemma 2.1.6. Let a compact set $\mathcal{G} \subset \mathcal{X}$ be composed of compact connected components, \mathcal{G}_j . If \mathcal{G}_j are all locally attractive, then \mathcal{G} is locally attractive.

Proof. \mathcal{G}_j being locally attractive implies the existence of \mathcal{U}_j where

$$\forall \mathbf{x}_0 \in \mathcal{U}_j, \quad \lim_{t \to \infty} |\mathbf{x}(t, \mathbf{x}_0)|_{\mathcal{G}_j} = 0.$$

Since union of open sets are open, over the open set, and $|\mathbf{x}|_{\mathcal{G}} = \min_j \{ |\mathbf{x}|_{\mathcal{G}_j} \}$,

$$\forall \mathbf{x}_0 \in \mathcal{U} = \bigcup_j \mathcal{U}_j, \quad \lim_{t \to \infty} |\mathbf{x}(t, \mathbf{x}_0)|_{\mathcal{G}} = \lim_{t \to \infty} \min_j \left\{ |\mathbf{x}(t, \mathbf{x}_0)|_{\mathcal{G}_j} \right\} = 0.$$

and thus \mathcal{G} is locally attractive.

Definition 2.1.7 (local asymptotic stability). The compact set $\mathcal{G} \subset \mathcal{X}$ is called locally asymptotically stable *if it is locally stable and locally attractive.*

Corollary 2.1.8. Let a compact set $\mathcal{G} \subset \mathcal{X}$ be composed of compact connected components, \mathcal{G}_j . If \mathcal{G}_j are all locally asymptotically stable, then \mathcal{G} is locally asymptotically stable from Corollary 2.1.3 and Lemma 2.1.6.

Definition 2.1.9 (global attractiveness). The compact set $\mathcal{G} \subset \mathcal{X}$ is called globally attractive *if*,

$$\forall \mathbf{x}_0 \in \mathcal{X}, \quad \lim_{t \to \infty} |\mathbf{x}(t, \mathbf{x}_0)|_{\mathcal{G}} = 0.$$

Definition 2.1.10 (Global Asymptotic Stability (GAS)). The compact set $\mathcal{G} \subset \mathcal{X}$ is called Globally Asymptotically Stable (GAS) if it is locally stable and globally attractive.

Definition 2.1.11 (almost-global attractiveness). The compact set $\mathcal{G} \subset \mathcal{X}$ is called almostglobally attractive if $\exists \mathcal{N} \subset \mathcal{X}$ with empty interior such that,

$$\forall \mathbf{x}_0 \in \mathcal{X} - \mathcal{N}, \quad \lim_{t \to \infty} |\mathbf{x}(t, \mathbf{x}_0)|_{\mathcal{G}} = 0.$$

Remark 2.1.12. Definition 2.1.11 is slightly different than the version presented in [4]. Instead of introducing and working with Lebesgue measure, we find it more convenient (and sufficient for our intended applications) to associate the almost notion with sets whose complements have empty interior (regardless of whether or not they have measure zero). In particular, an invariant set possessing a non-empty unstable manifold, cannot comprise the forward limit of any open set, hence attracts almost no initial conditions in our sense.

Definition 2.1.13 (Almost-Global Asymptotic Stability (AGAS) [4]). The compact set $\mathcal{G} \subset \mathcal{X}$ is called Almost-Globally Asymptotically Stable (AGAS) if it is locally stable and almost-globally attractive.

2.2 Potential Functions

Consider the autonomous system,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}),\tag{2.1}$$

with $\mathbf{f}: \mathcal{X} \to \mathbb{R}^m$ locally Lipschitz, where \mathcal{X} is compact and positive-invariant. In addition, let $\mathcal{C} \subset \mathcal{X}$ be the set of all fixed points, $\mathcal{C} := \{\mathbf{x} \in \mathcal{X} : \mathbf{f}(\mathbf{x}) = \mathbf{0}\}$. Assume that \mathcal{C} contains a compact subset, \mathcal{G} , composed of compact connected components, \mathcal{G}_j , and let $\mathcal{S} := \mathcal{C} - \mathcal{G}$ be its complement.

2.2.1 Lyapunov Functions

Definition 2.2.1 (Lyapunov Function). Consider the system (2.1), and a compact connected subset of its fixed points, $\mathcal{G} \subset \mathcal{C}$. If the continuously differentiable function, γ , defined over some open subset, $\mathcal{U} \subset \mathcal{X}$ with $\mathcal{G} \subset \mathcal{U}$, satisfies,

• $\alpha_1(|\mathbf{x}|_{\mathcal{G}}) \leq \gamma(\mathbf{x}) \leq \alpha_2(|\mathbf{x}|_{\mathcal{G}})$, where α_1, α_2 both belong to class \mathcal{K}_{∞} ,¹⁰

 $^{^{10}\}alpha$: $\mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to class \mathcal{K}_{∞} if $\alpha(0) = 0, \forall a, b \in \mathbb{R}_{\geq 0}, a > b \implies \alpha(a) > \alpha(b)$, and

•
$$\nabla \gamma(\mathbf{x})^T \mathbf{f}(\mathbf{x}) \leq 0$$
 with $\nabla \gamma(\mathbf{x})^T \mathbf{f}(\mathbf{x}) = 0$ if and only if $\mathbf{x} \in \mathcal{G}$,

then γ is a Lyapunov function and we say \mathcal{G} locally admits a Lyapunov function.

Remark 2.2.2. The definition above borrows the ISS-Lyapunov function definition in [5], eliminates the input, and relaxes the Lie-derivative upper bound as in [132]. Let us introduce an input term to (2.1), $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{u}_{\mathbf{x}}$, where $\mathbf{u}_{\mathbf{x}} : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ is a locally essentially bounded function. According to [132], an ISS-Lyapunov function, $\gamma : \mathcal{U} \subset \mathcal{X} \to \mathbb{R}_{\geq 0}$, with respect to a compact set, $\mathcal{G} \subset \mathcal{U}$, becomes a Lyapunov function for the zero-input system, $\mathbf{u}_{\mathbf{x}} = \mathbf{0}$. Then, \mathcal{G} is locally asymptotically stable.

Corollary 2.2.3. Following Remark 2.2.2, for the system (2.1), if \mathcal{G} locally admits a Lyapunov function, γ , as defined above, then it is locally asymptotically stable.

2.2.2 Chetaev Functions

Definition 2.2.4 (Chetaev Function [70]). For the system in (2.1), let $\mathbf{x}_c \in C$, and consider a continuously differentiable function, $\varrho : \mathcal{U} \subset \mathcal{X} \to \mathbb{R}$, defined over an open set around this critical point, $\mathbf{x}_c \in \mathcal{U}$. Assume $\varrho(\mathbf{x}_c) = 0$, and there exists \mathbf{x}_0 with arbitrarily small $|\mathbf{x}_0 - \mathbf{x}_c|$ such that $\varrho(\mathbf{x}_0) > 0$. Choose ε such that $\mathcal{B}_{\varepsilon} := {\mathbf{x} \in \mathcal{X} : |\mathbf{x} - \mathbf{x}_c| \le \varepsilon} \subset \mathcal{U}$, and let $\mathcal{M} := {\mathbf{x} \in \mathcal{B}_{\varepsilon} : \varrho(\mathbf{x}) > 0}$. ϱ is called a Chetaev function around this critical point if $\forall \mathbf{x} \in \mathcal{M}, \ \dot{\varrho}(\mathbf{x}) > 0$, and we say \mathbf{x}_c admits a Chetaev function.

Lemma 2.2.5 (Thm. 3.3 of [70]). For the system given in (2.1), if $\mathbf{x}_c \in C$ admits a Chetaev function, ϱ , as defined in Definition 2.2.4, then it is locally unstable.

Lemma 2.2.6. Consider a twice continuously differentiable function, $\varphi : \mathcal{U} \subset \mathcal{X} \to \mathbb{R}_{\geq 0}$, and a point, $\mathbf{x}_c \in \mathcal{U}$, where $\nabla \varphi(\mathbf{x}_c) = \mathbf{0}$. In addition, assume that the Hessian of φ evaluated at \mathbf{x}_c , $H_{\varphi}(\mathbf{x}_c) = \mathsf{D}^2_{\mathbf{x}}\varphi(\mathbf{x}_c)$, has a negative eigenvalue, $\lambda_{\varphi} < 0$, accompanied with $\overline{\lim_{a\to\infty} \alpha(a) = \infty}$ the eigenvector, \mathbf{v}_{φ} . Then, for $\mathbf{x} \in {\{\mathbf{x}_c + \epsilon \mathbf{v}_{\varphi} : \epsilon > 0\}}$, the function,

$$\varrho(\mathbf{x}) := \varphi(\mathbf{x}_c) - \varphi(\mathbf{x}), \tag{2.2}$$

is positive for arbitrarily small ϵ .

Proof. Consider the Taylor expansion of ρ ,

$$\begin{split} \varrho(\mathbf{x}) &= \varrho(\mathbf{x}_c) + \nabla \varrho(\mathbf{x}_c)^T [\mathbf{x} - \mathbf{x}_c] + [\mathbf{x} - \mathbf{x}_c]^T H_{\varrho}(\mathbf{x}_c) [\mathbf{x} - \mathbf{x}_c] + o(|\mathbf{x} - \mathbf{x}_c|^2) \\ &= -[\mathbf{x} - \mathbf{x}_c]^T H_{\varphi}(\mathbf{x}_c) [\mathbf{x} - \mathbf{x}_c] + o(|\mathbf{x} - \mathbf{x}_c|^2), \end{split}$$

where $o(|\mathbf{x} - \mathbf{x}_c|^2)$ represents the collection of all the higher order terms, $O(|\mathbf{x} - \mathbf{x}_c|^k)$ with k > 2. When evaluated for $\mathbf{x} = \mathbf{x}_c + \epsilon \mathbf{v}_{\varphi}$ with $\epsilon > 0$,

$$\varrho(\mathbf{x}_c + \epsilon \mathbf{v}_{\varphi}) = -\epsilon^2 \mathbf{v}_{\varphi}^T H_{\varphi}(\mathbf{x}_c) \mathbf{v}_{\varphi} + o(\epsilon^2) \ge -\epsilon^2 \lambda_{\varphi} + o(\epsilon^2),$$

is positive for arbitrarily small ϵ since $\lambda_{\varphi} < 0$.

Proposition 2.2.7. For the system given in (2.1), consider a fixed point $\mathbf{x}_c \in C$, and assume there exists a function, φ , satisfying the conditions laid out in Lemma 2.2.6. Then, the function in (2.2) is a Chetaev function and consequently \mathbf{x}_c is unstable.

Proof. Following Lemma 2.2.6, define the function ρ as in (2.2). Notice first that $\rho(\mathbf{x}_c) = 0$, and $\dot{\rho} = -\dot{\varphi} \ge 0$. Now, following Definition 2.2.4, define a set \mathcal{U} . Consequently, $\dot{\rho}(\mathbf{x}) > 0$ when $\mathbf{x} \in \mathcal{U}$. From Lemma 2.2.6 there exists a direction at which ρ stays positive as $\mathbf{x} \to \mathbf{x}_c$. Therefore, ρ is a Chetaev function, and from Lemma 2.2.5, \mathbf{x}_c is unstable.

2.2.3 Matching LaSalle (ML) Functions

Definition 2.2.8 (Matching LaSalle (ML) Function). For the system given in (2.1), the continuously differentiable function, $\varphi : \mathcal{X} \to \mathbb{R}$, is called a Matching LaSalle (ML) function if,

- $\nabla \varphi(\mathbf{x}) = \mathbf{0}$ if and only if $\mathbf{x} \in C$,
- $\dot{\varphi} = \nabla \varphi(\mathbf{x})^T \mathbf{f}(\mathbf{x}) \leq 0$, where $\dot{\varphi} = 0$ if and only if $\mathbf{x} \in \mathcal{C}$,
- for every \mathcal{G}_j , the function, $\gamma(\mathbf{x}) := \varphi(\mathbf{x}) \varphi(\mathbf{x}_c)$, with $\mathbf{x}_c \in \mathcal{G}_j$, is a Lyapunov function,
- for every $\mathbf{x}_c \in \mathcal{S}$, the function, $\varrho(\mathbf{x}) := -\gamma(\mathbf{x}) = \varphi(\mathbf{x}_c) \varphi(\mathbf{x})$, is a Chetaev function,

in which case we say the system admits an ML function.

Theorem 2.2.9. If the system (2.1) admits an ML function, φ , then C is globally attractive, G is locally asymptotically stable, and S is locally unstable.

Proof. From LaSalle's Invariance Principle (Theorem 3.4 of [70]), for every initial state, $\mathbf{x}_0 \in \mathcal{X}$, $\lim_{t\to\infty} \mathbf{x}(t, \mathbf{x}_0) \in \mathcal{M} \subseteq \mathcal{C}$ where \mathcal{M} is the biggest invariant subset of \mathcal{C} . Since in this case \mathcal{C} is composed of fixed points, it is invariant and thus $\mathcal{M} = \mathcal{C}$, meaning \mathcal{C} is globally attractive.

For every compact connected component, \mathcal{G}_j , the function, γ , given in Definition 2.2.8 is a local Lyapunov function, and from Corollary 2.2.3, \mathcal{G}_j is locally asymptotically stable. Then, via Corollary 2.1.8, we conclude \mathcal{G} is locally asymptotically stable.

Lastly, for all $\mathbf{x}_c \in S$, ρ from Definition 2.2.8 is a Chetaev function, and thus \mathbf{x}_c is locally unstable according to Lemma 2.2.5.

Proposition 2.2.10. For the system given in (2.1), assume there exists an ML function, φ . If at each $\mathbf{x}_c \in S$, the Jacobian, $\mathsf{D}_{\mathbf{x}}\mathbf{f}(\mathbf{x}_c)$, has a positive eigenvalue, then \mathcal{G} is AGAS. *Proof.* From Theorem 2.2.9, existence of the ML function, φ , implies \mathcal{C} is globally attractive, \mathcal{G} is locally asymptotically stable, and \mathcal{S} is locally unstable. In addition, for any $\mathbf{x}_c \in \mathcal{S}$, the Jacobian, $\mathsf{D}_{\mathbf{x}}\mathbf{f}(\mathbf{x}_c)$, has a positive eigenvalue. From Center Manifold Theorem (Thm. 3.2.1 of [48]), there exists an unstable manifold around this equilibrium point that is at least one dimensional. This implies the stable manifold around \mathcal{S} has empty interior, and thus, \mathcal{G} is almost-globally attractive. Since \mathcal{G} is already locally stable, we conclude \mathcal{G} is AGAS.

Corollary 2.2.11. For the system given in (2.1), assume there exists an ML function, φ , implying from Theorem 2.2.9 that C is globally attractive, and G is locally stable. If $S = \emptyset$, then C = G, and thus, it is GAS.

2.3 Embedding a System: A Special Case

In this section, we consider a special case of embedding a system that admits an ML function as in Definition 2.2.8 into a higher dimensional space. By the term *embedding* we mean that the forward limit set of the original system is embedded in that of the higher dimensional system with the same local stability properties at each point. A more desirable goal would be to start with an AGAS system, and *anchor* the original system in the sense of [41] whereby there is an invariant embedding of the original state space with conjugate restriction dynamics (implying among other consequences that the embedding of an AGAS set remains AGAS in the embedding space), however the degeneracies associated with our present constructions do not seem to afford that stronger conclusion. In particular, absent our present ability to find an explicit unstable eigenvalue in the linearized dynamics of the (higher dimensional) embedding system corresponding to that of the (lower dimensional) embedded model, we achieve our local stability results by recourse to a Chetaev function, postponing the local conjugacy (and consequently a global AGAS property for the embedding system) to conjectural status for future study.

Theorem 2.3.1. For the system given in (2.1), assume there exists an ML function, φ ,

as in Definition 2.2.8, and thus, from Theorem 2.2.9, C is globally attractive, G is locally asymptotically stable, and S is locally unstable. In addition, assume that for all $\mathbf{x}_c \in S$, the function is twice continuously differentiable over an open neighborhood, its Hessian evaluated at the critical point, $H_{\varphi}(\mathbf{x}_c)$, has a negative eigenvalue, $\lambda_{\varphi} < 0$, and its curvatures are bounded, $\sup_{\mathbf{x}_c \in S} \|H_{\varphi}(\mathbf{x}_c)\| \leq \kappa_{\varphi} < \infty$.

Now, consider the system,

$$\dot{\mathbf{z}} = \mathbf{g}(\mathbf{z}),\tag{2.3}$$

with $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^{m+k}$ and k > 0, where \mathcal{Y} is compact, and $\mathbf{g} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{m+k}$ is locally Lipschitz. In addition, assume $\mathcal{C} \times \mathcal{Y}$ is the set of all fixed points. If there exists a nonnegative continuously differentiable function, $\eta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, and a positive constant, ν_{φ} , satisfying $\eta(\mathbf{x}, \mathbf{y}) \leq \frac{\nu_{\eta}}{2} \nabla \varphi^{T}(\mathbf{x}) \nabla \varphi(\mathbf{x})$ with $\nu_{\eta} \kappa_{\varphi} < \nu_{\varphi}$, such that,

$$\varphi_E(\mathbf{x}, \mathbf{y}) := \nu_{\varphi} \varphi(\mathbf{x}) + \eta(\mathbf{x}, \mathbf{y}), \qquad (2.4)$$

has a non-positive Lie derivative, $\dot{\varphi}_E \leq 0$, with $\dot{\varphi}_E = 0$ if and only if $\mathbf{x} \in C$, then φ_E is an ML function for the system (2.3).

Proof. The first two conditions in Definition 2.2.8 are already assumed to be true. To investigate the conditions regarding Lyapunov and Chetaev functions, we will use the bounding relations,

$$\nu_{\varphi}\varphi(\mathbf{x}) \leq \varphi_{E}(\mathbf{x}, \mathbf{y}) \leq \nu_{\varphi}\varphi(\mathbf{x}) + \frac{\nu_{\eta}}{2}\nabla\varphi^{T}(\mathbf{x})\nabla\varphi(\mathbf{x}).$$
(2.5)

For the system (2.1), consider a compact connected component of the stable fixed point set, \mathcal{G}_j , which admits $\gamma(\mathbf{x}) = \varphi(\mathbf{x}) - \varphi(\mathbf{x}_c)$ as a local Lyapunov function according to Definition 2.2.8. For the higher dimensional system(2.3), consider the component $\mathcal{G}_j \times \mathcal{Y}$, and the function,

$$\begin{split} \gamma_E(\mathbf{x}, \mathbf{y}) &:= \varphi_E(\mathbf{x}, \mathbf{y}) - \varphi_E(\mathbf{x}_c, \mathbf{y}) \\ &= \nu_{\varphi} \gamma(\mathbf{x}) + \eta(\mathbf{x}, \mathbf{y}), \quad (\mathbf{x}, \mathbf{y}) \in \mathcal{U} \times \mathcal{Y}, \end{split}$$

with any $\mathbf{x}_c \in \mathcal{G}_j$. From (2.5), we have $\nu_{\varphi}\gamma(\mathbf{x}) \leq \gamma_E(\mathbf{x}, \mathbf{y}) \leq \nu_{\varphi}\gamma(\mathbf{x}) + \frac{\nu_{\eta}}{2}\nabla\varphi^T(\mathbf{x})\nabla\varphi(\mathbf{x})$. Then, γ_E is nonnegative and its Lie derivative, $\dot{\gamma}_E = \dot{\varphi}_E$, is non-positive, where both the function and its Lie derivative are zero if and only if $\mathbf{x} \in \mathcal{G}_j$. As the result, $\mathcal{G}_j \times \mathcal{Y}$ locally admits γ_E as a Lyapunov function.

For the system (2.1), consider a critical point, $\mathbf{x}_c \in S$. The Hessian of the upper bound function in (2.5) is,

$$\begin{split} \mathsf{D}_{\mathbf{x}}^{2} \left\{ \nu_{\varphi} \varphi(\mathbf{x}) + \frac{\nu_{\eta}}{2} \nabla \varphi^{T} \nabla \varphi \right\} \Big|_{\mathbf{x} = \mathbf{x}_{c}} &= \mathsf{D}_{\mathbf{x}} \left\{ \nu_{\varphi} \nabla \varphi + \nu_{\eta} H_{\varphi} \nabla \varphi \right\} \Big|_{\mathbf{x} = \mathbf{x}_{c}} \\ &= \nu_{\varphi} H_{\varphi}(\mathbf{x}_{c}) + \nu_{\eta} H_{\varphi}^{2}(\mathbf{x}_{c}) + \nu_{\eta} \mathsf{D}_{\mathbf{x}} \left\{ H_{\varphi} \right\} \Big|_{\mathbf{x} = \mathbf{x}_{c}} \nabla \varphi(\mathbf{x}_{c}) \\ &= \nu_{\varphi} H_{\varphi}(\mathbf{x}_{c}) + \nu_{\eta} H_{\varphi}^{2}(\mathbf{x}_{c}). \end{split}$$

Let \mathbf{v}_{φ} be the eigenvector for the negative eigenvalue of $H_{\varphi}(\mathbf{x}_c)$, $\lambda_{\varphi} < 0$. Then,

$$\begin{split} \mathsf{D}_{\mathbf{x}}^{2} \left\{ \nu_{\varphi} \varphi(\mathbf{x}) + \frac{\nu_{\eta}}{2} \nabla \varphi^{T} \nabla \varphi \right\} \Big|_{\mathbf{x} = \mathbf{x}_{c}} \mathbf{v}_{\varphi} &= \left[\nu_{\varphi} H_{\varphi}(\mathbf{x}_{c}) + \nu_{\eta} H_{\varphi}^{2}(\mathbf{x}_{c}) \right] \mathbf{v}_{\varphi} \\ &= \left[\nu_{\varphi} \lambda_{\varphi} + \nu_{\eta} \lambda_{\varphi}^{2} \right] \mathbf{v}_{\varphi} \\ &= \left[\nu_{\varphi} + \nu_{\eta} \lambda_{\varphi} \right] \lambda_{\varphi} \mathbf{v}_{\varphi}, \end{split}$$

where,

$$\nu_{\varphi} + \nu_{\eta}\lambda_{\varphi} \ge \nu_{\varphi} - \nu_{\eta}\kappa_{\varphi} > 0,$$

since $\nu_{\eta}\kappa_{\varphi} < \nu_{\varphi}$, and thus, the Hessian of the upper bound in (2.5) evaluated at $\mathbf{x}_c \in S$ has a negative eigenvalue, $[\nu_{\varphi} + \nu_{\eta}\lambda_{\varphi}]\lambda_{\varphi}$. For the higher dimensional system in (2.3), each member of corresponding set of critical points, $(\mathbf{x}_c, \mathbf{y})$, with $\mathbf{y} \in \mathcal{Y}$, has the function,

$$egin{aligned} arrho_E(\mathbf{x},\mathbf{y}) &:= arphi_E(\mathbf{x}_c,\mathbf{y}) - arphi_E(\mathbf{x},\mathbf{y}) \ &=
u_arphi arphi(\mathbf{x}_c) - ig[
u_arphi arphi(\mathbf{x}) + \eta(\mathbf{x},\mathbf{y})ig], \end{aligned}$$

where $\varrho_E(\mathbf{x}_c, \mathbf{y}) = 0$, and its Lie derivative $\dot{\varrho}_E(\mathbf{x}, \mathbf{y}) = -\dot{\varphi}_E(\mathbf{x}, \mathbf{y})$ is positive in the vicinity when $\mathbf{x} \neq \mathbf{x}_c$. From (2.5), this time we have $\nu_{\varphi}\varphi(\mathbf{x}_c) - \left[\nu_{\varphi}\varphi(\mathbf{x}) + \frac{\nu_{\eta}}{2}\nabla\varphi^T(\mathbf{x})\nabla\varphi(\mathbf{x})\right] \leq \varrho_E(\mathbf{x}, \mathbf{y})$. Since the Hessian of $\nu_{\varphi}\varphi(\mathbf{x}) + \frac{\nu_{\eta}}{2}\nabla\varphi^T(\mathbf{x})\nabla\varphi(\mathbf{x})$ evaluated at $\mathbf{x}_c \in \mathcal{S}$ has a negative eigenvalue, from Lemma 2.2.6, there exists a direction at which the function's lower bound, $\nu_{\varphi}\varphi(\mathbf{x}_c) - \left[\nu_{\varphi}\varphi(\mathbf{x}) + \frac{\nu_{\eta}}{2}\nabla\varphi^T(\mathbf{x})\nabla\varphi(\mathbf{x})\right]$, is positive arbitrarily close to the critical point. Therefore, ϱ_E is a valid Chetaev function.

2.4 Second Order Embedding

This section provides a generalization for the second order embedding of a first order system previously discussed in [73, 115], where we show that an *exact cancellation* term in the control policy is not required.

For the system given in (2.1), assume there exists an ML function, φ . Consider the *second* order lift, [73, 115],

$$\ddot{\mathbf{x}} = \dot{\mathbf{f}} - \nu_{\varphi} \nabla \varphi - \nu_{\mathbf{f}} \left[\dot{\mathbf{x}} - \mathbf{f} \right], \tag{2.6}$$

with the positive constants, ν_{φ} and $\nu_{\mathbf{f}}$, where $\dot{\mathbf{f}} := \mathsf{D}_{\mathbf{x}}\mathbf{f} \cdot \dot{\mathbf{x}}$, and the following potential function,

$$\varphi_S(\mathbf{x}, \dot{\mathbf{x}}) := \nu_{\varphi} \,\varphi(\mathbf{x}) + \frac{1}{2} \,|\dot{\mathbf{x}} - \mathbf{f}(\mathbf{x})|^2, \qquad (2.7)$$

whose Lie derivative,

$$\begin{split} \dot{\varphi}_{S} &= \nu_{\varphi} \nabla \varphi^{T} \dot{\mathbf{x}} + \left[\ddot{\mathbf{x}} - \dot{\mathbf{f}} \right]^{T} \left[\dot{\mathbf{x}} - \mathbf{f} \right] \\ &= \nu_{\varphi} \nabla \varphi^{T} \left[\mathbf{f} + \left[\dot{\mathbf{x}} - \mathbf{f} \right] \right] - \nu_{\varphi} \nabla \varphi^{T} \left[\dot{\mathbf{x}} - \mathbf{f} \right] - \nu_{\mathbf{f}} |\dot{\mathbf{x}} - \mathbf{f}|^{2} \\ &= \nu_{\varphi} \nabla \varphi^{T} \mathbf{f} - \nu_{\mathbf{f}} |\dot{\mathbf{x}} - \mathbf{f}|^{2} \\ &\leq 0, \end{split}$$

is zero if and only if $\mathbf{x} \in \mathcal{C}$ and $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ at the same time. Without further investigating whether φ_S is an ML function, notice that this approach relies on exact cancellation through the use of original ML function gradient, $\nabla \varphi$, which may not be available as a measurement. To address this issue, we introduce the following theorem:

Theorem 2.4.1. Consider the special case of (2.1), admitting a factorization,

$$\mathbf{f}(\mathbf{x}) := B(\mathbf{x})\mathbf{u}(\mathbf{x}),\tag{2.8}$$

where $B: \mathcal{X} \to \mathbb{R}^{m \times l}$, with $0 < l \leq m$, is full-rank matrix with bounded norm, $\kappa_B := \max_{\mathbf{x} \in \mathcal{X}} \|B(\mathbf{x})\| < \infty$. Assume there exists an ML function, φ , as defined in Definition 2.2.8, satisfying,

$$\nabla \varphi^T \mathbf{f} = \nabla \varphi^T B \, \mathbf{u} \le -\nu_{\mathbf{x}} \, |\nabla \varphi|^{\beta} \,, \tag{2.9}$$

with $\beta > 1$ and $\nu_{\mathbf{x}} > 0$.

Now, consider the following system,

$$\dot{\mathbf{x}} = B(\mathbf{x})\mathbf{k} \tag{2.10a}$$

$$\dot{\mathbf{k}} = \mathbf{u}_{\mathbf{k}}$$
 (2.10b)

with $\mathbf{k}, \mathbf{u}_{\mathbf{k}} \in \mathbb{R}^{l}$. Under the control law,

$$\mathbf{u}_{\mathbf{k}}(\mathbf{x}, \mathbf{k}) := \dot{\mathbf{u}}(\mathbf{x}) - \sigma_{\mathbf{f}}(\mathbf{k} - \mathbf{u}(\mathbf{x}))[\mathbf{k} - \mathbf{u}(\mathbf{x})], \qquad (2.11)$$

where $\dot{\mathbf{u}} := \mathsf{D}_{\mathbf{x}} \mathbf{u} \cdot \dot{\mathbf{x}}$, and $\sigma_{\mathbf{f}}$ is a locally Lipschitz scalar valued function with lower bound $0 < \nu_{\mathbf{f}} \leq \sigma_{\mathbf{f}}$, this system admits,

$$\varphi_S(\mathbf{x}, \mathbf{k}) := \nu_{\varphi} \varphi(\mathbf{x}) + \frac{\beta - 1}{\beta} |\mathbf{k} - \mathbf{u}|^{\frac{\beta}{\beta - 1}}, \qquad (2.12)$$

as an ML function.

Proof. For the system (2.10) combined with the control law (2.11), the set of fixed points is $C \times \{0\}$. To show this, observe that B is a full-rank matrix, and thus, the fixed points of (2.10a) satisfy $\mathbf{k} = \mathbf{0}$, whereas, the unique fixed point of (2.10b) is $\mathbf{k} = \mathbf{u}$.

Consider the change of coordinates, $(\mathbf{x}, \mathbf{k}) \mapsto (\mathbf{x}, \mathbf{r})$ where $\mathbf{r} := \mathbf{k} - \mathbf{u}(\mathbf{x})$. Resulting equivalent system,

$$\dot{\mathbf{x}} = B(\mathbf{x})[\mathbf{u}(\mathbf{x}) + \mathbf{r}]$$
(2.13a)

$$\dot{\mathbf{r}} = -\sigma_{\mathbf{f}}(\mathbf{r})\,\mathbf{r},\tag{2.13b}$$

has the same fixed point set. In addition, we can express (2.12) as,

$$\varphi_S(\mathbf{x}, \mathbf{r}) := \nu_{\varphi} \varphi(\mathbf{x}) + \frac{\beta - 1}{\beta} |\mathbf{r}|^{\frac{\beta}{\beta - 1}}, \qquad (2.14)$$

which is continuously differentiable since $\beta/(\beta-1) > 1$. Its gradient,

$$\nabla \varphi_S(\mathbf{x}, \mathbf{r}) = \begin{bmatrix} \nu_{\varphi} \nabla \varphi(\mathbf{x}) & |\mathbf{r}|^{\frac{2-\beta}{\beta-1}} \mathbf{r} \end{bmatrix}^T,$$

vanishes if and only if $(\mathbf{x}, \mathbf{r}) \in \mathcal{C} \times \{\mathbf{0}\}$. Its Lie derivative with respect to (2.13) is,

$$\begin{split} \dot{\varphi}_{S} &= \nu_{\varphi} \nabla \varphi^{T} \dot{\mathbf{x}} + |\mathbf{r}|^{\frac{2-\beta}{\beta-1}} \mathbf{r}^{T} \dot{\mathbf{r}} \\ &= \nu_{\varphi} \nabla \varphi^{T} B[\mathbf{u} + \mathbf{r}] + |\mathbf{r}|^{\frac{2-\beta}{\beta-1}} \mathbf{r}^{T} [-\sigma_{\mathbf{f}}(\mathbf{r}) \mathbf{r}] \\ &= \nu_{\varphi} \nabla \varphi^{T} B \mathbf{u} + \nu_{\varphi} \nabla \varphi^{T} B \mathbf{r} - \sigma_{\mathbf{f}}(\mathbf{r}) |\mathbf{r}|^{\frac{\beta}{\beta-1}} \\ &\leq -\nu_{\varphi} \nu_{\mathbf{x}} |\nabla \varphi|^{\beta} + \nu_{\varphi} \kappa_{B} |\nabla \varphi| |\mathbf{r}| - \nu_{\mathbf{f}} |\mathbf{r}|^{\frac{\beta}{\beta-1}} , \end{split}$$

where second term of this expression is bounded by,¹¹

$$\nu_{\varphi} \left| \nabla \varphi \right| \left| \mathbf{r} \right| \le \nu_{\varphi}^{\beta} \kappa_{B}^{\beta} \left[\frac{2}{\nu_{\mathbf{f}}} \right]^{\beta-1} \left| \nabla \varphi \right|^{\beta} + \frac{\nu_{\mathbf{f}}}{2} \left| \mathbf{r} \right|^{\frac{\beta}{\beta-1}},$$

resulting in,

$$\begin{split} \dot{\varphi}_{S} &\leq \left[-\nu_{\varphi}\nu_{\mathbf{x}} + \nu_{\varphi}^{\beta}\kappa_{B}^{\beta} \left[\frac{2}{\nu_{\mathbf{f}}} \right]^{\beta-1} \right] \ |\nabla\varphi|^{\beta} + \left[\frac{\nu_{\mathbf{f}}}{2} - \nu_{\mathbf{f}} \right] |\mathbf{r}|^{\frac{\beta}{\beta-1}} \\ &\leq \left[-\nu_{\mathbf{x}} + \nu_{\varphi}^{\beta-1}\kappa_{B}^{\beta} \left[\frac{2}{\nu_{\mathbf{f}}} \right]^{\beta-1} \right] \nu_{\varphi} \left| \nabla\varphi \right|^{\beta} - \frac{\nu_{\mathbf{f}}}{2} \left| \mathbf{r} \right|^{\frac{\beta}{\beta-1}}. \end{split}$$

With the choice, $\nu_{\varphi} = \frac{\nu_{\mathbf{f}}}{2} \left[\frac{\nu_{\mathbf{x}}}{2\kappa_B^{\beta}} \right]^{\frac{1}{\beta-1}}$, the upper bound becomes,

$$\begin{split} \dot{\varphi}_{S} &\leq -\frac{\nu_{\mathbf{x}}}{2} \nu_{\varphi} \left| \nabla \varphi \right|^{\beta} - \frac{\nu_{\mathbf{f}}}{2} \left| \mathbf{r} \right|^{\frac{\beta}{\beta-1}} \\ &\leq 0, \end{split}$$

where $\dot{\varphi}_S = 0$ if and only if $(\mathbf{x}, \mathbf{r}) \in \mathcal{C} \times \{\mathbf{0}\}.$

For the system (2.1), from Definition 2.2.8, \mathcal{G}_j locally admits the Lyapunov function, $\gamma(\mathbf{x}) = \varphi(\mathbf{x}) - \varphi(\mathbf{x}_c)$ with $\mathbf{x}_c \in \mathcal{G}_j$. Now, for the system (2.13), consider the component $\mathcal{G}_j \times \{\mathbf{0}\}$,

¹¹Let $a, b, \varsigma, \varepsilon, \omega \in \mathbb{R}_{>0}$, and consider the term ςab . If $\varsigma a \leq \varepsilon b^{1/\omega}$ then $\varsigma ab \leq \varepsilon b^{\left[1+1/\omega\right]}$. Otherwise, $\varsigma ab \leq \varepsilon^{-\omega} \varsigma^{\left[1+\omega\right]} a^{\left[1+\omega\right]}$. We conclude, $\varsigma ab \leq \varepsilon b^{\left[1+1/\omega\right]} + \varepsilon^{-\omega} \varsigma^{\left[1+\omega\right]} a^{\left[1+\omega\right]}$. Here, $\varsigma := \nu_{\delta} \kappa_{B}, \varepsilon := \nu_{\mathbf{r}}/2$, $a := |\nabla \delta|, b := |\mathbf{r} - \mathbf{g}|$, and $\omega := \beta - 1$.

where under the subset $\mathcal{U} \times \mathcal{R}$, the function,

$$egin{aligned} &\gamma_S(\mathbf{x},\mathbf{r}) := arphi_S(\mathbf{x},\mathbf{r}) - arphi_S(\mathbf{x}_c,\mathbf{0}) \ &=
u_arphi \gamma(\mathbf{x}) + rac{eta - 1}{eta} \, |\mathbf{r}|^{rac{eta}{eta - 1}} \,, \quad \mathbf{x}_c \in \mathcal{G}_j, \end{aligned}$$

is nonnegative and its Lie derivative, $\dot{\gamma}_S = \dot{\varphi}_S$ is non-positive over $\mathcal{U} \times \mathcal{R}$, where both the function and its Lie derivative are zero if and only if $(\mathbf{x}, \mathbf{r}) \in \mathcal{G}_j \times \{\mathbf{0}\}$, and thus $\mathcal{G}_j \times \{\mathbf{0}\}$ locally admits γ_S as a Lyapunov function.

For the system (2.1), from Definition 2.2.8, every $\mathbf{x}_c \in S$ admits the Chetaev function, $\varrho(\mathbf{x}) = \varphi(\mathbf{x}_c) - \varphi(\mathbf{x})$. Now, for the system (2.13), consider the critical point, $(\mathbf{x}_c, \mathbf{0})$, and the function,

$$egin{aligned} arrho_S(\mathbf{x},\mathbf{r}) &:= arphi_S(\mathbf{x}_c,\mathbf{0}) - arphi_S(\mathbf{x},\mathbf{r}) \ &=
u_arphi arrho(\mathbf{x}) - rac{eta-1}{eta} \, |\mathbf{r}|^{rac{eta}{eta-1}} \, , \end{aligned}$$

where $\rho_S(\mathbf{x}_c, \mathbf{0}) = 0$ and its Lie derivative, $\dot{\rho}_S(\mathbf{x}, \mathbf{r}) = -\dot{\varphi}_S(\mathbf{x}, \mathbf{r})$ is positive in the vicinity when $\mathbf{x} \neq \mathbf{x}_c$. Moreover, since ρ is a Chetaev function, for $\mathbf{r} = \mathbf{0}$, there is a direction at which $\rho_S(\mathbf{x}, \mathbf{r}) = \nu_{\varphi} \rho(\mathbf{x})$ is positive arbitrarily close to the critical point. Therefore ρ_S is a Chetaev function.

Corollary 2.4.2. In Theorem 2.4.1, if the Lie derivative upper bound (2.9) is replaced by a combination of terms, $\nabla \varphi(\mathbf{x})^T \mathbf{f} \leq -\sum_{i=0}^j \nu_{\mathbf{x}i} |\nabla \varphi|^{\beta_i}$, where $\nu_{\mathbf{x}i} > 0$ and $\beta_i > 1$ are constant, and j is a positive and finite constant, then replacing the function in (2.12) with,

$$\varphi_S(\mathbf{x}, \mathbf{r}) := \sum_{i=0}^{j} \left[\nu_{\varphi_i} \varphi(\mathbf{x}) + \frac{\beta_i - 1}{\beta_i} |\mathbf{k} - \mathbf{u}|^{\frac{\beta_i}{\beta_i - 1}} \right], \qquad (2.15)$$

suffices to reach the same conclusion.

Corollary 2.4.3 (AGAS). In Theorem 2.4.1, for the system (2.1) where the vector field obeys (2.8), assume that for every $\mathbf{x}_c \in S$, the Jacobian, $\mathsf{D}_{\mathbf{x}}\mathbf{f}(\mathbf{x}_c)$, has a positive eigenvalue.

Observe that the Jacobian of (2.13) evaluated at any fixed point $(\mathbf{x}_c, \mathbf{0})$ with $\mathbf{x}_c \in \mathcal{C}$, is,

$$\mathsf{D}_{\mathbf{x},\mathbf{r}} \left\{ \begin{array}{cc} B[\mathbf{u}(\mathbf{x}) + \mathbf{r}] \\ & -\sigma_{\mathbf{f}}(\mathbf{r}) \, \mathbf{r} \end{array} \right\} \Big|_{\mathbf{x}_{c} \in \mathcal{C}, \, \mathbf{r} = \mathbf{0}} = \begin{bmatrix} \mathsf{D}_{\mathbf{x}} \mathbf{f}(\mathbf{x}_{c}) & B(\mathbf{x}_{c}) \\ & -\sigma_{\mathbf{f}}(\mathbf{0}) \mathbf{I} \end{bmatrix},$$

which is a block triangular matrix and its eigenvalues are composed of the eigenvalues of $D_{\mathbf{x}}\mathbf{f}(\mathbf{x}_c)$ and $-\sigma_{\mathbf{f}}(\mathbf{0})\mathbf{I}$. Since $D_{\mathbf{x}}\mathbf{f}(\mathbf{x}_c)$, has a positive eigenvalue, from Proposition 2.2.10, we conclude $\mathcal{G} \times \{\mathbf{0}\}$ is AGAS.

Corollary 2.4.4 (Second Order Embedding). In Theorem 2.4.1, B = I is equivalent to the second order system $\ddot{\mathbf{x}} = \mathbf{u}_{\mathbf{k}}$. With the control law (2.11), the resulting system turns into a second order embedding,

$$\ddot{\mathbf{x}} = \dot{\mathbf{f}} - \nu_{\mathbf{f}} [\dot{\mathbf{x}} - \mathbf{f}], \qquad (2.16)$$

which is (2.6) without the undesired gradient term.

Chapter 3

Task Encoding for a Legged Robot

In this chapter, we present our bottom-up approach to task encoding for horizontal unicycle agents. Our strategy is suited for tasks that can be formulated as reaching a compact subset of the work space. In Section 3.1, we lay out our modeling decisions with regards to the environment the robot operates in, the task it is expected to execute, and the sensors it is equipped with. In Section 3.2, we focus on an unconstrained planar point particle agent. We present a reactive obstacle interaction model, and develop a combined control law maintaining autonomous hill ascent while avoiding obstacles. By utilizing the tools we have developed in Chapter 2, we show in Theorem 3.2.6 that the goal set representing the task can be made AGAS (as in Definition 2.1.13) through proper parameter selection. Then, we turn our attention to horizontal unicycle models with Section 3.2.2. In Section 3.2.2.1, we extend the point particle control law presented in the previous section, into the kinematic unicycle model. By utilizing Theorem 2.3.1, we show in Theorem 3.2.11 that the local stability properties of the goal set and other fixed points are maintained. In Section 3.2.2.2, we *lift* the kinematic unicycle control law into the dynamic unicycle model. This time we utilize Theorem 2.4.1, and we show in Theorem 3.2.15 that the local stability profile of all the fixed points stay the same.

3.1 World and Task

In this section, we first introduce a representation of the *world* the robots we consider in this thesis are assumed to operate in. This representation abstracts out many details regarding the actual environment, encouraging the construction of simplified sensorimotor algorithms for task execution in the expectation that the robot's mechanical *preflexes* [19, 53] will handle the rest. We proceed to formulate the autonomous hill ascent task for a robot that, for now, is a fully actuated point particle. The robot's goal—achieving a peak or ridge—is represented by the compact set of critical points of the punctured terrain height function. In this work, all our execution strategies incorporate the gradient of this task function to reach the goal set. Applying this gradient to achieve successful task execution on a physical robot platform requires accounting for the limitations of the underlying robot dynamics, which is the focus of Section 3.2.

3.1.1 The World Model

A summary of the model and accompanying assumptions presented in this section can be found in Table 3.1.

Definition 3.1.1 (Terrain). A terrain is represented by some unknown height function,

$$h \in C^{\infty}[\mathbb{R}^2, \mathbb{R}]. \tag{3.1}$$

Not only is h unknown, it is not necessarily a metrically full scale accurate copy of the actual work space, rather to be imagined as sufficiently smoothed and thus absent of spatial frequencies much below the robot's body length. The operative assumption is that any patch of such terrain is readily traversable by the robot's standard gaits outside of obstacle regions. The set of *obstacles* is given by excessively steep grades,

$$\mathcal{O} := \left\{ \mathbf{p} \in \mathbb{R}^2 : \left| \nabla h(\mathbf{p}) \right| \ge \Gamma_h \right\},\tag{3.2}$$

where Γ_h is an upper bound on the grades below which the robot is assumed to operate (i.e., obey the presumed *plant* model) without any failures.

Definition 3.1.2 (Hill). A hill is defined as a terrain punctured by a disjoint union of d obstacles,

$$\mathcal{O} = \prod_{i=1}^{d} \mathcal{O}_i, \tag{3.3}$$

where each obstacle, \mathcal{O}_i , is a closed disk parametrized by its center, \mathbf{p}_i , and radius, ρ_i ,

$$\mathcal{O}_i := \left\{ \mathbf{p} \in \mathbb{R}^2 : |\mathbf{p} - \mathbf{p}_i| \le \rho_i \right\}, \tag{3.4}$$

which we assume is unknown a priori but can be perceived upon its entrance into a sensor's spherical footprint radius, or sensor range, ρ_s ,

$$\mathcal{D}_{i} := \left\{ \mathbf{p} \in \mathbb{R}^{2} : |\mathbf{p} - \mathbf{p}_{i}| \le \rho_{i} + \rho_{s} \right\},$$
(3.5)

which we call the obstacle region. Lastly, the open annulus representing the free work space in the vicinity of each obstacle,

$$\mathcal{R}_i := \mathcal{D}_i - \mathcal{O}_i, \tag{3.6}$$

is called a region of interest.

Figure 3.1 illustrates an obstacle region, \mathcal{D}_i , composed of \mathcal{R}_i , and \mathcal{O}_i .

Now, consider the following set of assumptions imposed throughout our analysis to achieve the desired stability results.



Figure 3.1: Obstacle and sensor models. The thickness of the region of interest, \mathcal{R}_i , is set to be identical to the sensor range. Since the obstacles are assumed to be disk shaped in Definition 3.1.2, sensor output is a simply a rescaled version of the relative position of the obstacle center, as given in (3.28).

Assumption 3.1.3. The obstacles, \mathcal{O}_i , situated over the hill are,

1. suitably located: the obstacle regions, \mathcal{D}_i , do not contain any critical points,

$$\forall \mathbf{p} \in \mathcal{D}_i, \, \nabla h(\mathbf{p}) \neq 0, \tag{3.7}$$

2. suitably sized: obstacle radii are bounded within a fixed (but unknown) interval,

$$\rho_{min} \le \rho_i \le \rho_{max},\tag{3.8}$$

3. suitably separated: individual obstacle regions do not intersect,

$$\mathcal{D}_i \cap \mathcal{D}_j = \emptyset. \tag{3.9}$$

Assumption 3.1.4. The global terrain component,

$$\mathcal{T} := \left\{ \mathbf{p} \in \mathbb{R}^2 : h(\mathbf{p}) \ge h_{min} \right\},\tag{3.10}$$

is a compact and contractible subset of the plane for some suitably chosen (but unknown) h_{min} , in which case there is an (again unknown) upper bound,

$$h_{max} := \max_{\mathbf{p}\in\mathcal{T}} \left\{ h(\mathbf{p}) \right\} > h_{min}. \tag{3.11}$$

In addition, the boundary of the global terrain component, $\partial \mathcal{T}$, does not intersect with the obstacle regions,

$$\mathcal{D}_i \cap \partial \mathcal{T} = \emptyset, \tag{3.12}$$

and does not contain any critical points,

$$\forall \mathbf{p} \in \partial \mathcal{T}, \nabla h(\mathbf{p}) \neq 0. \tag{3.13}$$

Under these assumptions,

$$\mathcal{P} := \mathcal{T} - \mathcal{O}, \tag{3.14}$$

is a topological sphere world in the sense of [79], inserted into which a rigid body must be confined to a free space given as a correspondingly punctured subset of the planar rigid transformation group.¹² Of course, the robot has no prior knowledge of the shape and location of any obstacles but will use the sensed gradient field as an effective internal beacon to circumvent them in a manner we detail later in Section 3.2.1.

 $^{1^{12}}$ When the free space boundaries are exactly known, and the body is fully actuated then the problem admits an essentially global navigation function as was established in [116].

	Definition	Expression	
р	robot position	$\mathbf{p} \in \mathbb{R}^2$	
h	terrain function	$h \in C^{\infty}[\mathbb{R}^2, \mathbb{R}]$	(3.1)
Γ_h	obstacle grade threshold		
\mathcal{O}	set of obstacles	$\mathcal{O} := \left\{ \mathbf{p} \in \mathbb{R}^2 : \left abla h(\mathbf{p}) ight \geq \Gamma_h ight\}$	(3.2)
\mathbf{p}_i	obstacle location	$\mathbf{p}_i \in \mathbb{R}^2$	
$ ho_i$	obstacle radius	$\rho_i \in \mathbb{R}_{>0}$	
$ ho_{min}$	minimum obstacle radius		
$ ho_{max}$	maximum obstacle radius		
\mathcal{O}_i	disk obstacle	$\mathcal{O}_i := \left\{ \mathbf{p} \in \mathbb{R}^2 : \left \mathbf{p} - \mathbf{p}_i ight \leq ho_i ight\}$	(3.4)
ρ_s	sensor range		
\mathcal{D}_i	obstacle region	$\mathcal{D}_i := \left\{ \mathbf{p} \in \mathbb{R}^2 : \left \mathbf{p} - \mathbf{p}_i \right \le ho_i + ho_S ight\}$	(3.5)
\mathcal{R}_i	region of interest	$\mathcal{R}_i := \mathcal{D}_i - \mathcal{O}_i$	(3.6)
h_{min}	min hill elevation		
\mathcal{T}	global terrain component	$\mathcal{T} := \left\{ \mathbf{p} \in \mathbb{R}^2 : h(\mathbf{p}) \ge h_{min} ight\}$	(3.10)
h_{max}	max hill elevation	$h_{max} := \max_{\mathbf{p} \in \mathcal{T}} \left\{ h(\mathbf{p}) \right\}$	(3.11)
\mathcal{P}	work space	$\mathcal{P}:=\mathcal{T}-\mathcal{O}$	(3.14)
	Assumption		
O	composed of d disjoint disks	$\mathcal{O} = \coprod_{i=1}^d \mathcal{O}_i$	(3.3)
	suitably located	$\forall \mathbf{p} \in \mathcal{D}_i, \nabla h(\mathbf{p}) \neq 0$	(3.7)
\mathcal{O}_i	suitably sized	$ \rho_{min} \le \rho_i \le \rho_{max} $	(3.8)
	suitably separated	$\mathcal{D}_i\cap\mathcal{D}_j=\emptyset$	(3.9)
$\partial \tau$	does not intersect with obstacle regions	$\mathcal{D}_i \cap \partial \mathcal{T} = \emptyset$	(3.12)
01	does not contain any critical points	$\forall \mathbf{p} \in \partial \mathcal{T}, \nabla h(\mathbf{p}) \neq 0$	(3.13)

WORLD MODEL

Table 3.1: Fixed relations and (unknown) geometric parameters underlying the assumed world model

3.1.2 Task Model

A summary of the task model detailed in this section can be found in Table 3.2.

We define the task of *autonomous hill ascent* as reaching some local maximum of the terrain function, h, independent of the robot's initial state. In the present work, we assume that the hill is time invariant so that the gradient of its height function, ∇h , is purely a function of robot position. We find it convenient to adopt the traditions of the robot navigation literature (e.g. [79]) by inverting the terrain function as follows.

Definition 3.1.5 (Task). For the hill introduced in Definition 3.1.2, the smooth function,

$$\phi(\mathbf{p}) := h_{min} - h(\mathbf{p}), \tag{3.15}$$

is called a task function [76], because the task of autonomous hill ascent on \mathcal{P} is encoded as reaching its critical point set,

$$\mathcal{C}_{\phi} := [\nabla \phi]^{-1}(\mathbf{0}). \tag{3.16}$$

Observe that (3.7) from Assumption 3.1.3 implies,

$$\mathcal{C}_{\phi} \cap \mathcal{R}_i = \emptyset. \tag{3.17}$$

Based on Definition 3.1.2, the gradient of the task function, $\nabla \phi$, is bounded,

$$|\nabla \phi(\mathbf{p})| < \Gamma_h, \, \forall \mathbf{p} \in \mathcal{P}.$$
(3.18)

In addition, from (3.17), a common positive lower bound to its magnitude, Ω_h , exists over the regions of interest, \mathcal{R}_i ,

$$\exists \Omega_h > 0 : \Omega_h \le |\nabla \phi(\mathbf{p})|, \, \forall \mathbf{p} \in \mathcal{R}_i, \forall i \in \{1, \dots, d\}.$$
(3.19)

Because \mathcal{P} is compact and the task function is assumed to be smooth (at least twice continuously differentiable), it has bounded curvatures over the terrain. Thus, defining the Hessian,

$$H_{\phi} := \mathsf{D}_{\mathbf{p}}^{T}(\nabla \phi) = \mathsf{D}_{\mathbf{p}}^{2}(\phi), \qquad (3.20)$$

we have,

$$\kappa_{\phi} := \sup_{\mathbf{p} \in \mathcal{P}} \|H_{\phi}(\mathbf{p})\| < \infty.$$
(3.21)

Assumption 3.1.6. For the task function in Definition 3.1.5, the critical point set, C_{ϕ} has a compact subset, \mathcal{G}_{ϕ} , composed of n compact connected components, \mathcal{G}_{ϕ}^{j} , and let $\mathcal{S}_{\phi} := C_{\phi} - \mathcal{G}_{\phi}$ denote its complement, where,

1. Every $\mathcal{G}_{\phi}{}^{j}$ locally admits,

$$\gamma_{\phi}(\mathbf{p}) = \phi(\mathbf{p}) - \phi(\mathbf{p}_c), \quad \mathbf{p}_c \in \mathcal{G}_{\phi}^{\,j}, \tag{3.22}$$

as a Lyapunov function (as in Definition 2.2.1),

2. At every critical point, $\mathbf{p}_c \in \mathcal{S}_{\phi} := \mathcal{C}_{\phi} - \mathcal{G}_{\phi}$, the Hessian has a negative eigenvalue,

$$\forall \mathbf{p}_c \in \mathcal{S}_{\phi}, \quad \exists \lambda_{\phi} < 0, \exists \mathbf{v}_{\phi} : H_{\phi}(\mathbf{p}_c) \mathbf{v}_{\phi} = \lambda_{\phi} \mathbf{v}_{\phi}. \tag{3.23}$$

Consequently, from Proposition 2.2.7,

$$\varrho_{\phi}(\mathbf{p}) = \phi(\mathbf{p}_c) - \phi(\mathbf{p}), \quad \mathbf{p}_c \in \mathcal{S}_{\phi}, \tag{3.24}$$

is a Chetaev function.

For an unconstrained planar agent, $\mathbf{p} \in \mathcal{P}$, in the presence of no obstacles, $\mathcal{O} = \emptyset$, $\mathcal{P} = \mathcal{T}$ is bounded by a level set of ϕ and, hence, is positive invariant. Based on Assumption 3.1.6, the system,

$$\dot{\mathbf{p}} = -\nabla\phi(\mathbf{p}),\tag{3.25}$$

admits ϕ as an ML function as defined in Definition 2.2.8. From Theorem 2.2.9, this implies C_{ϕ} is globally attractive, \mathcal{G}_{ϕ} is locally asymptotically stable, and \mathcal{S}_{ϕ} is locally unstable. In

addition, since its Jacobian evaluated at any unstable equilibrium, $-H_{\phi}(\mathbf{p}_c)$ with $\mathbf{p}_c \in S_{\phi}$, has a positive eigenvalue, we conclude via Proposition 2.2.10 that \mathcal{G}_{ϕ} is Almost Globally Asymptotically Stable (AGAS) under the flow (3.25) induced by the gradient field.

TASK MODEL					
	Definition	Expression			
ϕ	task function	$\phi(\mathbf{p}) := h_{min} - h(\mathbf{p})$	(3.15)		
\mathcal{C}_{ϕ}	task critical set	$\mathcal{C}_\phi := [abla \phi]^{-1}(0)$	(3.16)		
$ \nabla \phi $	task gradient magnitude	$ \nabla \phi(\mathbf{p}) < \Gamma_h, \forall \mathbf{p} \in \mathcal{P}.$	(3.18)		
Ω_h	min task gradient magnitude over \mathcal{R}_i	$\exists \Omega_h > 0 : \Omega_h \le \nabla \phi(\mathbf{p}) , \forall \mathbf{p} \in \mathcal{R}_i, \forall i \in \{1, \dots, d\}$	(3.19)		
H_{ϕ}	task Hessian	$H_{\phi} := D^2_{\mathbf{p}}(\phi)$	(3.20)		
κ_{ϕ}	task curvature bound	$\kappa_{\phi} := \sup_{\mathbf{p} \in \mathcal{P}} \ H_{\phi}(\mathbf{p})\ < \infty$	(3.21)		
	Assumption				
\mathcal{G}_{ϕ}	compact connected components	$\mathcal{G}_{\phi} = \bigcup \mathcal{G}_{\phi}{}^j \subset \mathcal{C}_{\phi}$			
	local Lyapunov functions for every $\mathcal{G}_{\phi}{}^{j}$	$\gamma_{\phi}(\mathbf{p}) = \phi(\mathbf{p}) - \phi(\mathbf{p}_c), \mathbf{p}_c \in \mathcal{G}_{\phi}{}^j$	(3.22)		
	complement subset	$\mathcal{S}_\phi := \mathcal{C}_\phi - \mathcal{G}_\phi$			
\mathcal{S}_{ϕ}	Hessian with negative eigenvalue	$\forall \mathbf{p}_c \in \mathcal{S}_{\phi}, \ \exists \lambda_{\phi} < 0, \exists \mathbf{v}_{\phi} : H_{\phi}(\mathbf{p}_c) \mathbf{v}_{\phi} = \lambda_{\phi} \mathbf{v}_{\phi}$	(3.23)		
	Chetaev functions for every $\mathbf{p}_c \in \mathcal{S}_{\phi}$	$arrho_{\phi}(\mathbf{p})=\phi(\mathbf{p}_{c})-\phi(\mathbf{p})$	(3.24)		

Table 3.2: Nomenclature and (unknown) geometric parameters underlying the task model. In the absence of obstacles, the task would be achieved by simply following the terrain gradient field (3.25).

3.1.3 Sensor Models

A summary of sensor models including varying assumptions on available measurements for different robot models can be found in Table 3.3.

We assume a sensory suite that has no direct measurement of the robot's position, as in a GPS-denied environment [11, 147]. The available sensors include a vestibular sensor that captures the terrain gradient, ∇h , and a limited exteroceptive sensor that can detect nearby (up to ρ_s away) obstacles. Specifically, for the obstacle \mathcal{O}_i , the robot can only sense location of the closest point on the obstacle relative to its own location,

$$\boldsymbol{\ell}_{i}(\mathbf{p})\Big|_{\mathcal{R}_{i}} := \left[\arg\min_{\bar{\mathbf{p}}\in\mathcal{O}_{i}}|\mathbf{p}-\bar{\mathbf{p}}|\right] - \mathbf{p}.$$
(3.26)

The geometric nature of this measurement in relation to an obstacle region can be found in Figure 3.1. Note that, since \mathcal{O}_i is a disk with the radius ρ_i , the obstacle distance can be written as,

$$\left|\boldsymbol{\ell}_{i}(\mathbf{p})\right|\Big|_{\mathcal{R}_{i}} = \min_{\bar{\mathbf{p}}\in\mathcal{O}_{i}}\left|\mathbf{p} - \bar{\mathbf{p}}\right| = \left|\mathbf{p} - \mathbf{p}_{i}\right| - \rho_{i}, \qquad (3.27)$$

and the relative obstacle location becomes,

$$\boldsymbol{\ell}_{i}(\mathbf{p})\Big|_{\mathcal{R}_{i}} = -\frac{|\mathbf{p} - \mathbf{p}_{i}| - \rho_{i}}{|\mathbf{p} - \mathbf{p}_{i}|} [\mathbf{p} - \mathbf{p}_{i}], \qquad (3.28)$$

although the robot has no prior information about \mathbf{p}_i or ρ_i (nor do we find it useful to attempt any estimation of those parameters).

We assume an inertial frame fixed at some absolute position in \mathcal{P} with a fixed orientation, relative to which we introduce world frame coordinates for the robot's position, $\mathbf{p} \in \mathcal{P}$, and heading, $\theta \in \mathbb{S}^1$, jointly written as $\mathbf{q} := [\mathbf{p}, \theta] \in SE(2)$. We place a frame in the robot's body whose origin in world frame coordinates is located at \mathbf{p} and whose orientation, aligned with its fore-aft direction of motion, is given by the world frame vector direction $\mathbf{n}(\theta) := [\cos \theta, \sin \theta]^T$, so the transformation into body coordinates, \mathbf{x}^b , of a vector, $\mathbf{x} \in \mathbb{R}^2$, in world frame coordinates is given by $\mathbf{x}^b = R^T(\theta) \mathbf{x}$, where $R(\theta) = [\mathbf{n}(\theta), \bar{\mathbf{n}}(\theta)]$, and $\bar{\mathbf{n}}(\theta) := [-\sin \theta, \cos \theta]^T$.

For the horizontal unicycle models, relative obstacle location, ℓ_i , is available through a local omni-directional range and bearing sensor (in our implementation, a Laser Imaging, Detection And Ranging (LIDAR) unit detailed in Section 4.1.1.1), $\ell_i^{b}(\mathbf{p})$. Similarly, the terrain function gradient, ∇h , is available as a local measurement through a vestibular sensor (in our implementation, an Inertial Measurement Unit (IMU) detailed in Section 4.1.1.2), $\nabla h^{b}(\mathbf{p})$.

	Definition	Expression
ℓ_i	relative obstacle location	$\left. \boldsymbol{\ell}_i(\mathbf{p}) \right _{\mathcal{R}_i} := \arg\min_{\bar{\mathbf{p}} \in \mathcal{O}_i} \left \mathbf{p} - \bar{\mathbf{p}} \right - \mathbf{p} (3.26)$
θ	robot heading	$ heta \in \mathbb{S}^1$
\mathbf{q}	robot pose	$\mathbf{q} := \begin{bmatrix} \mathbf{p}, \theta \end{bmatrix} \in SE(2)$
$\mathbf{n}(\theta)$	robot direction vector	$\mathbf{n}(\theta) := \left[\cos\theta, \sin\theta\right]^T$
$\bar{\mathbf{n}}(\theta)$	orthogonal direction vector	$\bar{\mathbf{n}}(heta) := \begin{bmatrix} -\sin heta, \cos heta \end{bmatrix}^T$
$R(\theta)$	robot rotation matrix	$R(\theta) = \left[\mathbf{n}(\theta), \mathbf{\bar{n}}(\theta)\right]$
$\{.\}^b$	body frame transformation	$\mathbf{x}^b := R^T(heta) \mathbf{x}, \mathbf{x} \in \mathbb{R}^2$

SENSOR MODELS

Table 3.3: Sensor models

3.2 Robot Control

As noted at the end of Section 3.1.2, $\dot{\mathbf{p}} = -\nabla \phi$ is a provably correct ascent strategy for a fully actuated point particle on an obstacle-free terrain. However, this leaves open the question of how to handle hills obstructed by unknown obstacles, even those as simply shaped and situated as introduced in Definition 3.1.2. Here, we first construct a control law for a fully actuated point robot (Section 3.2.1.2) using the previously defined task model (Section 3.1.2) that guarantees collision-free essentially global convergence to the goal set, \mathcal{G}_{ϕ} , subject to the foregoing assumptions (Definition 3.1.2) about the interaction of the terrain with the obstacle field (Section 3.2.1.1). We then extend the construction, successively, to the kinematic (Section 3.2.2.1) and dynamic (Section 3.2.2.2) unicycle models with no further restrictions on the terrain parameters.

3.2.1 Autonomous Point Particle Hill Ascent with Obstacle Avoidance

The obstacle model and the point particle control policy arising from combining this model with the initial task function, including conditions guaranteeing successful performance, are summarized in Table 3.4.

	Definition	Expression		
ψ_i	obstacle function	$\psi_i \Big _{\mathcal{R}_i} = \frac{1}{2\rho_S} \big[\rho_S - \boldsymbol{\ell}_i(\mathbf{p}) \big]^2$	(3.29)	
φ	combined task function	$\varphi := \phi + \nu_{\psi} \sum_{i=1}^{d} \psi_i$	(3.33)	
$ u_{\psi}$	control parameter			
C	combined task critical set	$\mathcal{C} = \mathcal{G}_\phi \cup \mathcal{S}_\phi \cup \mathcal{S}_\psi$	(3.35)	
\mathcal{S}_ψ	spurious critical set	$\mathcal{S}_{\psi} := \bigcup_{i=1}^{d} \mathcal{S}_{\psi_i}$	(3.36)	
\mathcal{S}_{ψ_i}	spurious critical set component	$\mathcal{S}_{\psi_i} := \{\mathbf{p} \in \mathcal{R}_i : \nabla \phi = -\nu_{\psi} \nabla \psi_i\}$	(3.36)	
	Assumptions about the sensors and the environment			
$\ell_i, \nabla \phi$	available as measurements			
$\Omega_h, \Gamma_h, \kappa_\phi, \rho_{max}$	known parameters			
$ ho_{max}$	max obstacle radius sufficiently small	$ ho_{max} < rac{\Omega_h}{\kappa_\phi}$	(3.55)	
	Conditions on the Design Parameters			
$ u_{\psi} $	control parameter sufficiently large	$\nu_{\psi} > \Gamma_h$	(3.56)	
$ ho_{S}$	sensor range sufficiently small	$\rho_{\mathcal{S}} < \frac{1}{\left[1 - \frac{\Omega_h}{\nu_{\psi}}\right]} \left[\frac{\Omega_h}{\kappa_{\phi}} - \rho_{max}\right]$	(3.57)	

POINT PARTICLE CONTROL

Table 3.4: Summary of point particle control definitions, assumptions, and sufficient conditions for successful execution. Additional summaries of corresponding world, task, and sensor models are located in Table 3.1, Table 3.2, and Table 3.3, respectively.

3.2.1.1 Obstacle Model

The simplified assumptions about obstacle shapes and locations afford an intuitively straightforward sensor-based repelling local field.

Definition 3.2.1 (Obstacle Function). For the obstacle \mathcal{O}_i , the obstacle function, ψ_i , is defined to be a local potential function,

$$\psi_{i} = \begin{cases} \frac{1}{2\rho_{\mathcal{S}}} \left[\rho_{\mathcal{S}} - |\boldsymbol{\ell}_{i}(\mathbf{p})| \right]^{2} &, \quad \mathbf{p} \in \mathcal{R}_{i} \\ 0 &, \quad otherwise. \end{cases}$$
(3.29)

When constrained in its region of interest, the obstacle function can be rewritten as,

$$\psi_i \Big|_{\mathcal{R}_i} = \frac{1}{2\rho_{\mathcal{S}}} \big[\rho_{\mathcal{S}} - |\boldsymbol{\ell}_i(\mathbf{p})| \big]^2 = \frac{1}{2\rho_{\mathcal{S}}} \big[\rho_i + \rho_{\mathcal{S}} - |\mathbf{p} - \mathbf{p}_i| \big]^2.$$

Its gradient can be derived as,

$$\begin{split} \nabla \psi_i \Big|_{\mathcal{R}_i} &= \mathsf{D}_{\mathbf{p}}^T(\psi_i) \Big|_{\mathcal{R}_i} = \mathsf{D}_{\mathbf{p}}^T(\frac{1}{2\rho_{\mathcal{S}}} \big[\rho_{\mathcal{S}} + \rho_i - |\mathbf{p} - \mathbf{p}_i| \big]^2) \\ &= -\frac{\rho_i + \rho_{\mathcal{S}} - |\mathbf{p} - \mathbf{p}_i|}{\rho_{\mathcal{S}} |\mathbf{p} - \mathbf{p}_i|} \big[\mathbf{p} - \mathbf{p}_i \big]. \end{split}$$

By utilizing (3.27), the gradient can be rewritten as,

$$\nabla \psi_i \Big|_{\mathcal{R}_i} = -\frac{\rho_{\mathcal{S}} - |\boldsymbol{\ell}_i(\mathbf{p})|}{\rho_{\mathcal{S}} |\mathbf{p} - \mathbf{p}_i|} [\mathbf{p} - \mathbf{p}_i] = \frac{\rho_{\mathcal{S}} - |\boldsymbol{\ell}_i(\mathbf{p})|}{\rho_{\mathcal{S}} |\boldsymbol{\ell}_i(\mathbf{p})|} \boldsymbol{\ell}_i(\mathbf{p}).$$
(3.30)

This gradient can be computed from available sensory measurements, where its magnitude is simply,

$$\left|\nabla\psi_{i}\right|\Big|_{\mathcal{R}_{i}} = \frac{\rho_{\mathcal{S}} - |\boldsymbol{\ell}_{i}(\mathbf{p})|}{\rho_{\mathcal{S}}},\tag{3.31}$$

which is strictly monotonic and decreases linearly with the obstacle distance. For an unconstrained planar agent, the control law, $\dot{\mathbf{p}} = -\nabla \psi_i$, results in, $\dot{\psi}_i = -|\nabla \psi_i|^2$, implying any trajectory starting in \mathcal{R}_i asymptotically approaches its outer boundary.

The obstacle function Hessian constrained in its region of interest can be derived as,

$$\begin{split} H_{\psi_i} \Big|_{\mathcal{R}_i} &= \mathsf{D}_{\mathbf{p}}^T \left\{ -\frac{\rho_i + \rho_s - |\mathbf{p} - \mathbf{p}_i|}{\rho_s |\mathbf{p} - \mathbf{p}_i|} [\mathbf{p} - \mathbf{p}_i] \right\} \\ &= \mathsf{D}_{\mathbf{p}}^T \left\{ \frac{\rho_i + \rho_s}{\rho_s} \left[\frac{1}{\rho_i + \rho_s} - \frac{1}{|\mathbf{p} - \mathbf{p}_i|} \right] [\mathbf{p} - \mathbf{p}_i] \right\} \\ &= \frac{\rho_i + \rho_s}{\rho_s} \Big[[\frac{1}{\rho_i + \rho_s} - \frac{1}{|\mathbf{p} - \mathbf{p}_i|}] \mathbf{I} + \frac{1}{|\mathbf{p} - \mathbf{p}_i|^3} [\mathbf{p} - \mathbf{p}_i] [\mathbf{p} - \mathbf{p}_i]^T \Big], \end{split}$$

where I is the 2×2 identity matrix. Now let,

$$\mathbf{n}_{\psi_i} := \frac{1}{|\nabla \psi_i|} \nabla \psi_i = \frac{1}{|\mathbf{p} - \mathbf{p}_i|} [\mathbf{p} - \mathbf{p}_i],$$
$$\bar{\mathbf{n}}_{\psi_i} := R(\pi/2) \mathbf{n}_{\psi_i},$$

and observe that $\mathbf{n}_{\psi_i} \mathbf{n}_{\psi_i}^T + \bar{\mathbf{n}}_{\psi_i} \bar{\mathbf{n}}_{\psi_i}^T = \mathbf{I}$. Then the Hessian can be reorganized as,

$$H_{\psi_{i}}\Big|_{\mathcal{R}_{i}} = \frac{\rho_{i} + \rho_{S}}{\rho_{S}} \left[\left[\frac{1}{\rho_{i} + \rho_{S}} - \frac{1}{|\mathbf{p} - \mathbf{p}_{i}|} \right] \mathbf{I} + \frac{1}{|\mathbf{p} - \mathbf{p}_{i}|} \mathbf{n}_{\psi_{i}} \mathbf{n}_{\psi_{i}}^{T} \right] \\ = \frac{\rho_{i} + \rho_{S}}{\rho_{S}} \left[\left[\frac{1}{\rho_{i} + \rho_{S}} - \frac{1}{|\mathbf{p} - \mathbf{p}_{i}|} \right] \mathbf{\bar{n}}_{\psi_{i}} \mathbf{\bar{n}}_{\psi_{i}}^{T} + \frac{1}{\rho_{i} + \rho_{S}} \mathbf{n}_{\psi_{i}} \mathbf{n}_{\psi_{i}}^{T} \right] \\ = \frac{1}{\rho_{S}} \left[\left[1 - \frac{\rho_{i} + \rho_{S}}{|\mathbf{p} - \mathbf{p}_{i}|} \right] \mathbf{\bar{n}}_{\psi_{i}} \mathbf{\bar{n}}_{\psi_{i}}^{T} + \mathbf{n}_{\psi_{i}} \mathbf{n}_{\psi_{i}}^{T} \right], \qquad (3.32)$$

revealing that it has one positive eigenvalue (associated with the gradient eigenvector \mathbf{n}_{ψ_i}) and one negative eigenvalue (associated with the orthogonal eigenvector tangent to the level set of ψ_i).

3.2.1.2 Combined Control Law

As a candidate for expressing the combined hill ascent and obstacle avoidance task, consider the following potential function,

$$\varphi := \phi + \nu_{\psi} \sum_{i=1}^{d} \psi_i, \qquad (3.33)$$

with the positive constant, ν_{ψ} , and the control law based on its gradient,

$$\dot{\mathbf{p}} = -\nabla\varphi(\mathbf{p}) = -\nabla\phi(\mathbf{p}) - \nu_{\psi} \sum_{i=1}^{d} \nabla\psi_i(\mathbf{p}), \qquad (3.34)$$

where ν_{ψ} becomes the *control parameter*. Define the fixed point set as, $\mathcal{C} := \{\mathbf{p} : \nabla \varphi = \mathbf{0}\}$. Then, the resulting Lie derivative, $\dot{\varphi} = -|\nabla \varphi|^2 \leq 0$, is zero if and only if $\mathbf{p} \in \mathcal{C}$. According to our world model, Definition 3.1.2, the obstacle regions on a hill neither intersect with each other nor with the goal. Then, critical point set can be partitioned as,

$$\mathcal{C} = \mathcal{G}_{\phi} \cup \mathcal{S}_{\phi} \cup \mathcal{S}_{\psi}, \tag{3.35}$$

where \mathcal{G}_{ϕ} , \mathcal{S}_{ϕ} are as defined in Assumption 3.1.6, and

$$\mathcal{S}_{\psi} := \bigcup_{i=1}^{d} \mathcal{S}_{\psi_i} := \bigcup_{i=1}^{d} \{ \mathbf{p} \in \mathcal{R}_i : \nabla \phi = -\nu_{\psi} \nabla \psi_i \},$$
(3.36)

represents the set of all spurious critical points emerging from the construction in (3.33). Note that the individual components, S_{ψ_i} , can be empty. In addition, over \mathcal{G}_{ϕ} and \mathcal{S}_{ϕ} , $\varphi = \phi$. This means local Lyapunov and Chetaev functions of Assumption 3.1.6 are maintained for this system. To show that φ is an ML function for the system (3.34) as in Definition 2.2.8, and investigate whether \mathcal{G}_{ϕ} is AGAS (as in Definition 2.1.13), we need to characterize the stability of \mathcal{S}_{ψ} .

Note that an alternative construction for this combined task is the quotient introduced in [79],

$$\varphi_S := \frac{\phi}{\widetilde{\psi}},\tag{3.37}$$

where,

$$\widetilde{\psi} = \prod_{i=1}^{d} \widetilde{\psi}_i := \prod_{i=1}^{d} \left[\frac{\rho_s}{2} - \psi_i \right],$$

vanishes as the state approaches any of the obstacles (for the obstacle \mathcal{O}_i , as $|\boldsymbol{\ell}_i(\mathbf{p})| \to \rho_i$, the obstacle function (3.29), $\psi_i(\mathbf{p}) \to \rho_s/2$), and the quotient based function increase unboundedly. Unfortunately, because we assume that the actual height function, ϕ , is unknown, the gradient of the quotient,

$$\nabla \varphi_S = \frac{1}{\widetilde{\psi}} \left[\nabla \phi - \frac{\phi}{\widetilde{\psi}} \nabla \widetilde{\psi} \right] = \frac{1}{\widetilde{\psi}} \left[\nabla \phi - \frac{\phi}{\widetilde{\psi}} \sum_{i=1}^d \nabla \widetilde{\psi}_i \right], \tag{3.38}$$

can not be recovered by the agent due to the sensory limitations imposed in Section 3.1.3. Even if the task potential function, ϕ , was available as a measurement, $\tilde{\psi}$ is not twice continuously differentiable on \mathcal{P} (for obstacle \mathcal{O}_i , ψ_i is once continuously differentiable at the outer boundary of \mathcal{R}_i), which implies φ_S is not a valid navigation function. Recent work [109] demonstrates that the potential function in (3.37) yields a navigation function in the case of a convex hill with convex obstacles. In the specific case that the hill is quadratic, then a provably correct (continuous and piecewise smooth but non-gradient) vector field has been developed [8] whose expression can be computed directly in real-time from online sensor-based measurements. Since natural hills have various ridges and local maxima that may all be of interest in the hill climbing problem, we prefer not to assume that ϕ is convex (much less quadratic) in this paper. For these reasons, we can not rely on these existing formal methods and must define and develop an understanding of the new controller construction (3.34).

For the obstacle, \mathcal{O}_i , define the following partitioning of the region of interest, \mathcal{R}_i ,

$$\mathcal{U}_{i} := \left\{ \mathbf{p} \in \mathcal{R}_{i} : |\boldsymbol{\ell}_{i}(\mathbf{p})| < \left[1 - \frac{\Gamma_{h}}{\nu_{\psi}}\right] \rho_{\mathcal{S}} \right\},$$
(3.39)

$$\mathcal{V}_{i} := \left\{ \mathbf{p} \in \mathcal{R}_{i} : |\boldsymbol{\ell}_{i}(\mathbf{p})| > \left[1 - \frac{\Omega_{h}}{\nu_{\psi}}\right] \rho_{\mathcal{S}} \right\},$$
(3.40)

$$\mathcal{W}_i := \mathcal{R}_i - \{\mathcal{U}_i \cup \mathcal{V}_i\}, \qquad (3.41)$$

where observe that \mathcal{U}_i is non-empty if and only if $\nu_{\psi} > \Gamma_h$, in which case Figure 3.2 illustrates the region of interest, \mathcal{R}_i , and the partitioning, \mathcal{U}_i , \mathcal{V}_i , and \mathcal{W}_i .

Proposition 3.2.2. For the control law (3.34), the choice of control parameter obeying (3.56), $\nu_{\psi} > \Gamma_h$, where Γ_h is the task gradient magnitude upper bound (3.18), guarantees that the workspace, \mathcal{P} , is positive-invariant.

Proof. To show positive-invariance, we need to show that, under the control law (3.34), the



Figure 3.2: Regions Around the Obstacle. From (3.39), $\rho_{u_i} := \left[1 - \frac{\Gamma_h}{\nu_{\psi}}\right] \rho_{\mathcal{S}}$, and from (3.40), $\rho_{\nu_i} := \frac{\Omega_h}{\nu_{\psi}} \rho_{\mathcal{S}}$.

system can not leave \mathcal{P} through its boundaries, $\partial \mathcal{P}$. Based on (3.14),

$$\partial \mathcal{P} = \partial \mathcal{T} \cup \partial \mathcal{O}, \tag{3.42}$$

where \mathcal{T} is the global terrain component (3.10), and \mathcal{O} is the set of all obstacles (3.3), and Assumption 3.1.4 implies $\partial \mathcal{O} \cap \partial \mathcal{T} = \emptyset$, letting us test these boundary components separately.

Based on Assumption 3.1.4, the obstacle functions, ψ_i , defined in (3.29) are zero over the global terrain component boundary. From (3.33), this implies $\forall \mathbf{p} \in \partial \mathcal{T}, \varphi(\mathbf{p}) = \phi(\mathbf{p})$, where ϕ is the task function defined in (3.15). Then, based on (3.10), $\partial \mathcal{T}$ is a level set of φ . Since under the control law (3.34), $\dot{\mathbf{p}} = -|\nabla \varphi|^2$, we conclude $\forall \mathbf{p}_0 \in \mathcal{P} - \partial \mathcal{T}$ and $\forall t \ge 0$, $\mathbf{p}(t, \mathbf{p}_0) \notin \partial \mathcal{T}$.

For any obstacle, \mathcal{O}_i , the value of corresponding obstacle function, ψ_i , is inversely related to

obstacle distance, $|\ell_i|$, where the level sets are at fixed distances. Any choice of the control parameter obeying (3.56) guarantees that U_i defined in (3.39) is not empty, over which the Lie derivative of the obstacle function is,

$$\begin{split} \dot{\psi_i}\Big|_{\mathcal{U}_i} &= \nabla \psi_i^T \left[-\nabla \phi - \nu_{\psi} \nabla \psi_i \right] = -\nabla \psi_i^T \nabla \phi - \nu_{\psi} \left| \nabla \psi_i \right|^2 \\ &\leq \left| \nabla \psi_i \right| \left| \nabla \phi \right| - \nu_{\psi} \left| \nabla \psi_i \right|^2 = \left| \nabla \psi_i \right| \left[\left| \nabla \phi \right| - \nu_{\psi} \left| \nabla \psi_i \right| \right] \\ &< \left| \nabla \psi_i \right| \left[\Gamma_h - \nu_{\psi} \left| \nabla \psi_i \right| \right]. \end{split}$$

From (3.31),

$$\left|\nabla\psi_{i}\right|_{\mathcal{U}_{i}} > \frac{\rho_{\mathcal{S}} - \left[1 - \frac{\Gamma_{h}}{\nu_{\psi}}\right]\rho_{\mathcal{S}}}{\rho_{\mathcal{S}}} = \frac{\Gamma_{h}}{\nu_{\psi}}.$$
(3.43)

It follows that,

$$\dot{\psi}_i \Big|_{\mathcal{U}_i} < |\nabla \psi_i| \ [\Gamma_h - \Gamma_h] = 0. \tag{3.44}$$

This means, $\forall \mathbf{p} \in \mathcal{U}_i, \frac{d}{dt} \{ |\boldsymbol{\ell}_i(\mathbf{p})| \} > 0$, implying the system can not leave \mathcal{P} through $\partial \mathcal{O}_i$. \Box

Proposition 3.2.3. For the control law in (3.34) combined with the choice of control parameter in (3.56), a necessary condition for any $\mathbf{p}_c \in \mathcal{R}_i$ to be a fixed point, $\mathbf{p}_c \in \mathcal{S}_{\psi_i}$, is, $\mathbf{p}_c \in \mathcal{W}_i$, with \mathcal{W}_i as defined in (3.41).

Proof. For obstacle \mathcal{O}_i , we have already shown in the proof of Proposition 3.2.2 that the choice of the control parameter in (3.56) guarantees that \mathcal{U}_i is non-empty, where $\forall \mathbf{p} \in \mathcal{U}_i$, $\dot{\psi}_i(\mathbf{p}) < 0$. Over \mathcal{V}_i , the Lie derivative of the task function is,

$$\begin{split} \dot{\phi} \Big|_{\mathcal{V}_{i}} &= \nabla \phi^{T} \left[-\nabla \phi - \nu_{\psi} \nabla \psi_{i} \right] = - \left| \nabla \phi \right|^{2} - \nu_{\psi} \nabla \phi^{T} \nabla \psi_{i} \\ &\leq - \left| \nabla \phi \right|^{2} + \nu_{\psi} \left| \nabla \phi \right| \left| \nabla \psi_{i} \right| = \left| \nabla \phi \right| \left[- \left| \nabla \phi \right| + \nu_{\psi} \left| \nabla \psi_{i} \right| \right] \\ &< \left| \nabla \phi \right| \left[-\Omega_{h} + \nu_{\psi} \left| \nabla \psi_{i} \right| \right]. \end{split}$$

where remember from (3.19) that $\forall \mathbf{p} \in \mathcal{R}_i, 0 < \Omega_h \leq |\nabla \phi| (\mathbf{p})$. Based on (3.31),

$$\left|\nabla\psi_{i}\right|_{\mathcal{V}_{i}} < \frac{\rho_{\mathcal{S}} - \left[1 - \frac{\Omega_{h}}{\nu_{\psi}}\right]\rho_{\mathcal{S}}}{\rho_{\mathcal{S}}} = \frac{\Omega_{h}}{\nu_{\psi}},\tag{3.45}$$

resulting in,

$$\dot{\phi}\Big|_{\mathcal{V}_i} < |\nabla\phi| \left[\Omega_h - \Omega_h\right] = 0.$$
(3.46)

Assume $\mathbf{p}_c \in \mathcal{U}_i \cup \mathcal{V}_i$ is indeed a critical point under the control law (3.34), $\dot{\varphi}(\mathbf{p}_c) = -|\nabla \varphi(\mathbf{p}_c)|^2 = 0$, which implies $\dot{\phi}(\mathbf{p}_c) = -\nabla \phi(\mathbf{p}_c)^T \nabla \varphi(\mathbf{p}_c)$ and $\dot{\psi}_i(\mathbf{p}_c) = -\nabla \psi_i(\mathbf{p}_c)^T \nabla \varphi(\mathbf{p}_c) = 0$. This is a contradiction since according to (3.44) $\mathbf{p}_c \in \mathcal{U}_i$ implies $\dot{\psi}_i(\mathbf{p}_c) < 0$, and according to (3.46) $\mathbf{p}_c \in \mathcal{V}_i$ implies $\dot{\phi}(\mathbf{p}_c) < 0$.

Observe that the obstacle distance restricted to \mathcal{R}_i can be represented as,

$$\left|\mathbf{p} - \mathbf{p}_{i}\right|_{\mathcal{R}_{i}} = \rho_{i} + \left|\boldsymbol{\ell}_{i}(\mathbf{p})\right| = \rho_{i} + \alpha_{i} \,\rho_{\mathcal{S}}, \,\alpha_{i} \in (0, \, 1).$$

$$(3.47)$$

When we apply this representation into the negative eigenvalue of H_{ψ_i} in (3.32), we have,

$$\frac{1}{\rho_{\mathcal{S}}} \left[1 - \frac{\rho_i + \rho_{\mathcal{S}}}{\rho_i + \alpha_i \, \rho_{\mathcal{S}}} \right] = -\frac{1}{\rho_{\mathcal{S}}} \frac{\rho_{\mathcal{S}} \left[1 - \alpha_i \right]}{\alpha_i \, \rho_{\mathcal{S}} + \rho_i} = -\frac{1 - \alpha_i}{\rho_i + \alpha_i \, \rho_{\mathcal{S}}}.$$
(3.48)

For $\mathbf{p} \in \mathcal{W}_i$, we can derive from (3.39) and (3.40) that,

$$\left[1 - \frac{\Gamma_h}{\nu_{\psi}}\right] \rho_{\mathcal{S}} \le |\boldsymbol{\ell}_i(\mathbf{p})| \le \left[1 - \frac{\Omega_h}{\nu_{\psi}}\right] \rho_{\mathcal{S}}, \ \mathbf{p} \in \mathcal{W}_i, \tag{3.49}$$

and thus,

$$1 - \frac{\Gamma_h}{\nu_{\psi}} \le \alpha_i \Big|_{\mathcal{W}_i} \le 1 - \frac{\Omega_h}{\nu_{\psi}},\tag{3.50}$$

which is independent of both ρ_{s} and ρ_{i} , letting us conclude that over \mathcal{W}_{i} , the magnitude of

this eigenvalue is inversely related to ρ_s and ρ_i . We proceed with providing more specific upper bounds for ρ_s and the maximum value of ρ_i , ρ_{max} , so that, at a spurious critical point, the magnitude of the negative eigenvalue of $H_{\psi_i}(\mathbf{p}_c)$ is big enough to guarantee that the combined function's Hessian, $H_{\varphi}(\mathbf{p}_c)$, has a negative eigenvalue.

Lemma 3.2.4 (Theorem 4.3.7 of [54]). For any matrix $M \in \mathbb{R}^{m \times m}$, let $\lambda_1(M) \ge \lambda_2(M) \ge$... $\ge \lambda_m(M)$ denote its eigenvalues, and consider two Hermitian matrices $A, B \in \mathbb{R}^{m \times m}$. For indices i, j satisfying $1 \le i + j - 1 \le m$,

$$\lambda_{i+j-1}(A+B) \le \lambda_i(A) + \lambda_j(B), \tag{3.51}$$

and for indices i, j such that $1 \leq i + j - m \leq m$,

$$\lambda_{i+j-m}(A+B) \ge \lambda_i(A) + \lambda_j(B), \tag{3.52}$$

For m = 2, Lemma 3.2.4 results in

$$\lambda_{2}(A) + \lambda_{1}(B) \leq \lambda_{1}(A + B)$$

$$\lambda_{1}(A) + \lambda_{2}(B) \leq \lambda_{1}(A + B)$$

$$\lambda_{1}(A + B) \leq \lambda_{1}(A) + \lambda_{1}(B),$$
(3.53)

and,

$$\lambda_{2}(A) + \lambda_{2}(B) \leq \lambda_{2}(A + B)$$

$$\lambda_{2}(A + B) \leq \lambda_{1}(A) + \lambda_{2}(B)$$

$$\lambda_{2}(A + B) \leq \lambda_{2}(A) + \lambda_{1}(B).$$
(3.54)

Proposition 3.2.5. For the control law (3.34) combined with a control parameter choice based on (3.56), $\nu_{\psi} < \Gamma_h$, assume the maximum permissible obstacle radius, ρ_{max} , obeys (3.55),

$$\rho_{max} < \frac{\Omega_h}{\kappa_\phi}.$$

Any choice of obstacle sensor range, ρ_{s} , satisfying (3.57),

$$\rho_{\mathcal{S}} < \frac{1}{\left[1 - \frac{\Omega_h}{\nu_{\psi}}\right]} \left[\frac{\Omega_h}{\kappa_{\phi}} - \rho_{max}\right],$$

guarantees that, for any critical point described in (3.36), $\mathbf{p}_c \in S_{\psi}$, $H_{\varphi}(\mathbf{p}_c)$ has a negative eigenvalue.

Proof. For obstacle region, \mathcal{O}_i , let $\lambda_1(H_{\phi}) \geq \lambda_2(H_{\phi})$, $\lambda_1(H_{\psi_i}) \geq \lambda_2(H_{\psi_i})$, and $\lambda_1(H_{\varphi}) \geq \lambda_2(H_{\varphi})$ be the pairs of eigenvalues for, H_{ϕ} , H_{ψ_i} , and, $H_{\varphi} = H_{\phi} + \nu_{\psi}H_{\psi_i}$, respectively.

Assume there exists a critical point, $\mathbf{p}_c \in \mathcal{W}_i$. From (3.54), we have,

$$\lambda_2(H_{\varphi}) \le \lambda_1(H_{\phi}) + \nu_{\psi}\lambda_2(H_{\psi_i}).$$

Observe from (3.21), $|\lambda_1(H_{\phi})| \leq \kappa_{\phi}$, and consider the representation for $\lambda_2(H_{\psi_i})$ given in (3.48),

$$\lambda_2(H_{\varphi}) \le \kappa_{\phi} - \nu_{\psi} \frac{1 - \alpha_i}{\rho_i + \alpha_i \, \rho_{\varsigma}}.$$

If we utilize the obstacle radius upper bound (3.8) and the upper bound for α_i in (3.50),

$$\lambda_2(H_{\varphi}) \le \kappa_{\phi} - \nu_{\psi} \frac{1 - \left[1 - \frac{\Omega_h}{\nu_{\psi}}\right]}{\rho_{max} + \left[1 - \frac{\Omega_h}{\nu_{\psi}}\right]\rho_{\mathcal{S}}} = \kappa_{\phi} - \frac{\Omega_h}{\rho_{max} + \left[1 - \frac{\Omega_h}{\nu_{\psi}}\right]\rho_{\mathcal{S}}}$$

Then, a sufficient condition for $\lambda_2(H_{\varphi})$ to be negative is,

$$\kappa_{\phi} < \frac{\Omega_h}{\rho_{max} + \left[1 - \frac{\Omega_h}{\nu_{\psi}}\right]\rho_s},$$

which can be rearranged as

$$\rho_{\mathcal{S}} < \frac{1}{\left[1 - \frac{\Omega_h}{\nu_{\psi}}\right]} \left[\frac{\Omega_h}{\kappa_{\phi}} - \rho_{max}\right].$$

Theorem 3.2.6. Consider an unconstrained planar agent operating on a hill as described in Definition 3.1.2 where the maximum obstacle radius in (3.8) satisfies,

$$\rho_{max} < \frac{\Omega_h}{\kappa_\phi},\tag{3.55}$$

where Ω_h is defined in (3.19) as the lower bound of the task gradient magnitude over all regions of interest, \mathcal{R}_i , and κ_{ϕ} defined in (3.21) as the task curvature bound. Assume the agent is equipped with a vestibular sensor capturing the terrain gradient, ∇h , and a set of obstacle sensors, ℓ_i (3.26). If the choice of control parameter satisfies,

$$\nu_{\psi} > \Gamma_h, \tag{3.56}$$

where Γ_h is the hill gradient magnitude upper bound (3.18), and the selected sensor range satisfies,

$$\rho_{\mathcal{S}} < \frac{1}{\left[1 - \frac{\Omega_h}{\nu_{\psi}}\right]} \left[\frac{\Omega_h}{\kappa_{\phi}} - \rho_{max}\right],\tag{3.57}$$

then, \mathcal{P} is positive invariant, and the control law in (3.34) admits φ as an ML function (as in Definition 2.2.8). Furthermore, under this control law, \mathcal{G}_{ϕ} is AGAS (in the sense of Definition 2.1.13).

Proof. First of all, from Proposition 3.2.2, for the system (3.34), any control parameter choice satisfying (3.56) guarantees that the workspace, \mathcal{P} , is positive invariant. Over \mathcal{G}_{ϕ} and \mathcal{S}_{ϕ} , $\varphi = \phi$. Then, from Assumption 3.1.6, \mathcal{G}_{ϕ}^{j} and \mathcal{S}_{ϕ} admit local Lyapunov and
Chetaev functions, respectively, as required in Definition 2.2.8. Proposition 3.2.5 shows that, given a maximum obstacle radius obeying (3.55), any choice of obstacle sensor range satisfying (3.57) guarantees that the combined function Hessian evaluated at the critical point, $H_{\varphi}(\mathbf{p}_c)$ with $\mathbf{p}_c \in S_{\psi}$, has a negative eigenvalue. From Proposition 2.2.7, this implies that $\mathbf{p}_c \in S_{\psi}$ admit Chetaev functions as required Definition 2.2.8. Then, (3.34) admits φ as an ML function. Since the Jacobian of the system evaluated at the unstable critical point set, $-H_{\varphi}(\mathbf{p}_c)$ with $\mathbf{p}_c \in S_{\phi} \cup S_{\psi}$, has a positive eigenvalue, we conclude from Proposition 2.2.10 that \mathcal{G}_{ϕ} is AGAS.

An important aspect of Theorem 3.2.6 is that the constraint on ρ_s is inversely related to the choice of ν_{ψ} . The implication of this is that on one hand, ν_{ψ} needs to be larger than Γ_h , but on the other hand, a bigger control parameter value implies a tighter bound on a sufficient choice of ρ_s . A special case where this trade-off is irrelevant is when $\nabla \phi$ is approximately constant. This implies the curvature bound, κ_{ϕ} , is approximately zero. Then the second part of the constraint where κ_{ϕ} is in the denominator becomes arbitrarily large and ρ_s can be chosen accordingly independent of the choice of ν_{ψ} .

Example 3.2.7. To illustrate scenarios under which insufficient ρ_{max} and ρ_s values lead into an undesired spurious critical point, consider the following hill task function,

$$\phi = \sqrt{\left[\left[\mathbf{p}_{h} - \mathbf{p}\right]^{T} \mathbf{e}_{1}\right]^{2} + 1} + \sqrt{\left[\left[\mathbf{p}_{h} - \mathbf{p}\right]^{T} \mathbf{e}_{2}\right]^{2} + 1},$$
(3.58)

where $\mathbf{p}_h = [0, 10]^T$ denotes the only critical point of the hill which is stable. This construction is similar to the saturation term in [115], which results in a gradient vector with bounded magnitude, $\Gamma_h = \sqrt{2}$, and a positive-definite Hessian with bounded curvatures, $\kappa_{\phi} = 1/\xi = 1$. Let us introduce a single obstacle located at the origin, $\mathbf{p}_c = \mathbf{0}$, resulting in the minimum task gradient magnitude over the obstacle region, $\Omega_h = 1$. Then, according to



Figure 3.3: Level sets of the combined potential field in Example 3.2.7 for three different sets of choices for ν_{ψ} , ρ_{max} , and ρ_{s} . The inner arc represents the boundary of the obstacle, \mathcal{O}_{1} , and the outer arc represents the boundary of the obstacle region, \mathcal{D}_{1} . For all three cases the choice for $\nu_{\psi} = 2.0$ is sufficiently big. (a) $\rho_{max} = 0.5$, $\rho_{s} = 0.5$ are both sufficiently small and there is a single unstable critical point, (b) $\rho_{max} = 0.5$, $\rho_{s} = 1.5$ where ρ_{s} is too big and there are three critical points one of which is stable, (c) $\rho_{max} = 1.5$, $\rho_{s} = 0.5$, where ρ_{max} is too big and there are three critical points one of which is stable.

Theorem 3.2.6, any spurious critical point emerging under the parameter set satisfying,

$$\rho_{max} < 1 \ (3.55), \quad \nu_{\psi} > \sqrt{2} \ (3.56), \quad \rho_{\mathcal{S}} < \frac{1}{\left[1 - \frac{1}{\nu_{\psi}}\right]} \left[1 - \rho_{max}\right](3.57),$$

is unstable. Figure 3.3 illustrates the levels sets of the combined potential function under three different sets of parameters. For all three cases, $\nu_{\psi} = 2$ is sufficiently big. Then the obstacle radius, $\rho_{max} = 0.5$, results in the sensor radius upper bound, $\rho_{s} < 2 \cdot [1 - 0.5] = 1$. On the other hand, for a more extreme environment in which $\rho_{max} = 1.5$, no ρ_{s} choice can be sufficient as the upper bound for this radius becomes negative.

Corollary 3.2.8. In Theorem 3.2.6, we considered autonomous ascent over a physical hill while avoiding disk obstacles. This same obstacle avoidance approach can be incorporated for any virtual hill satisfying the assumptions laid out in Section 3.1.

3.2.2 Horizontal Unicycle Models

In the previous section, we provide a control strategy for a planar agent with no kinematic constraints that is tasked with autonomous hill ascent, where we show in Theorem 3.2.6 that, trough the sufficient choice of parameters summarized in Table 3.4, φ is an ML function for the system (3.34). Moreover, under these conditions, \mathcal{G}_{ϕ} is AGAS (as in Definition 2.1.13). In this section, we extend this approach into horizontal unicycle models. These models, especially the kinematic unicycle, are popular in the literature since they successfully capture lack of lateral mobility of various mobile platforms including legged platforms like RHex [88, 121]. However, as the locomotion speed increases, the assumption that the translational and rotational velocity values can change instantaneously poses problems with control strategies based on the model, motivating modifications such as the rotational velocity based approach in [128], and the dynamic unicycle model [106, 108]. In this work, for the slow pace operation of the robot we utilize the kinematic unicycle model, whereas for the fast pace operation with the same gait we choose to use the dynamic unicycle model. The horizontal unicycle extensions rely on successive embeddings of the point particle control law while preserving the local stability and instability of the original fixed point set. As discussed in Remark 3.2.12, our current theoretical efforts are not sufficient to show that the almost-global attractiveness of \mathcal{G}_{ϕ} is maintained, and thus, we can not conclude that the embeddings of \mathcal{G}_{ϕ} are AGAS. Yet, we conjecture that this is indeed the case.

3.2.2.1 Kinematic Unicycle

For a summary of additional modeling decisions made for the kinematic unicycle achieving desired stability results, refer to Table 3.5.

	Definition	Expression	
$\mathbf{e}_1,\mathbf{e}_2$	Cartesian unit vectors	$\begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	
$B(\theta)$	kinematic unicycle map	$B(\theta) := \begin{vmatrix} \mathbf{n}(\theta) \mathbf{e}_1^T \\ \mathbf{e}_2^T \end{vmatrix}$	(3.59)
φ_{ku}	kinematic unicycle task function	$arphi_{_{ku}} := u_{_{ heta}} arphi + rac{1}{2} \left ec{ abla} arphi + \left abla arphi \right ^2$	(3.60)
\mathbf{u}_{ku}	kinematic unicycle input	$\mathbf{u}_{_{ku}} := -\sigma_{arphi}(abla arphi) egin{bmatrix} abla arphi & \ & u_{ heta} \end{bmatrix} abla arphi^b$	(3.62)
$\sigma_{\varphi}(\nabla\varphi)$	configurable positive scalar function	L _	
$ u_{ heta}$	control parameter		
	Assumptions about the sensors	and the environment	
$\ell_i{}^b, \nabla \phi^b$	available as measurements		
$ ho_{min}$	known parameter		
	Conditions on the Design Paran	neters	
$ u_{\theta}$	sufficiently large	$\nu_{\theta} > 4 \left[\kappa_{\phi} + \nu_{\psi} \left[\frac{1}{\rho_{min}} + \frac{1}{\rho_{s}} \right] \right]$	(3.69)

KINEMATIC UNICYCLE CONTROL

Table 3.5: Summary of definitions, assumptions, and sufficient conditions for kinematic unicycle control, which is developed on top of the point particle control (Table 3.4), with the difference in available measurements.

Consider the planar kinematic unicycle [88],

$$\dot{\mathbf{q}} = B(\theta)\mathbf{u}_{ku}; \quad B(\theta) := \begin{bmatrix} \mathbf{n}(\theta)\mathbf{e}_1^T \\ \mathbf{e}_2^T \end{bmatrix},$$
(3.59)

with $\mathbf{I} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, where $\mathbf{u}_{ku} \in \mathbb{R}^2$ consists of translational and rotational velocity inputs, respectively.

One way to have a kinematic unicycle execute the task of autonomous hill ascent while avoiding obstacles is to build upon the gradient of the combined ML function, $\nabla\varphi$, by introducing an *angle error* term, φ_{θ} , between the gradient vector angle, $\angle(\nabla\varphi)$, and the robot's heading, θ . A widely used approach for error tracking on SO(2) (and SO(3) as in [74]), $\varphi_{\theta} := 1 + \cos(\theta - \angle(\nabla\varphi)) = 1 + \frac{1}{|\nabla\varphi|}\nabla\varphi^{T}\mathbf{n}(\theta)$, is not smooth when $\nabla\varphi = 0$. To address this, we propose a modified angle error energy term, $\tilde{\varphi}_{\theta} := |\nabla\varphi|^{2}\varphi_{\theta} = \nabla\varphi^{T}\nabla\varphi + |\nabla\varphi|\nabla\varphi^{T}\mathbf{n} = \frac{1}{2}|\nabla\varphi + |\nabla\varphi|\mathbf{n}|^{2}$, and the function,

$$\varphi_{ku} := \nu_{\theta} \varphi + \frac{1}{2} \left| \nabla \varphi + \left| \nabla \varphi \right| \mathbf{n} \right|^2, \qquad (3.60)$$

where $\varphi_{ku}: SE(2) \to \mathbb{R}_{\geq 0}$ and $\nu_{\theta} > 0$ constant. The gradient of this function is,

$$\nabla \varphi_{ku} = \mathsf{D}_{\mathbf{q}}^{T} (\nu_{\theta} \varphi + \frac{1}{2} |\nabla \varphi + |\nabla \varphi| \, \mathbf{n}|^{2}),$$

with the components,

-

$$\begin{split} \mathsf{D}_{\mathbf{p}}^{T}(\frac{1}{2} \left| \nabla \varphi + \left| \nabla \varphi \right| \mathbf{n} \right|^{2}) &= \mathsf{D}_{\mathbf{p}}^{T}(\nabla \varphi + \left| \nabla \varphi \right| \mathbf{n}) \left[\nabla \varphi + \left| \nabla \varphi \right| \mathbf{n} \right] \\ &= \left[H_{\varphi} + \frac{1}{2 \left| \nabla \varphi \right|} H_{\varphi} \nabla \varphi \mathbf{n}^{T} \right] \left[\nabla \varphi + \left| \nabla \varphi \right| \mathbf{n} \right], \end{split}$$

and,

$$\begin{split} \frac{\partial}{\partial \theta} (\frac{1}{2} \left| \nabla \varphi + \left| \nabla \varphi \right| \mathbf{n} \right|^2) &= \frac{\partial}{\partial \theta} (\nabla \varphi + \left| \nabla \varphi \right| \mathbf{n})^T \left[\nabla \varphi + \left| \nabla \varphi \right| \mathbf{n} \right] \\ &= \left| \nabla \varphi \right| \bar{\mathbf{n}}^T \left[\nabla \varphi + \left| \nabla \varphi \right| \mathbf{n} \right]. \end{split}$$

Then,

$$\nabla \varphi_{ku} = \begin{bmatrix} \nu_{\theta} \nabla \varphi \\ 0 \end{bmatrix} + \begin{bmatrix} [H_{\varphi} + \frac{1}{|\nabla \varphi|} H_{\varphi} \nabla \varphi \, \mathbf{n}^{T}] [\nabla \varphi + |\nabla \varphi| \, \mathbf{n}] \\ |\nabla \varphi| \, \bar{\mathbf{n}}^{T} [\nabla \varphi + |\nabla \varphi| \, \mathbf{n}] \end{bmatrix}$$
$$= \begin{bmatrix} \nu_{\theta} \nabla \varphi \\ 0 \end{bmatrix} + \begin{bmatrix} [2 + \frac{\mathbf{n}^{T} \nabla \varphi}{|\nabla \varphi|}] H_{\varphi} \nabla \varphi + |\nabla \varphi| H_{\varphi} \mathbf{n} \\ |\nabla \varphi| \, \bar{\mathbf{n}}^{T} \nabla \varphi \end{bmatrix}$$
$$= \begin{bmatrix} \nu_{\theta} \mathbf{I} \\ |\nabla \varphi| \, \bar{\mathbf{n}}^{T} \end{bmatrix} \nabla \varphi + \begin{bmatrix} [1 + \frac{|\nabla \varphi + |\nabla \varphi| |\mathbf{n}|^{2}}{2|\nabla \varphi|^{2}}] H_{\varphi} \nabla \varphi + |\nabla \varphi| H_{\varphi} \mathbf{n} \\ 0 \end{bmatrix}.$$
(3.61)

Observe that the second term in the gradient contains the Hessian, H_{φ} , which is not available as a sensory measurement, preventing us from directly utilizing this gradient for a control policy as in [88]. On the other hand, the following control policy,

$$\mathbf{u}_{ku} := -\sigma_{\varphi}(|\nabla\varphi|)B(\theta)^{T} \begin{bmatrix} |\nabla\varphi| \mathbf{I} \\ \nu_{\theta}\bar{\mathbf{n}}^{T} \end{bmatrix} \nabla\varphi$$
$$= -\sigma_{\varphi}(|\nabla\varphi|) \begin{bmatrix} |\nabla\varphi| \\ \nu_{\theta} \end{bmatrix} \nabla\varphi^{b}, \qquad (3.62)$$

where σ_{φ} is a smooth and positive scalar valued function of the gradient magnitude (representing any scaling or saturation that could be introduced to the control law to respect limitations on applicable inputs), can be recovered from available measurements. Moreover, under this control law, the resulting system,

$$\dot{\mathbf{q}} = -\sigma_{\varphi}(|\nabla\varphi|) \begin{bmatrix} \mathbf{n}(\theta)\mathbf{e}_{1}^{T} \\ \mathbf{e}_{2}^{T} \end{bmatrix} \begin{bmatrix} |\nabla\varphi| \\ \nu_{\theta} \end{bmatrix} R^{T}(\theta)\nabla\varphi$$
$$= -\sigma_{\varphi}(|\nabla\varphi|) \begin{bmatrix} |\nabla\varphi| [\mathbf{n}^{T}(\theta)\nabla\varphi]\mathbf{n}(\theta) \\ \nu_{\theta}\bar{\mathbf{n}}^{T}(\theta)\nabla\varphi \end{bmatrix}, \qquad (3.63)$$

has the fixed point set, $\mathcal{C} \times \mathbb{S}^1$.

Proposition 3.2.9. Consider the planar kinematic unicycle agent given in (3.59). For

the control policy given in (3.62), the choice of parameter, (3.69), guarantees that for the resulting system (3.63), the function, (3.60), has a non-positive Lie derivative, that is zero if and only if $\mathbf{p} \in \mathcal{C}$.

Proof. For the system (3.63) the function, (3.60), has the Lie derivative,

$$\dot{\varphi}_{ku} = \nabla \varphi_{ku}^{T} B(\theta) \mathbf{u}_{ku} = -\sigma_{\varphi} \nabla \varphi_{ku}^{T} \begin{bmatrix} |\nabla \varphi| \mathbf{n} \mathbf{n}^{T} \\ \nu_{\theta} \bar{\mathbf{n}}^{T} \end{bmatrix} \nabla \varphi,$$

where, substituting for $\nabla \varphi_{ku}$ from (3.61) and noting that $\mathbf{nn}^T + \bar{\mathbf{n}}\bar{\mathbf{n}}^T = \mathbf{I}$, results in,

$$\dot{\varphi}_{ku} = -\sigma_{\varphi} \left| \nabla \varphi \right| \nu_{\theta} \nabla \varphi^{T} \nabla \varphi - \sigma_{\varphi} \left| \nabla \varphi \right| \left[\left[1 + \frac{\left| \nabla \varphi + \left| \nabla \varphi \right| \mathbf{n} \right|^{2}}{2 \left| \nabla \varphi \right|^{2}} \right] \nabla \varphi + \left| \nabla \varphi \right| \mathbf{n} \right]^{T} H_{\varphi} \mathbf{n} \mathbf{n}^{T} \nabla \varphi.$$

By utilizing Cauchy-Schwarz¹³ and matrix norm¹⁴ inequalities [135],

$$\begin{split} \dot{\varphi}_{ku} &\leq -\sigma_{\varphi}\nu_{\theta} \left|\nabla\varphi\right|^{3} + \sigma_{\varphi} \left[\left[1 + \frac{4\left|\nabla\varphi\right|^{2}}{2\left|\nabla\varphi\right|^{2}}\right] \left|\nabla\varphi\right| + \left|\nabla\varphi\right| \right] \kappa_{\varphi} \left|\nabla\varphi\right|^{2} \\ &\leq -\sigma_{\varphi}\nu_{\theta} \left|\nabla\varphi\right|^{3} + \sigma_{\varphi} \left[3\left|\nabla\varphi\right| + \left|\nabla\varphi\right|\right] \kappa_{\varphi} \left|\nabla\varphi\right|^{2} \\ &\leq -\sigma_{\varphi} \left[\nu_{\theta} - 4\kappa_{\varphi}\right] \left|\nabla\varphi\right|^{3}, \end{split}$$

and from (3.64),

$$\nu_{\theta} > 4 \big[\kappa_{\phi} + \nu_{\psi} \big[\frac{1}{\rho_{min}} + \frac{1}{\rho_{\mathcal{S}}} \big] \big],$$

guarantees that $\dot{\varphi}_{_{ku}} \leq 0$ and $\dot{\varphi}_{_{ku}} = 0$ if and only if $\mathbf{p} \in \mathcal{C}$.

Lemma 3.2.10. For the point particle control parameter choice (3.56), and obstacle sensor range choice (3.56), the maximum curvature for the combined task function, φ , in (3.33),

¹³Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$. Then, $|\mathbf{x}^T \mathbf{y}| \le |\mathbf{x}| |\mathbf{y}|$. ¹⁴Let $\mathbf{x} \in \mathbb{R}^m$ and $A \in \mathbb{R}^k \times \mathbb{R}^m$. Then, $|A\mathbf{x}| \le ||A|| |\mathbf{x}|$.

 $\kappa_{\varphi} := \sup_{\mathbf{p} \in \mathcal{P}} \|H_{\varphi}(\mathbf{p})\|, \text{ is bounded by,}$

$$\kappa_{\varphi} \le \kappa_{\phi} + \nu_{\psi} \left[\frac{1}{\rho_{min}} + \frac{1}{\rho_{\varsigma}} \right]. \tag{3.64}$$

Proof. When $\mathbf{p} \in \mathcal{P} - \bigcup_{i=1}^{n} \mathcal{R}_{i}$, the combined task function, φ , shares the same curvature bound as the task function, ϕ , $||H_{\varphi}(\mathbf{p})|| \leq \kappa_{\phi}$, $\forall \mathbf{p} \in \mathcal{P} - \bigcup_{i=1}^{n} \mathcal{R}_{i}$. Thus, in this proof we will be investigating the curvature bounds over the regions of interest, \mathcal{R}_{i} . As in Proposition 3.2.5, let $\lambda_{1}(H_{\phi}) \geq \lambda_{2}(H_{\phi})$, $\lambda_{1}(H_{\psi_{i}}) \geq \lambda_{2}(H_{\psi_{i}})$, and $\lambda_{1}(H_{\varphi}) \geq \lambda_{2}(H_{\varphi})$ be the pairs of eigenvalues for, H_{ϕ} , $H_{\psi_{i}}$, and, $H_{\varphi} = H_{\phi} + \nu_{\psi}H_{\psi_{i}}$, respectively.

Note that,

$$\|H_{\varphi}\| = \max\left\{ |\lambda_1(H_{\varphi})|, |\lambda_2(H_{\varphi})| \right\}.$$
(3.65)

From (3.54), we have $\lambda_2(H_{\varphi}) \geq \lambda_2(H_{\phi}) + \nu_{\psi}\lambda_2(H_{\psi_i})$, where recall from Proposition 3.2.5 that $\lambda_2(H_{\varphi})$ is negative. In addition, observe from (3.21) that $|\lambda_2(H_{\phi})| \leq \kappa_{\phi}$, and consider the representation for $\lambda_2(H_{\psi_i})$ given in (3.48). Then,

$$\begin{split} |\lambda_{2}(H_{\varphi})| &\leq |\lambda_{2}(H_{\phi}) + \nu_{\psi}\lambda_{2}(H_{\psi_{i}})| \\ &\leq |\lambda_{2}(H_{\phi})| + \nu_{\psi} \left|\lambda_{2}(H_{\psi_{i}})\right| \\ &\leq \kappa_{\phi} + \nu_{\psi}\frac{1 - \alpha_{i}}{\alpha_{i} \rho_{S} + \rho_{i}}, \end{split}$$

where, by combining (3.8) with the fact that $\alpha_i > 0$, we reach,

$$|\lambda_2(H_{\varphi})| < \kappa_{\phi} + \nu_{\psi} \frac{1}{\rho_{min}}.$$
(3.66)

Similarly, via (3.53), we have, $\lambda_1(H_{\varphi}) \leq \lambda_1(H_{\phi}) + \nu_{\psi}\lambda_1(H_{\psi_i})$. Notice that $\lambda_1(H_{\varphi}) < 0$ implies $|\lambda_1(H_{\varphi})| \leq |\lambda_2(H_{\varphi})|$, thus we focus on when $\lambda_1(H_{\varphi}) > 0$. Once again, from (3.21) we have $|\lambda_1(H_{\phi})| \leq \kappa_{\phi}$ and $\lambda_1(H_{\psi_i})$ is provided in (3.32), resulting in,

$$\begin{aligned} |\lambda_1(H_{\varphi})| &\leq |\lambda_1(H_{\phi}) + \nu_{\psi}\lambda_1(H_{\psi_i})| \\ &\leq |\lambda_1(H_{\phi})| + \nu_{\psi} |\lambda_1(H_{\psi_i})| \\ &\leq \kappa_{\phi} + \nu_{\psi} \frac{1}{\rho_s}. \end{aligned}$$
(3.67)

Then,

$$\|H_{\varphi}\| \leq \max\left\{\kappa_{\phi} + \nu_{\psi}\frac{1}{\rho_{min}}, \kappa_{\phi} + \nu_{\psi}\frac{1}{\rho_{s}}\right\}$$
$$\leq \kappa_{\phi} + \nu_{\psi}\max\left\{\frac{1}{\rho_{min}}, \frac{1}{\rho_{s}}\right\}$$
$$\leq \kappa_{\phi} + \nu_{\psi}\left[\frac{1}{\rho_{min}} + \frac{1}{\rho_{s}}\right].$$
(3.68)

The main constraint for the planar kinematic unicycle is the lack of lateral mobility. Thus, it is intuitively clear that the success of any gradient tracking strategy depends on curvatures of corresponding potential function. With the following theorem, we provide a sufficient condition for the control parameter, ν_{θ} , based on curvature bounds of the ML function, φ , guaranteeing that under the control law (3.62), the function (3.60) is an ML function.

Theorem 3.2.11. Consider the point particle system given in (3.34), and assume that, for this system, there exists an ML function φ over the positive invariant set \mathcal{P} , where for all unstable critical points, $\mathbf{p}_c \in \mathcal{S}$, its Hessian, $H_{\varphi}(\mathbf{p}_c)$, has a negative eigenvalue, and its curvatures are bounded, $\sup_{\mathbf{p}\in\mathcal{P}} \{ \|H_{\varphi}(\mathbf{p})\| \} \leq \kappa_{\varphi} < \infty$. For the planar kinematic unicycle agent (3.59), under the control policy given in (3.62), if the choice of control parameter satisfies,

$$\nu_{\theta} > 4\kappa_{\varphi},\tag{3.69}$$

then the resulting system (3.63) admits φ_{ku} as an ML function.

Proof. To achieve the desired result, we need to show that the requirements for Theorem 2.3.1 hold. The assumptions regarding φ already follow its description in Theorem 2.3.1. In addition, we show in Proposition 3.2.9 that if the choice of ν_{θ} satisfies (3.69), then it follows that the Lie derivative of φ_{ku} is non-positive, and zero if and only if $\mathbf{p} \in C$. Lastly, the continuously differentiable nonnegative function, $\eta(\mathbf{p}, \theta) := \frac{1}{2} |\nabla \varphi(\mathbf{p}) + |\nabla \varphi(\mathbf{p})| \mathbf{n}(\theta)|^2$ satisfies that $\eta(\mathbf{p}, \theta) \leq 2 \nabla \varphi^T(\mathbf{p}) \nabla \varphi(\mathbf{p})$. We conclude from Theorem 2.3.1 that φ_{ku} is an ML function.

Remark 3.2.12. In Theorem 3.2.11, we could not conclude that $\mathcal{G} \times \mathbb{S}^1$ is AGAS. This is because all the eigenvalues of the Jacobian of (3.63) over any critical point are zero, hence, the Center Manifold Theorem (Thm. 3.2.1 of [48]) is not applicable to show that the stable manifold around unstable equilibria have empty interior. Yet, we conjecture that this is indeed the case, and $\mathcal{G} \times \mathbb{S}^1$ is AGAS.

Remark 3.2.13. Theorem 3.2.11 does not examine the positive invariance of the original set, \mathcal{P} . This is because, in general, we can not make positive invariance claims for the embedded system, as discussed in [78]. Instead, around every connected component of its boundary, $\partial \mathcal{P}$, [78] shows the positive invariance of the lowest boundary energy set of φ_{du} . In principal, one can exploit this observation in the present setting to define a danger zone that circumscribes each physical obstacle within the larger (more conservative) region cut out by the level set of its lowest energy boundary point ¹⁵. Since this is indeed positive invariant, knowing the robot's ascent initiates outside these zones will guarantee that it avoid the obstacle for all future time. In implementation, however, determining such danger zones would require advance knowledge of the obstacle's position—quite at odds with the intended application setting. In practice, in none of the hundreds of empirical trials to be presented in

¹⁵We conjecture that, by defining a ring shaped danger zone around the boundary of obstacle, \mathcal{O}_i , we can guarantee that, given that the robot starts out of the danger zone, it will not cross over this obstacle boundary. We further conjecture that this zone's radius depends on the kinematic unicycle control parameter choice, ν_{θ} . A bigger ν_{θ} value results in a smaller danger zone radius.

Section 4.2.2 has the robot ever penetrated inside such a danger zone to hit an obstacle—an effective safety property for which we offer intuitive explanation in the accompanying text.

Let us now consider the task of autonomous hill ascent's suitability for Theorem 3.2.11. For the point particle agent, when the conditions for Theorem 3.2.6 are satisfied, the system (3.34) admits φ as an ML function over the positive invariant set \mathcal{P} , where, $\forall \mathbf{p}_c \in \mathcal{S}_{\phi} \cup \mathcal{S}_{\psi}$, $H_{\varphi}(\mathbf{p}_c)$ has a negative eigenvalue. Furthermore, in Lemma 3.2.10, we provide a specific finite upper bound over κ_{φ} . By utilizing this bound we further conclude that, for the task of autonomous hill ascent, the tighter bound,

$$\nu_{\theta} > 4 \left[\kappa_{\phi} + \nu_{\psi} \left[\frac{1}{\rho_{min}} + \frac{1}{\rho_{s}} \right] \right], \qquad (3.70)$$

guarantees that Theorem 3.2.11 holds.

As discussed in Remark 3.2.13, resulting kinematic unicycle controller does not guarantee positive invariance of \mathcal{P} , but safety is maintained in working practice for reasons we now intuitively describe. Given that the robot starts in close vicinity of $\partial \mathcal{O}_i$, a penetration into an actual physical object is very unlikely as it depends on the robot's initial heading instantaneously zeroing out the obstacle gradient's translational component right as the hill ascent gradient is driving the robot into the obstacle. Meanwhile, the aggregate rotational component is still steering the robot away from the obstacle. As mentioned in the Remark, an appropriately defined danger zone formally eliminates even this unlikely event, but requires advance information about the obstacle that we cannot assume in the present application setting.

Remark 3.2.14 (Parameter Availability). In the actual implementation, as described in Section 4.1.4, Γ_h , Ω_h , κ_{ϕ} , ρ_{min} , and ρ_{max} are not known, and we do not attempt to estimate them. Instead, the controller performance requires proper tuning of control and sensor parameters.

3.2.2.2 Dynamic Unicycle

A summary of additional modeling decisions made for the dynamic unicycle agent to extend the kinematic unicycle stability results can be found in Table 3.6.

	Definition	Expression	
r	velocity state vector	$\mathbf{r} \in \mathbb{R}^2$	
\mathbf{u}_{du}	dynamic unicycle input	$\mathbf{u}_{_{du}} := \dot{\mathbf{u}}_{_{ku}} - u_{\mathbf{r}} ig[\mathbf{r} - \mathbf{u}_{_{ku}} ig] \in \mathbb{R}^2$	(3.72)
$\nu_{\mathbf{r}}$	control parameter	$\nu_{\mathbf{r}} > 0$	
	Assumption		
$\dot{\mathbf{u}}_{ku}$	kinematic unicycle input derivative available		

DYNAMIC UNICYCLE CONTROL

Table 3.6: Summary of additional definitions and assumptions for dynamic unicycle control, which is based upon the kinematic unicycle control (Table 3.5).

Consider the dynamic unicycle [106, 108], a second order system of the robot pose, \mathbf{q} ,

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{r}} \end{bmatrix} = \begin{bmatrix} B(\theta)\mathbf{r} \\ \mathbf{u}_{du} \end{bmatrix}, \qquad (3.71)$$

with $B(\theta)$ as in (3.59). The velocity input vector of the kinematic unicycle, \mathbf{u}_{ku} in (3.59), is replaced by a velocity state vector, $\mathbf{r} \in \mathbb{R}^2$, whose evolution is controlled via an acceleration input vector, $\mathbf{u}_{du} \in \mathbb{R}^2$.

Consider the control policy which is based on a *second order embedding* (for more information, consult Section 2.4) of the kinematic unicycle input, \mathbf{u}_{ku} (3.62),

$$\mathbf{u}_{du} := \dot{\mathbf{u}}_{ku} - \nu_{\mathbf{r}} [\mathbf{r} - \mathbf{u}_{ku}], \qquad (3.72)$$

with the dynamic unicycle control parameter, $\nu_{\mathbf{r}} > 0$, constant. The fixed point set of the resulting system is of the form, $\mathcal{C} \times \mathbb{S}^1 \times \{\mathbf{0}\}$. Observe that this policy includes the time derivative of the kinematic unicycle input, which is a function of the combined ML function gradient, $\nabla \varphi$. However the Hessian of this function, H_{φ} , is not available. We discuss the details regarding the numerical approximation of this derivative in Section 4.1.

Theorem 3.2.15. For the dynamic unicycle given in (3.71), consider the control policy given in (3.62) & (3.72), with the choice of parameter (3.69), and assume that the requirements for Theorem 3.2.11 are satisfied. The resulting system admits,

$$\varphi_{du} := \nu_{\varphi} \varphi_{ku} + \frac{2}{3} \left[\mathbf{r} - \mathbf{u}_{ku} \right]^{3/2}, \qquad (3.73)$$

as an ML function.

Proof. Our main goal in this proof is to utilize our results for second order embedding of a system admitting an ML function, presented in Theorem 2.4.1. Observe that the candidate function follows the ML function construction, (2.12), with $\beta = 3$. Thus, we need to show that (3.71) combined with (3.72) satisfies the requirements for Theorem 2.4.1.

Recall from Theorem 3.2.11 that the kinematic unicycle input, \mathbf{u}_{ku} , given in (3.62), with a sufficiently large choice of ν_{θ} , results in the system (3.63) admitting φ_{ku} as an ML function. Then, the only condition of Theorem 2.4.1 we need to investigate is concerning the Lie derivative bound of the ML function, (2.9). If we take the norm of $\nabla \varphi_{ku}$ in (3.61), we see that,

$$\begin{split} |\nabla\varphi_{ku}| &\leq \left| \begin{array}{c} \nu_{\theta}\mathbf{I} \\ |\nabla\varphi|\,\bar{\mathbf{n}}^{T} \end{array} \right| \cdot |\nabla\varphi| + \left| \left[1 + \frac{|\nabla\varphi + |\nabla\varphi|\,\mathbf{n}|^{2}}{2\,|\nabla\varphi|^{2}} \right] H_{\varphi}\nabla\varphi \right| + |\nabla\varphi| \cdot |H_{\varphi}\mathbf{n}| \\ &\leq \nu_{\theta}\,|\nabla\varphi| + |\nabla\varphi|^{2} + 3\kappa_{\varphi}\,|\nabla\varphi| + \kappa_{\varphi}\,|\nabla\varphi| \\ &\leq \left[\nu_{\theta} + \Gamma_{\varphi} + 4\kappa_{\varphi} \right] \cdot |\nabla\varphi| \,, \end{split}$$

where observe that, $\Gamma_{\varphi} := \max_{\mathbf{p}\in\mathcal{P}} |\nabla\varphi(\mathbf{p})|$, is bounded. We show in the proof of Proposition 3.2.9 that, for sufficiently large ν_{θ} , $\nabla\varphi_{ku}^{T}B(\theta)\mathbf{u}_{ku} \leq -[\nu_{\theta} - 4\kappa_{\varphi}] \cdot |\nabla\varphi|^{3}$ where it is

zero if and only if $\nabla \varphi_{ku} = \mathbf{0}.$ This implies that,

$$\nabla \varphi_{ku}^{T} B(\theta) \mathbf{u}_{ku} \leq -\frac{\nu_{\theta} - 4\kappa_{\varphi}}{\left[\nu_{\theta} + \Gamma_{\varphi} + 4\kappa_{\varphi}\right]^{3}} \cdot |\nabla \varphi_{ku}|^{3},$$

and thus, from Theorem 2.4.1, (3.71) combined with (3.72) admits (3.73) as an ML function.

Remark 3.2.16. As in Theorem 3.2.11, we can not conclude that $\mathcal{G}_{\phi} \times \mathbb{S}^1 \times \{\mathbf{0}\}$ is AGAS. This is due to reasons underlined in Remark 3.2.12, and we similarly conjecture that this is indeed the case.

Remark 3.2.17. Similar to Theorem 3.2.11, Theorem 3.2.15 does not examine the positive invariance of the original set, \mathcal{P} . Following Remark 3.2.13, we can define danger zones around obstacles and, starting from [78], show that, for any initial condition outside these non-safe zones, the robot will never cross over the obstacle boundary.

In the case of the hill ascent control law (3.34), we conjecture that, the size of these danger zones depend upon the initial velocity of the robot in combination with the ring-shaped danger zones defined for its kinematic unicycle *template* in (3.59).

Part II

Autonomous Behaviors with a Legged Robot

Chapter 4

Autonomous Hill Ascent

In this chapter, we report on the implementation of the Autonomous Hill Ascent behavior presented in [57], an application of task level autonomy wherein a legged robot achieves unassisted ascent of outdoor forested terrain in a variety of challenging settings (Figure 1.1). To support different scenarios, we have implemented a slow pace (up to 0.7m/sec) and a fast pace (up to 1.5m/sec) horizontal unicycle control law. This work, (in concert with the initial implementations reported in [62]) offers the first documented account of completely autonomous ascent over naturally populated hillsides by a robotic platform at speeds comparable to human uphill hiking and flat surface walking¹⁶. The climbing algorithms have useful provable properties with respect to a greatly simplified world model that abstracts away details of terrain (negotiated by the mechanical stability properties of the vehicle) and obstacle shape (irrelevant at the relatively coarse scale afforded by the obstacles' presumptive sufficiently low density). Despite the model's dramatically simplified assumptions, it approximates the reality of forested ascent sufficiently well that we have logged thousands of body lengths of successful, entirely unassisted robot climbs in natural unstructured woodland and parkland settings.

 $^{^{16}\}text{Based}$ on an uphill hiking speed for a 10° hill of 0.56m/sec [83] and a walking speed on flat terrain of 1.46m/sec [72].

In Section 4.1.1, we go through implementation details for the autonomous hill ascent behavior on the RHex robotic platform [42, 121], and the challenges faced to achieve reliable performance, where Figure 4.1 and Figure 4.2 illustrate kinematic and dynamic unicycle control law implementations, respectively. We document the results on extensive experimentation with RHex in Section 4.2. These results include performance experiments both at walking and running speeds and another set of experiments conducted to compare the dynamic and kinematic unicycle control laws based on specific resistance.

4.1 Implementation

Both kinematic unicycle and dynamic unicycle control law implementations start with processing the two physical sensors, LIDAR and IMU, to generate the sensory inputs (hill gradient, hill incline, and obstacle) expected by the *Task Encoder*. The *Task Encoder* is responsible for combining these inputs into a task gradient, in addition to filtering these inputs for successful execution. The output of the *Task Encoder* is applied to the *Kinematic Unicycle Control* module. In the case of the kinematic unicycle implementation, the resulting input vector, \mathbf{u}_{ku} introduced in (3.62), is then directly fed to the robot as a velocity input. For the dynamic unicycle implementation, this output and its derivative are fed into the *Internal System* representing the dynamic unicycle extension. The *Internal System* state is then utilized as the velocity input to the robot.

4.1.1 Sensors

In this section we provide a list of sensors used for implementing autonomous hill ascent. The first of these is an exteroceptive sensor that can be realized through use of a LIDAR hardware unit mounted on a robot, whereas the other two are vestibular sensors that rely on a conventional IMU.



Figure 4.1: Kinematic unicycle implementation. Measurements from the two physical sensors, IMU and LIDAR, are processed to provide the sensory inputs expected by the task encoder module. In return, this module provides the combined task gradient for the kinematic unicycle control law, which is fed into the robot locomotion module as the velocity input.

4.1.1.1 Obstacle Sensor

The obstacle sensor is an abstract map,

$$\sigma_d: SE(2) \to \underbrace{\mathbb{R}^2 \times \ldots \times \mathbb{R}^2}_{d \ copies},$$

over the hill, h, as defined in Definition 3.1.2. From each position and heading on the plane, $\mathbf{q} \in SE(2)$, this map returns a set of vectors, $\ell_i^{\ b}$, in the body frame based on the d obstacles located over \mathcal{P} as described in Section 3.1.1. Given the sensor range, ρ_s , and the field of view limit, β_M , denoting visibility constraints for distance and relative bearing, respectively, this sensor first performs a radial quantization of the visible portion where each slice has the arc length of k, and for each of these slices, it returns the distance to the nearest excessive grade. Then, it clusters these readings into candidate obstacles. Candidates satisfying a minimum arc length threshold are registered as obstacles, \mathcal{O}_i , represented by a local frame vector pointing towards their closest member, $\ell_i^{\ b}$.

In our implementation, we use the output from a fixed LIDAR unit, $\{\mathbf{d}_i^{\ b}\}$, placed horizontally on the body frame. [127] previously discussed some of the limitations of placing the



Figure 4.2: Dynamic unicycle implementation. The diagram follows Figure 4.1, except the kinematic unicycle input and its derivative is utilized by the internal system representing dynamic unicycle extension, instead. This system's state is applied to the robot locomotion module as the velocity input.

LIDAR unit with no pitch down angle, including inability in detecting obstacles that are lower than the height of the beam and any problems variation in body pitch may cause. In our case, the laser scanner plane is at a height such that any obstacle that it cannot see is assumed to be surmountable and any obstacle that it can see is assumed to be insurmountable with respect to the standard alternating tripod gait. For the chosen fixed placement of this unit, our robot interprets as an obstacle anything (tree, rock, slope increase, wall) that rises more than 25cm over a run set by ρ_s above the existing slope—hence, abstractly, this sensor is indeed responding to an excessively steep grade corresponding to the terrain model above. The LIDAR unit cannot sense beyond a distance of 4m, to which the *infinite* reading of its maximum depth scale is calibrated (i.e. $\rho_{s} < 4.0$). The field of view extends roughly $\pm 120^{\circ}$ off center (i.e. $\beta_M \leq 120^{\circ}$), and it is divided into 682 slices (i.e. $k = 240^{\circ}/682$). The way we process the LIDAR output is somewhat similar to [40], with some differences such as LIDAR unit placement, and availability of the robot's pose relative to any reference. In addition, unlike [40], where both lateral discontinuities along the measurement plane and longitudinal discontinuities along the direction of motion are taken into account, we only rely on lateral discontinuities in the clustering. Instead, a short term memory is realized at task encoding stage (Section 4.1.2).

4.1.1.2 Hill Gradient Sensor

Similar to [134], given the robot's frame of reference $(\mathbf{x}^b, \mathbf{y}^b, \mathbf{z}^b)$ from an IMU, the local instantaneous hill gradient, ∇h^b , can be computed through the direction of the normalized gravity vector, $\mathbf{n}_{\mathbf{g}}$,

$$\nabla h^b := R(\theta)^T \nabla h = - \begin{bmatrix} \mathbf{n}_{\mathbf{g}}^T \mathbf{x}^b \\ \mathbf{n}_{\mathbf{g}}^T \mathbf{y}^b \end{bmatrix}.$$
(4.1)

4.1.1.3 Hill Incline Sensor

Given the robot's frame of reference $(\mathbf{x}^b, \mathbf{y}^b, \mathbf{z}^b)$ from an IMU and the steepest ascent gradient magnitude $|\nabla h^b|$, the tangent of the instantaneous hill incline angle, α , can be computed through,

$$\tan \alpha = -\frac{\left|\nabla h^b\right|}{\mathbf{n}_{\mathbf{g}}^T \mathbf{z}^b}.$$
(4.2)

4.1.2 Task Encoder

The task encoder module is responsible for constructing the combined task gradient in the body frame. To do this, it first computes two components: the obstacle function gradient, $\nabla \psi^b$, from the obstacle sensor output, $\{\ell_i^b\}$, and the hill ascent task gradient, $\nabla \phi^b$, from the hill gradient sensor output, ∇h^b . The module applies exponential smoothing¹⁷ to both of these components to prevent oversensitivity to non-persistent disturbances on the hill gradient and obstacle measurements. This filter also provides a remedy to cyclic body pose-variations stemming from robot locomotion and potential repercussions of the limited

¹⁷For the raw data sequence, $\{\mathbf{x}_t\}$, the smoothed data sequence, $\{\mathbf{y}_t\}$, can be generated as: $\mathbf{y}_0 = \mathbf{x}_0$ and $\mathbf{y}_t = \alpha \mathbf{x}_t + (1 - \alpha)\mathbf{y}_{t-1}$, where $0 < \alpha < 1$ [20]. In this way, the current value, \mathbf{y}_t , is affected by all previous values of the raw data sequence but older members of the sequence have exponentially diminishing importance.

field of view. We discuss some of these concerns further in Section 4.1.3. The output gradient, $\nabla \varphi^b$, is used by the kinematic unicycle control law introduced in (3.62).

4.1.3 Sensory and Physical Limitations

In this section, we address various sensory and physical limitations encountered during the implementation and present the developed solutions.

4.1.3.1 Bounded Control Inputs

One major concern in any robotic implementation is to make sure that the control inputs generated by the policy are realizable on the physical platform. We utilize a Fourier-style saturation term as introduced in [115], $\mu : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ with $\mu(\mathbf{x}) = \sqrt{|\mathbf{x}|^2 + \xi^2}$ where $\xi > 0$ constant, to construct the scalar valued function in the kinematic unicycle input, (3.62),

$$\sigma_{\varphi}(|\nabla\varphi|) := \nu_{\mathbf{p}} \frac{1}{\mu^2(\nabla\varphi)} = \nu_{\mathbf{p}} \frac{1}{|\nabla\varphi|^2 + \xi^2},$$

where $\nu_{\mathbf{p}} > 0$ is the scaling constant. This construction is suitable since it is bounded,

$$\sigma_{\varphi}(|\nabla \varphi|) \le \nu_{\mathbf{p}} \frac{1}{\xi^2},$$

and applied to (3.62), it saturates the individual components, resulting in,

1

$$\mathbf{u}_{ku} \le \nu_{\mathbf{p}} \begin{bmatrix} 1\\ \frac{\nu_{\theta}}{2\xi} \end{bmatrix}.$$
(4.3)

4.1.3.2 Field of View

As discussed in Section 4.1.1.1, we chose to use a LIDAR unit to realize the obstacle sensor. While it is very simple to utilize this sensor and process its output, its limited field of view presents a disparity from the model described in Section 3.1.3. As such, we limited the robot's translational movement to avoid any motion out of the field of view. In other words, if the combined gradient vector in body coordinates, $\nabla \varphi^b$, points a direction out of the LIDAR's field of view, the robot is only allowed to execute the rotational velocity input. To avoid undesired high frequency switching at certain extreme cases, a short term memory on $\nabla \varphi^b$ is realized through exponential smoothing of both of its components during task encoding. Although we have not performed a full analysis, we conjecture that the hybrid system emerging from this implementation results in a similarly stable behavior.

4.1.3.3 Cyclic Body Pose Variations

Cyclic pitch and roll variations caused by the robot locomotion is a nontrivial source of noise on the hill gradient measurements. When directly applied to the controller, the noise this vector results in the robot oscillating around its expected path. To remedy this effect, we utilize exponential smoothing on both the hill gradient and hill incline sensor outputs.

Especially at low grades, the inherent noise becomes significant, where filtering of the hill gradient vector is not sufficient. To avoid this, we introduce a dead band, where if the slope angle sensed is less than 6° , it is deemed as flat terrain, and the hill gradient sensor returns a unit vector aligned with the robot's forward direction. As a consequence, the robot does not stop at summits, and maintain its progress across intermediate plateaus.

4.1.3.4 Dynamic Unicycle Input

The control input introduced in Section 3.2.2.2, (3.72), includes the time derivative of the kinematic unicycle control input, (3.62), which is a function of the combined LaSalle function gradient, $\nabla \varphi$. Since the Hessian, H_{φ} , is not available to the robot, we chose to use the two-sample finite difference approximation,

$$\dot{\mathbf{u}}_{ku}(t_{i+1}) \approx \frac{1}{t_{i+1} - t_i} [\mathbf{u}_{ku}(t_{i+1}) - \mathbf{u}_{ku}(t_i)].$$
(4.4)

Despite its limited accuracy, since our dynamic unicycle approach can be loosely described as low-pass filtering of \mathbf{u}_{ku} applied to the system, in practice this approximation suffices to achieve the desired behavior with minimal computational cost¹⁸.

4.1.4 Parameter Tuning

As discussed in Section 3.2 in detail, stability of the goal set relies upon the choice of a suitable set of parameters. For the point particle agent (3.34), these parameters are obstacle function gain, ν_{ψ} , and the sensor range, ρ_{s} . These choices depend upon the maximum task gradient magnitude, Γ_{h} , the minimum task gradient magnitude over the obstacle regions of interest, Ω_{h} , the task curvature bound, κ_{ϕ} , and the maximum permissible obstacle radius, ρ_{max} , all of which are not available. Additionally, the success of the kinematic unicycle control law (3.62) relies on a sufficient choice of the kinematic unicycle coefficient, ν_{θ} , which depends on another unknown parameter, the minimum permissible obstacle radius, ρ_{min} . Lastly, even though the theoretical success of the dynamic unicycle control law (3.72) is guaranteed, the implemented behavior's performance depends on the choice of the dynamic unicycle control spin, $\nu_{\mathbf{r}}$.

To overcome the unavailability of all these parameters, we have devised a tuning policy for

¹⁸The acceleration dynamics governing the second order unicycle model are implemented via an Euler integral. Thus, only the finite difference in the numerator appears in the update equation for the velocity.

kinematic and dynamic unicycle agents, where we first make sure the parameter choices result in successful behavior without any obstacles. We start with a sufficiently large ν_{θ} value to succeed at walking speed over a variety of patches of hilly terrain with no obstacles. Too high of a value for ν_{θ} results in jittery robot behavior which is taken into consideration. Next, at running speed, we tune a $\nu_{\mathbf{r}}$ level that provides a fast enough convergence to the desired uphill walking behavior. Third, we manually pick different ρ_{s} values for slow and fast pace behaviors, where the value for the latter behavior is picked to be slightly larger to achieve a more graceful reaction of the physical platform to sensed changes in the environment. Based on the choice of ρ_{s} , we tune a sufficient ν_{ψ} value resulting in successful robot behavior at both walking and running speeds over a variety of initial conditions around a single obstacle.

4.2 Experimental Results

In this section, we present the results of multiple hill ascents conducted at two different sites with the RHex platform [42, 121]. We first describe the experimental sites used for all the experiments. We proceed with the first set of experiments where we test the performance of the kinematic unicycle controller, (3.62), at a slower (*walking*) pace and the dynamic unicycle controller, (3.72), at a faster (*running*) pace. We provide details about the experimental procedures, present the results at both speed levels, and discuss some of the common issues. We then discuss a second set of experiments that compare the two unicycle controllers at both speed levels, with details on procedures and analysis of the results.

4.2.1 Experiment Sites

To test the autonomous hill climbing behavior at both walking and running speed levels, several experiments were conducted in two different sites. The first of these, Penn Park¹⁹,

¹⁹3000 Walnut St, Philadelphia, PA 19104

has a human-built grassy hill patch which provided medium to steep slope angles (up to 36°) with a sparse obstacle course containing some young trees and the supports of a pedestrian crossing bridge. The latter, by introducing rectangular shapes and a wall into the mix of obstacles, constituted violations to the world model which added more diversity to the experiments. One side of this patch contained no obstacles but maintained the steep slope angle for around 15 meters. These experiments also tested the effects of the two unicycle control policies on the robot's pitch and roll stability at such high grades.

The second location, Ridley Creek State Park²⁰, provided some more difficult challenges for the robot with its uneven terrain and a dense distribution of both detectable and nondetectable obstacles. The detectable obstacle set included fully grown trees, medium to large size bushes and even fallen trunks, whereas non-detectable obstacle set included small size bushes, small rocks, fallen branches and pits hidden by a thick layer of leaves.

4.2.2 Performance Experiments

4.2.2.1 Procedure



Figure 4.3: Procedure for performance experiments. The steps taken to generate reported results are categorized as *on-site* and *post-processing*.

The procedure followed for performance experiments is summarized in Figure 4.3. The steps taken to generate reported results are categorized as *on-site* experiments and accompanying

 $^{^{20}1023}$ Sycamore Mills Rd, Media, PA 19063

measurements, and *post-processing* of recorded videos.

Every trial starts at a new location and an initial slope angle of at least 6° , where we record a video from its beginning on the initial slope until the operator declares its termination (annotated as either a *summit*, *edge*, or *fault* to be detailed below).²¹ At each declared terminus, we measure the direct distance traveled from the start to finish via a measurement wheel,²² and sample the hill incline angles with a digital level. Additional annotations on sensory inputs, their interpretations and control outputs are logged accordingly.²³

We generate time-to-travel and detectable and non-detectable obstacle encounter information manually from recorded data. In the processing of the video records, any abrupt change in robot heading is recorded as an obstacle encounter. If this abrupt change stemmed from the controller reacting to the presence of a tree or a high bush (taller than the LIDAR scan line) in the close path of the robot, the obstacle encountered is labeled as detectable. Otherwise, the obstacle is labeled as non-detectable. Lastly, any interruption in robot operation is recorded as a robot fault. If the fault is caused by an obstacle, it is labeled with the type of obstacle encountered. Otherwise, a label describing the issue is used. Lack of summit detection as described in Section 4.1.3.3 allows the robot to keep accumulating climbing statistics, where we do not count local ascents as summits in the count of Table 4.1 and Table 4.2. For the same reason, some experiments interrupted by faults are resumed through operator intervention but we report and account for each of these interruptions as faults in Table 4.1 and Table 4.2. Lastly, some trials are ended by the operator because of the robot reaching the edge of the course. This artificial boundary means two different things depending on the site. In Ridley Creek State Park, this specifically meant the end of lightly vegetated section of the hill patch the robot was operating on. Even if the robot could potentially keep going, it would be hard for the operator to go in and fetch the robot

²¹These video runs are available for download at http://cmass.seas.upenn.edu/hillascentjournal.

²²Especially at Ridley Creek State Park, the robot failed to receive any GPS signal, and thus, we could not report path lengths the robot traversed.

²³Data generated from manual processing of the experiment videos can be accessed from http://cmass. seas.upenn.edu/hillascentjournal in spreadsheet format. In addition, corresponding data log files and scripts to load them are included.

in case something went wrong, and thus, it was logical to end the session then. In the case of Penn Park, we had a bridge orthogonal to the uphill direction at the end of the hill patch, and the robot could keep going on the bridge as it had an uphill section. But this would be challenging for both the robot and the operator due to active pedestrian and bike traffic, and thus, the operator ended the run when the robot reached this area.

4.2.2.2 Results

Walking Speed A dataset of eleven experiments collected on four different hill sections tests the walking speed behavior, as summarized in Table 4.1. Overall, the robot climbed around half a kilometer²⁴ (461.8 meters, or 810.2 body lengths) of hilly terrain while encountering 111 obstacles and successfully avoiding 107 of them. 49 of the avoided obstacles were detectable by its sensor (trees, tall bushes and walls), and 58 of them were not detectable (short bushes, fallen branches and logs). The 4 obstacles the robot failed to avoid were not detectable by its sensor. In other words, the steepest ascent controller, with no obstacle avoidance term introduced, would otherwise have failed to avoid and likely become entrapped by 49 additional obstacles that the robot was able to avoid in these tests.

Trials 6 through 10 ended in a summit and Trials 1, 4 and 11 ended when the robot reached an edge of the course (an artificial boundary picked by the operator beyond which it is not safe for the robot). Trials 2, 3 and 5 were stopped when the robot suffered a fault i.e., it got stuck on a small but rigid branch from the under brush—a non-detectable but insurmountable obstacle. These failures of the world model (specifically, the assumption of Section 3.1.1 that any obstacle unseen by the sensor is surmountable), could be addressed by improvements in sensing or locomotion primitives that lie beyond the scope of this paper. Trial 2 contained an intermittent fault where a thick branch trapped the left rear leg. The operator pulled this branch off and the trial continued. Of note, Trial 11 tested the limits of

 $^{^{24}}$ The total runtime for all walking speed experiments is 37 minutes and 8 seconds. The distance reported is the direct distance between initial and final locations. The length of the path the robot traversed is not available.

#	Location	Description	Direct Distance	Hill Slope	Runtime	D. O.	N. O.	Faults	Finish
			(meters)	(degrees)	(min:sec)				
1	Ridley Creek	Medium Forest	69.5	10 - 15	5:22	8	11	-	Edge
2	Ridley Creek	Medium Forest	62.3	3 - 15	6:14	4	12	2x N.O.	Fault
3	Ridley Creek	Medium Forest	62.9	6 - 15	4:55	3	13	N.O.	Fault
4	Ridley Creek	Medium Forest	96.8	3 - 15	7:31	11	19	-	Edge
5	Ridley Creek	Steep Forest	18.7	15 - 18	1:47	2	3	N.O.	Fault
6	Penn Park	Medium Grassy	27.5	3 - 12	1:54	1	0	-	Summit
7	Penn Park	Medium Grassy	20.9	3 - 20	1:36	2	0	-	Summit
8	Penn Park	Steep Grassy	36.2	3 - 33	2:34	7	0	-	Summit
9	Penn Park	Steep Grassy	22.6	3 - 33	1:46	4	0	-	Summit
10	Penn Park	Medium Grassy	28.8	3 - 20	2:17	7	1	-	Summit
11	Penn Park	Steep Grassy	15.6	15 - 36	1:12	0	0	-	Edge

Table 4.1: Eleven outdoor hill climbing behavior trials including 49 detectable obstacles (D.O.) successfully avoided and 58 non-detectable obstacles (N.O.) successfully mechanically traversed over around half a kilometer of climbing with only 4 faults.

the hill ascent controller, where the hill incline angle reached 36° . Yet, the robot successfully traversed this patch of hill and reached the edge.

Running Speed A dataset of nine experiments collected on three of the same hill sections as the walking speed experiments tests the running speed behavior, as summarized in Table 4.2. Overall, the robot climbed 357.8 meters²⁵ (or 627.7 body lengths) of hilly terrain while encountering 89 obstacles, and successfully avoiding 85 of them. 41 of the avoided obstacles were detectable by its sensor (trees, tall bushes and walls), and 44 of them were not detectable (short bushes, fallen branches and logs). The robot got entrapped by 4 obstacles. Twice it got caught up on (non-detectable) rigid branches; another two times the navigation failed to clear 2 detectable obstacles, both of which violated the world model in a manner detailed below. In other words, the steepest ascent controller, with no obstacle avoidance term introduced, would otherwise have hit and likely become entrapped by 41 additional obstacles that the robot was able to avoid in these tests.

Trials 15 through 18 ended in a summit and Trials 13 and 20 ended when the robot reached

 $^{^{25}}$ In total, running speed experiments took only 11 minutes and 22 seconds. The distance reported is the direct distance between initial and final locations. The length of the path the robot traversed is not available.

#	Location	Description	Direct Distance (meters)	Hill Slope (degrees)	Runtime (min:sec)	D. O.	N. O.	Faults	Finish
12	Ridley Creek	Medium Forest	58.7	10 - 15	1:43	3	11	N.O., W.M.V.	Fault
13	Ridley Creek	Medium Forest	87.3	3 - 15	2:55	7	17	-	Edge
14	Ridley Creek	Medium Forest	89.5	3 - 15	3:18	12	16	W.M.V, N.O.	Fault
15	Penn Park	Medium Grassy	19.3	3 - 12	0:19	1	0	-	Summit
16	Penn Park	Medium Grassy	19.7	3 - 20	0:22	2	0	-	Summit
17	Penn Park	Steep Grassy	19.3	3 - 33	0:24	3	0	-	Summit
18	Penn Park	Steep Grassy	19.1	3 - 33	0:49	5	0	-	Summit
19	Penn Park	Medium Grassy	24.2	3 - 20	0:58	6	0	Hardware	Fault
20	Penn Park	Steep Grassy	12.5	15 - 36	0:34	2	0	$2 \mathrm{x}$ Flip	Edge

Table 4.2: Nine outdoor hill climbing behavior trials including 41 detectable obstacles (D.O.) successfully avoided and 44 non-detectable obstacles (N.O.) successfully mechanically traversed over around 350 meters of climbing with only 4 obstacle interaction based faults. 2 of these occurred due to robot failure over non-detectable obstacles. The other 2 occurred due to world model violations (W.M.V.) where a complex set of obstacles resulted in the robot control strategy failure.

an edge of the course (see the previous section for details of this termination condition). Trials 12, 14 and 19 were stopped when the robot incurred a fault after the reported distance had been covered. Trial 12 was terminated when the robot reached a fallen trunk and failed to walk around it (which could be considered and edge). This trial also contained an intermittent fault where a thick branch trapped left middle leg. The operator pulled this branch off and the trial continued. Trial 14 ended with the robot climbing over a short bush and losing traction as it can be seen on Figure 4.4. In addition, this trial was interrupted when the robot encountered a concave obstacle region formed by a wide tree and a big fallen branch. The operator moved the robot out of the trap and the trial continued. Trial 19 ended with a hardware failure where the left middle leg cracked. Similar to the previous section, Trial 20 tested the limits of the hill ascent controller where the hill incline angle reached 36°. Unlike the walking speed experiment, the robot would have flipped at two different instances without any operator intervention. Each of these interventions are marked as faults.



Figure 4.4: An extreme case: small bush trapping the robot at the end of Trial 14.

4.2.2.3 Common Issues

In this section, we address some of the issues encountered during the experiments. These issues all arose from disparities between the world model and the terrain encountered.

One of the common themes observed during these experiments was the non-detectable obstacle interfering with the steepest ascent direction measurements. An extreme case of such an encounter occurred in Trial 14. Three frames during this encounter shown in Figure 4.5 illustrate how the robot's interaction with a big fallen branch provided enough variations in body pitch and roll to interfere with the steepest ascent direction measurements. At the end of the encounter, the robot briefly paused as the pitch down motion put the steepest ascent direction at the back of the robot. This interaction is not recorded as a failure as the robot reacted to this interference as expected and it maintained its progress after this encounter.

Another problem encountered more than once during the experiments was a small branch stalling one of the legs. Even if the robot manages to keep moving, this situation can result in further damage due to a stalled motor and the operator intervention is inevitable without some specialized proprioceptive sensory suite and control modifications focusing



Figure 4.5: An extreme case: three frames illustrating the robot's interaction with a nondetectable obstacle and its effects on the steepest ascent direction during trial 1.

on detecting and recovering from such modes of failure, e.g. as in [63]. An example where operator intervention was inevitable occurred at the end of Trial 14. Figure 4.4 shows the result of robot's encounter with a bush tall enough to create enough interference with steepest ascent direction yet not tall enough to be detected by the obstacle sensor, where the robot partially climbed over the bush before getting stuck. Similarly, world model violations on obstacle shapes can result in failure of the control laws. A simple obstacle like a fallen trunk may result in entrapment of the robot with such simplified control laws as in Trials 12 and 14. Tackling all these problems is beyond the scope of this work but we provide some future directions to address such issues in Chapter 8.

4.2.3 Model Comparison Experiments

In this section, we compare the specific resistance of the robot governed by kinematic and dynamic unicycle controllers at both walking and running speeds; first without any obstacles, and then with a single obstacle introduced. The specific resistance formula²⁶ [121,

 $^{^{26}}$ Our measurements do not take the work done against gravity into account. For more accurate specific resistance measurements and thus fair comparisons of locomotion between level, sloped, and vertical surfaces, [50] proposes a model containing the original equation (4.5) plus an experimentally fitted correction term. Since our goal is to compare two different control laws over the same hill patches with the same initial and final elevations (an thus, with the same work against gravity), we chose to utilize the original specific resistance measurement.

144] is,

$$SR = \frac{P_{avg}}{m \, g \, v_{avg}}.\tag{4.5}$$

The average power, P_{avg} , is calculated by processing the power usage data provided by a custom battery monitoring solution, and the average speed, v_{avg} , is calculated by processing the position data from the GPS module contained by the IMU unit. The robot mass, m is measured to be 9.22 kg, and gravity, g, is assumed to be 9.81 m/s^2 .

4.2.3.1 Procedure

The procedures followed for the two sets of model comparison experiments are summarized in Figure 4.6 and Figure 4.7. The steps taken to generate reported results are categorized as *on-site* experiments and accompanying measurements, and *post-processing* of recorded logs from the robot.



Figure 4.6: Procedure for model comparison experiments with no obstacle. The steps taken to generate reported results are categorized as *on-site* and *post-processing*.

For the first set of experiments, where we seek the steady state performance with no obstacles, both P_{avg} and v_{avg} are computed over a portion of the trial where the translational velocity command applied to the lower level locomotion control reaches at least 95% of its maximum permissible value.²⁷ These experiments were all conducted on the same (roughly

 $^{^{27}}$ From (3.62) combined with (4.3), for a specific hill grade, a disparity between robot's heading and steepest ascent direction results in a smaller translational velocity component. On the other hand, when the

 $10^{\circ} - 15^{\circ}$ inclined) hill patch at Penn Park, comprising five trials per control choice and locomotion speed combination with identical start and finish locations (separated by a distance of roughly 20 body lengths), but initiated with varied headings relative to the hill ascent direction.²⁸



Figure 4.7: Procedure for model comparison experiments with a single obstacle. The steps taken to generate reported results are categorized as *on-site* and *post-processing*.

For the second set of experiments where the effect of an obstacle on robot performance is investigated, these average values are calculated over the portion where the combined task controller is active. In addition to specific resistance values based on these overall averages, we also computed moving averages with a window width of 50 samples (about 2.7 seconds), a corresponding series of specific resistance values, SR_w , and its mean and variance. These experiments were conducted with a single tree located roughly 3 body lengths above the 20° initial inclination angle. For each controller locomotion speed combination, three trials with the same start and finish locations were conducted. In contrast to the previous trials, there was no variation in initial heading to ensure the robot's interaction with the obstacle.

	No Obstacle							
Trial	Walk	xing	Running					
	Kinematic	Dynamic	Kinematic	Dynamic				
1	1.50	1.59	1.08	1.13				
2	1.34	1.63	1.14	1.17				
3	1.90	1.74	1.26	1.18				
4	1.55	1.57	1.18	1.17				
5	1.68	1.86	1.46	1.17				
Average	1.60	1.68	1.22	1.16				

Table 4.3: Comparative specific resistance values for kinematic and dynamic controllers operating at two different speeds in the absence of an obstacle. The kinematic controller exhibits lower cost of transport at walking speed and the dynamic controller exhibits better performance at running speed.

4.2.3.2 Results

The results for the first set of experiments, conducted at Penn Park, over the portion of the first hill patch with no obstacles, are summarized in Table 4.3. The average specific resistance values over five trials suggest the kinematic unicycle controller achieves only 5% improvement in cost of transport at walking speeds, whereas the dynamic unicycle model exhibits only 4% improvement at running speeds.

Table 4.4 summarizes the results for the second set of experiments conducted over the portion of the first hill adjacent to the location for the first set, containing a single obstacle. The average specific resistance over the three trials suggests that at walking speeds the kinematic unicycle achieves roughly 13% better efficiency than the alternative dynamic controller when interacting with an obstacle. Conversely, at running speeds, the dynamic unicycle controller achieves 15% better efficiency. The same trend can also be observed in mean values of SR_w . The mean and variance of SR_w show a bigger disparity between kinematic and dynamic unicycle controllers at running speeds, where, on average, there is

robot is aligned with the steepest ascent direction, translational velocity component reaches its maximum level.

 $^{^{28}}$ These heading variations were introduced manually but roughly (to the operator's best ability) consistently over the four sets of experiments.

	Single Obstacle											
Trial	Walking						Running					
	Kinematic			Dynamic		Kinematic			Dynamic			
			R_w	SR SF		R_w			R_w		SR_w	
		$\mid m$	v		$\mid m$	v	ыn	$\mid m$	v		$\mid m$	v
1	1.46	1.56	0.28	1.69	1.73	0.13	1.69	1.88	0.21	1.43	1.52	0.06
2	1.53	1.59	0.17	1.63	1.70	0.25	1.65	1.82	0.29	1.46	1.64	0.31
3	1.43	1.60	0.30	1.78	1.86	0.22	1.64	1.93	0.79	1.33	1.40	0.08
Average	1.48	1.58	0.25	1.70	1.76	0.20	1.66	1.88	0.43	1.41	1.52	0.15

a 20% change in mean and 65% change in variance.

Table 4.4: Comparative specific resistance, SR, and moving average based specific resistance series, SR_w , mean (m) and variance (v) for kinematic and dynamic controllers operating at two different speeds in the presence of an obstacle. The kinematic controller exhibits lower cost of transport at walking speed and the dynamic controller exhibits better performance at running speed. The obstacle avoidance maneuver incurs additional cost. Mean values of SR_w agree with the trends seen in SR values. At running speed, disparity between kinematic and dynamic unicycle controllers in terms of mean and variance values of SR_w is more prevalent.

The specific resistance values reported at Table 4.3 and Table 4.4 are higher than the previously reported [42] value for the platform, 0.9, which was recorded over featureless flat terrain. This is expected as our specific resistance measurement method does not take operating over inclines into account. Finally, despite showing some improvements in line with our intuition, the comparison experiments do not reveal a clear advantage for choosing one of the control strategies over the other for specific locomotion speeds or terrain conditions.

Chapter 5

Autonomous Stairwell Ascent

In this chapter, we report on another autonomous behavior for a legged robot previously reported in [58], allowing it to negotiate a non-trivial indoor environment thanks to its well designed preflex and feedback mediated controls. The term *preflex* [19] denotes a purely mechanical loop arising from the interaction of a designed, shaped body or compliant limb with some naturally occurring geometric and mechanical features of the robot's environment. The feedback policies we use all approach the ideal (and in many cases represent a formal instantiation) of an attractor-basin selected by some state-based switching logic implementing the *prepares* relation according to the sequential composition method proposed in [24]. Thus, the phrase *algorithmically simple* refers to our robot's sole reliance on the switched composition of online controllers to achieve autonomy.

Section 5.1 presents all the modeling decisions made regarding operational environment, task, robot, and sensors. In Section 5.2, we present the details of the behavior implementation, where Figure 5.3 describes the two-stage process enabling the robot to travel multi-flight stairwells in flowchart form. We end the chapter with experimental results presented in Section 5.3.
5.1 Robot and Task

This section details the sensorimotor models and simple world models underlying the empirical stairwell ascent behavior. The world model of the stair ascent task is complicated by the intermittent disappearance of the gradient beacon field (on flat landings) and the need to find specifically *marked* obstacles (flights of stairs) whereon a distinctly different gait yields robust ascent. The stair ascent behavior is accordingly complicated, and formal statements of correctness would have a stochastic character governed by the statistical properties of real stairwells. Although a formal demonstration of correctness lies beyond the scope of the present paper, we aim to present in this section a precise enough account of the world and sensorimotor models so as to enable future analysis (when coupled with our description in the following section of the behavior that relies upon them).

5.1.1 World Model

We now follow [57, 62] to introduce the very simple model of the *world* (Section 3.1.1), that will abstract away almost all the physical properties of the stairs and landings to provide a uniform view of the robot's task within its environment. This abstraction is appropriate on a platform such as RHex whose normal walking gait can safely handle small obstacles (debris or uneven surfaces). For the stairs, in this work, we assume no obstacle is present and the robot's stair climbing gait [99] can reliably traverse various stair designs.

5.1.1.1 The Stairwell Model

Definition 5.1.1. stairwell A stairwell is defined to be a piecewise constant terrain (Definition 3.1.1). Each constant (and compact) component is called a landing and it is surrounded by boundary obstacles (walls, cliffs) including a subset called a stair that connects it to the next landing. Note that we define a stair purely in terms of its perceptual features as detailed below in Section 5.1.4.5.

5.1.2 Robot Model

We utilize two different models depending which stairwell component the robot is currently operating. For the operations over a landing, we utilize the kinematic unicycle model introduced in Section 3.2.2.1. Over the stairs, on the other hand, we assume the stair climbing gait [99] reduces down to a scalar point particle tracking the single dimensional gradient defined by the slope of the stairs.

5.1.3 Task Model

The task of *autonomous stairwell ascent* requires that the robot locomote from any initial position and orientation over a stairwell to some landing with no upward *stair* boundaries.

5.1.4 Sensor Models

In this section we provide a list of abstract sensor models used for implementing the autonomous stairwell ascent behavior. These sensors are a succession of exteroceptive sensors that can be realized through the use of a LIDAR hardware unit mounted on a legged robot.

5.1.4.1 Depth Sensor

The depth sensor is an abstract map,

$$\sigma_E: SE(2) \times \mathcal{B} \times \mathcal{T} \to \mathcal{R}$$

that returns from each position and heading in the plane, $(\mathbf{p}, \theta) \in SE(2)$, bearing angle, $\beta \in \mathcal{B} := [\beta_m, \beta_M]$, and body pitch, $\tau \in \mathcal{T} := [\tau_m, \tau_M]$, a distance, $\rho \in \mathcal{R} := [0, \rho_M]$.

In our implementation, we use the output from a fixed LIDAR unit to realize this depth map. The arc extends roughly $\pm 120^{\circ}$ off center. The distance profile corresponds to the first depth at which the LIDAR unit records a return. The LIDAR unit cannot detect beyond a distance of $\rho_M := 4m$, to which the *infinite* reading of its maximum depth scale is calibrated.

5.1.4.2 Gap Sensor

The gap sensor is an abstract map,

$$\sigma_G: SE(2) \to \mathcal{B}$$

that returns for each position and orientation at which the robot is pointing, the center, $\sigma_G(\mathbf{p}, \theta) = \xi$ of an arc segment $[\xi - S, \xi + S] \subset \mathcal{B}$, a window within which the interval depth is maximum

$$\xi := \operatorname*{argmax}_{\beta_m + S \leq \bar{\beta} \leq \beta_M - S} I[\beta, S],$$

where,

$$I[\bar{\beta}, S] := \min_{\bar{\beta} - S \le \beta \le \bar{\beta} + S} \frac{\sigma_E(\mathbf{p}, \theta, 0, \beta)}{(1 - K)\cos^6(\beta - \bar{\beta}) + K},$$

contains the introduced bias towards lower bearing differences to emulate the search for a rectangular opening on the robot's path.

5.1.4.3 Pitch Scan Sensor

The pitch scan sensor, $\sigma_P : SE(2) \times \mathcal{B} \times \mathcal{T} \to \mathcal{R} \times \mathcal{B} \times \mathcal{T}$ is defined as,

$$\sigma_P(\mathbf{p},\theta,\tau_m,\tau_M) := \{ (\sigma_E(\mathbf{p},\theta,\tau,\beta),\,\beta,\,\tau) : \beta \in \mathcal{B}, \tau \in [\tau_m,\,\tau_M] \subset \mathcal{T} \}$$



Figure 5.1: The pitch wiggle behavior for up and down scans, with inactive legs removed for clarity.

and is implemented by running the depth sensor at each bearing angle within the field of view and pitch angle achieved via a coordinated motion of the legs — a *pitch wiggle* self-manipulation [62, 65, 122].

The pitch wiggle is a sensorimotor routine utilizing the planar LIDAR to measure ranges in many planes. With a LIDAR unit positioned horizontally with respect to the ground, a stair for example will appear similar to a wall. Unlike many robots that attach a LIDAR unit to a motorized tilting mechanism, we use RHex's natural ability to self-manipulate to a variety of angles in order to sweep the LIDAR's sensing plane. This maneuver produces a large variation in body pitch (either up or down) with no internal forces or toe slip²⁹, and is depicted in Figure 5.1. A more precise treatment of this self-manipulation behavior is presented in [65].

²⁹This pitch change can be easily derived from the geometry of the C-shaped legs.

5.1.4.4 Cliff Sensor

The cliff sensor, $\sigma_C : SE(2) \times \mathcal{B} \times \mathcal{T} \to \{0, 1\}$ is the composition $\sigma_{CD} \circ \sigma_P$. The pitch scan sensor, σ_P is pitched through a downward interval ($\tau_m < \tau_M < 0$) to scan a mid distance rectangular region on robot's path. The cliff detection sensor

$$\sigma_{CD}: \mathcal{R} \times \mathcal{B} \times \mathcal{T} \to \{0, 1\}$$

compares the results from σ_P with predicted range values from current pitch and bearing angles and returns a binary value based on the persistence of segments with extreme negative error. It contains two stages. In the first stage, ground range prediction error

$$\sigma_{GE}(\rho,\beta,\tau) := \mu(\beta,\tau) - \rho$$

is computed for every $(\rho, \beta, \tau) \in \sigma_P$ through the ground range prediction function $\mu: \mathcal{B} \times \mathcal{T} \to \mathcal{R}$ as

$$\mu(\beta,\tau) := \frac{0.5 \, l \tan(-\tau) + h_s}{\tan(-\tau)} \cdot \frac{1}{\cos\beta}$$

where, assuming that LIDAR is located at the geometric center, l is the length of robot's body and h_s is the total height of the LIDAR and robot body. After a unidirectional threshold, a binary value based on the persistence of segments with extreme negative error is returned.

5.1.4.5 Stair Sensor

The stair sensor, $\sigma_S : SE(2) \times \mathcal{B} \times \mathcal{T} \to \mathcal{R} \times \mathcal{B} \times \mathbb{S}^1 \times \{0, 1\}$ is the composition $\sigma_{SD} \circ \sigma_P$. The pitch scan sensor, σ_P is pitched through an upward interval $(0 < \tau_m < \tau_M)$.

The stair detection sensor $\sigma_{SD}: \mathcal{R} \times \mathcal{B} \times \mathcal{T} \to \mathcal{R} \times \mathcal{B} \times \mathbb{S}^1 \times \{0, 1\}$ returns the range ρ_S , bearing β_S and normal angle θ_S of the stairwell and a binary variable c_S indicating if the sensor is

confident about this detection. It outputs zero if it can not detect stairs. It is implemented in three stages. To detect and extract output parameters, a stairwell is modeled as a set of vertical plane segments with increasing horizontal offset where offset difference between successive plane segments are within a predefined interval $[d_l, d_u] \subset \mathcal{R}$. At the first stage, a line segment extractor $\sigma_{LS}(\sigma_P) := \{p_{\mathcal{L}}^i\}$ finds and parameterizes line segments,

$$\mathcal{L}^{i} := \{ (\rho, \beta, \tau) \in \sigma_{P} : \rho \cos(\beta) = \rho \sin(\beta) a^{i} + b^{i}, \tau = \tau^{i} \},\$$

on the LIDAR scanning plane for every pitch angle. A line segment is represented with five parameters: pitch angle τ^i , bearing interval boundaries β_m^i , β_M^i , normal angle $n^i = atan(-a^i)$, and horizontal offset $d^i = b^i \cos \tau^i$ where,

$$p_{\mathcal{L}}^i := (\tau^i, \beta_m^i, \beta_M^i, n^i, d^i).$$

Once all the line segments are extracted and parameterized, vertical plane segment extractor $\sigma_{PS}(\sigma_{LS}) := \{p_{\mathcal{P}}^j\}$ groups these line segments into vertical plane segments

$$\mathcal{P}^j := \left\{ p_{\mathcal{L}}^k \in \sigma_{LS} : [\beta_m^k, \beta_M^k] \cap [\beta_m^{k+1}, \beta_M^{k+1}] \neq \emptyset, n^k = n^j, d^k = d^j \right\}$$

by comparing individual bearing angle intervals, normal angles and horizontal offsets and performs a parametrization. A plane segment is represented by six parameters: pitch interval boundaries τ_m^j, τ_M^j , total bearing interval boundaries $\beta_m^j = \min_k \beta_m^k, \ \beta_m^j = \max_k \beta_M^k$, normal angle n^j and horizontal offset d^j where

$$p_{\mathcal{P}}^j := (\tau_m^j, \tau_M^j, \beta_m^j, \beta_M^j, n^j, d^j)$$

Finally, the stair extractor $\sigma_{SE}(\sigma_{PS}) := p_S$ returns the range, bearing and heading angles of the stairwell and a binary confidence variable if detected. It outputs zero otherwise. It

first extracts a stair candidate

$$S := \{ p_{\mathcal{P}}^{k} \in \sigma_{PS} : n^{k} = n_{S}, d^{k=0} = \rho_{S}, d_{l} \le d^{k+1} - d^{k} \le d_{u} \\ , [\tau_{m}^{k}, \tau_{M}^{k}] \cap [\tau_{m}^{k+1}, \tau_{M}^{k+1}] = \emptyset, [\beta_{m}^{k}, \beta_{M}^{k}] \cap [\beta_{m}^{k+1}, \beta_{M}^{k+1}] \neq \emptyset \}$$

by comparing pitch and bearing intervals, normal angles and horizontal offsets. A stairwell is represented by four parameters: stair distance ρ_S , stair central bearing angle β_S , stair heading $\theta_S = n_S + \theta$, and a binary confidence indicator c_S that is nonzero if minimum pitch angle; $\tau_m^{k=0}$ and absolute bearing angle $|\beta_S|$ are both within some confidence intervals

$$p_{\mathcal{S}} := (\rho_S, \beta_S, \theta_S, c_S)$$

The actual implementation employs two more preprocessing stages. During the first stage, beginning from the lowest pitch angle, any infinite reading for a specific bearing is replaced by the reading for the same bearing from the lower pitch angle scan. During the next stage, simple edge detector is employed to segment individual pitch angle scans into intervals.

5.2 Autonomous Stairwell Ascent

Because of the additional perceptual and motor activity associated with finding and negotiating stairs, the autonomous stairwell ascent behavior has greater complexity than the autonomous hill ascent in Chapter 4. Although we address the overall task through the systematic construction of pre-image backchaining [89], our reliance upon preflexes implies that not all the action steps will admit well-defined attractors and basins as required for the very robust and formally more powerful variant of sequential composition [24]. We report here on the presently functioning constituents of this behavior and leave for future work their formal reconciliation into that more powerful (but restrictive) framework.



Figure 5.2: Implementation details of the stair sensor. For all the graphs the vertical axis denotes the body pitch and the horizontal axis denotes relative bearing angle in degrees. The top two graphs contain the raw readings and the output of a simple filter.



Figure 5.3: Flow chart describing autonomous stair climbing.

5.2.1 The Stair Climbing Behavior

RHex robots have been climbing single-flight stairs for nearly a decade since Buehler's group first developed the appropriate gait [99] and they perform quite reliably on a variety of typical human-scale staircases. This capability owes much to the preflex yaw stabilization conferred by in-phase contra-lateral legs (providing a wide base of support on each successive stair) along with the metachronal gait that engages the circular legs just in time to place the body weight quasi-statically on the tread of the stair [77]. The preflexes arising from this gait ensure that RHex-style legged platforms ascend stairs in open-loop as if in the presence of the perceptually active steepest ascent stabilizing controller on hills (Chapter 4)³⁰.

The previous paragraph describes a controller that climbs the stairs essentially by establishing a (a component of the underlying *prepares* graph in the sense of [24]) from the domain of any individual step (say the first one) to any higher step (such as the last step). When

 $^{^{30}}$ And, as it turns out, at least as reliably in this task open-loop mode as any tracked robots under feedback control since the latter must place their weight on the nose of the stair for each step [103], which is contrary to the way stairs are intended to be used.

a *next* stairwell has been located on a given landing, in order to enter the domain of the stair climbing controller, the robot uses a transition from walking to stair climbing that has also been shown empirically to be reliable when the robot is walking towards the start of a stairwell $[51]^{31}$. The transition from previous stair to next landing is accomplished by a simple *stair exit* controller, triggered by the robot body pitch (as reported for a different robot in [51]), that commands a few open loop forward steps.

5.2.2 Landing Exploration Behavior

Once the robot climbs through a flight of stairs and reaches a new landing, a sequence of controllers (as summarized in Figure 5.3) is activated to drive it out of the prior goal set (i.e., the sensed zero-grade event that triggered the stair exit controller) and into the basin of the next as follows:

• Stair Detector

This controller first calls the stair sensor, σ_S and returns $(\rho_S, \beta_S, \theta_S, c_S)$ (Section 5.1.4.5). For nonzero output, this controller performs an open loop move to the relative pose $(\rho_S, \beta_S, \theta_S - \theta)$. If $c_S = 1$, the robot transitions into stair climbing behavior. Otherwise, it transitions back to σ_S for further investigation.

If σ_S returns 0, the robot switches to the Open Detector Controller.

• Open Detector

By calling the gap sensor σ_G (Section 5.1.4.2), this controller picks the most open bearing angle. At the beginning of each landing, the sign of this bearing angle is declared as the preferred direction to be used in case of future conflicts.

If no suitably open bearing angle is available (if $\sigma_E \circ \sigma_G(\mathbf{p}, \theta) < 1m$, i.e. the robot is in a corner) the robot simply rotates by 90° through the preferred direction and

 $^{^{31}}$ Now using the virtual contact sensor of [63] instead of mere controller error to trigger the same transition more reliably.

transitions back to the Stair Detector. In the presence of a suitably open bearing, the robot rotates to this angle and switches to the Cliff Detector Controller.

• Cliff Detector

This controller first runs the cliff detector sensor σ_C (Section 5.1.4.4) to ensure it will not fall by pursuing this new heading. If this controller returns 0, robot walks for up to one meter, otherwise it rotates back through the preferred direction and transitions back to the Stair Detector Controller.

It seems unreasonable to expect any deterministic guarantees that the robot can reach the basin of the next stairwell ascent controller (i.e., the first steps of the next upward stairs) through this sequence of controllers flight through this slow process. Empirically, the data show that this behavior finds the subsequent stairwell with very high probability as landings are generally metrically small, topologically simple and not maze-like.

With this *landing exploration* behavior the (informal) sequential composition backchaining is completed and produces a roughly cyclic iterated path through controllers until the robot reaches the top of a stairwell. Figure 5.3 summarizes the entire behavior.

5.3 Experimental Results

To test the autonomous stairwell climbing behavior we ran the robot on 10 of the many different stairwells in 4 nearby buildings³², as Table 5.1 summarizes³³. We distinguish *behavior faults* (arising from inadequacies in either the algorithm or the sensorimotor capabilities that sub-serve it) from *robot faults* (failures due to mechanical or electronic unreliability). Only two of the stairwells met the requirements of our world model. Specifically, they exhibited

³²There were three additional stairwells that were attempted but on which the robot made no progress due to their having either open risers or glossy painted risers that the laser scanner could not see well if at all. This is a limitation of the sensor and these stairwells are not reported.

³³Naturally every stairwell is unique, and even within a stairwell the rise, run, width, steps per flight, landing size, style, and wall type can vary significantly. Listed here are typical values for a given stairwell that attempt to convey some of these differences without providing full blueprints.

#	Violation	Rise (cm)	Run (cm)	Landing	Landing Size (cm x cm)	# Flights	# Stairs	Time (hour:min:sec)	# Scans (stair, cliff)	Behavior	Robot
1	-	15.3	28.0	Straight	189 x 150	2	11	0:01:51	2, 0	-	-
2	-	15.3	28.0	Straight	$327 \ge 150$	2	11	0:03:01	3, 1	-	-
3	Glass	17.4	29.6	Straight	$192 \ge 143$	2	27	0:02:27	2, 0	-	-
4	Glass	16.7	26.9	Mixed	$256 \ge 277$	3	25	0:07:20	7, 4	-	-
5	Various	17.5	31.4	U-Left	$768 \ge 653$	6	81	0:50:05	47, 36	1S	1N, 2L
6	Window	18.2	26.3	U-Left	$486 \ge 222$	7	60	0:25:25	33, 22	1S, 1T	2N, 2L
7	Glass	16.2	28.5	U-Left	$471 \ge 252$	10	111	1:03:25	51, 36	$1\mathrm{C}$	3N, 3L
8	Glass	17.3	27.2	U-Left	$349 \ge 156$	10	112	0:54:40	55, 39	2T, 1C	1N, 1LD, 1L
9	Mesh	17.3	27.2	Mixed	$293 \ge 137$	11	112	0:44:54	44, 26	1T, 2W	2N, 1LD
10	Heater	17.5	26.0	U-Left	$228 \ge 122$	14	181	1:00:59	49, 27	2T	1LD, 1I

Table 5.1: Ten indoor stairwell climbing behavior trials covering 731 stairs in 67 flights with a total of 12 behavioral problems. World model violations are briefly described. Rise, Run and Landing Size dimensions are given in centimeters (cm). Scans column contains two numbers; Stair Scans and Cliff Scans. Behavior faults are categorized as (S)tair Detection, (C)liff Detection, Stair (T)ransition, and (W)all Collision. Robot faults fall into 4 categories; (N)etwork Communication, (L)eg Failures, (L)I(D)AR Failures, and (I)MU Failures.

solid, detectable walls and no significant stepped features on the landings. The rest of the stairwells violated our world model assumptions, but the robot was still able to climb with significant success. Thus the simple world model, while upon initial inspection seems to provide a general description of most stairwells, in practice there is a surprisingly rich diversity of stairwell structures, notwithstanding which, even in the face of unexpected variation, the stair climbing behavior based on this simple world model assumption was able to perform with reasonable success. The robot had only two false positives on stair detection throughout 67 flights of stairs. In particular one of these two failures occurred because the specific landing had a window whose frame combined with the wall fit the stairwell model described in Section 5.1.4.5. The other failure could be avoided by opting out too small pitch angles as they managed to create enough features to mislead the plane segment extractor. Similarly there were only two wall collision based failures and both happened on stairwell number 5 where the laser scanner could see through the mesh walls and detect open space even though the mesh is actually an obstacle to the robot, leading to collisions that in turn precipitated faults requiring operator intervention. Cliff detection thresholds were rather conservative during the experiments to avoid any false positives which resulted in two possible cliff falls avoided by operator intervention. The remaining 6 behavior failures occurred during initial stair transitions. These could be avoided by more intense sensor integration which is out of the scope of our current efforts.

In addition to behavioral faults, there were 21 robot faults over all 67 flights. The majority of these arose from a leg failing to respond (8 times) and from network communication issues (9 times) — the former due to known power distribution issues partially addressed in the midst of experimentation and expected to be fully resolved in very near future. Additionally, there were 3 LIDAR failures each of which happened due to overheating. These failures resulted in low quality readings which we were able to fix by power cycling the LIDAR. In the future these failures can be fully avoided by simple heat dissipation solutions. We had an IMU failure once due to a loose USB cable which happened during stair climbing and so the robot could not detect the end of the stairs.

Overall the behavior was able to climb a total of 731 stairs in 67 flights while encountering only 12 behavioral faults in over 5 hours of testing. In almost every stairwell, there were many other incidents that could be considered faults (such as a leg hitting a wall, open loop walking leaving the robot at the wrong angle, etc.) but the robust preflexes and reactive behaviors prevented these from requiring a human intervention.

Part III

World Model Violations

Chapter 6

Dynamical Trajectory Replanning for Uncertain Environments

In Chapter 3, we rely on the basic assumption that the *world* the robot operates in contains circular obstacles that are suitably located, sized, and separated. This simplification enables our systematic approach to the point particle and horizontal unicycle control strategies. On the other hand, in Chapter 4, we document two instances of robot operation being interrupted while negotiating with obstacles that violate this assumption. Moreover, consider the kind of obstacles depicted in Figure 6.1, Figure 6.3, and Figure 6.4. For a point particle agent governed by the control law of Section 3.2.1.2, depending on its initial state, *entrapment* is inevitable. How can we modify this strategy in a way that reacts to such violations successfully, but still maintains successful task execution? In this chapter, we present the work from [115], where we investigate a control law accomplishing the desired behavior.

We start the chapter with the motivation (Section 6.1) and background ideas (Section 6.2) behind this approach. We proceed with the controller design (Section 6.3). We, then, present two applications of the construction (Section 6.4) accompanied with simulation studies (Section 6.5). We end this chapter with a discussion on how to extend this approach to a kinematic unicycle agent despite the fact that, in its current form, this control law violates both the sensory capability and mobility assumptions of Chapter 3 (Section 6.6).

6.1 Motivation

Unlike the first order point particle model of Section 3.2.1.2, this work focuses on the tracking problem for a fully actuated, force-controlled, unit-mass point-mechanism with configuration space $\mathcal{Q} := \mathbb{R}^n$ subject to a force disturbance \mathbf{d} ,

$$\ddot{\mathbf{q}} = Q[\mathbf{q}; \mathbf{r}] + \mathbf{d},\tag{6.1}$$

where Q denotes a causal functional of the trajectory of the plant \mathbf{q} and of a desired reference motion path \mathbf{r} . Because physical actuators suffer severe limitations we restrict attention to designs for which both the force input (the output of Q) and the rate of mechanical work (omitting work done by the disturbance) are bounded.

In the traditional robotics and control paradigm [29] some higher level planner generates a sufficiently smooth³⁴ reference trajectory $\mathbf{r} : \mathbb{R} \to \mathcal{Q}$ that encodes the task at hand. The presumably task-naive but tracking-expert controller produces forces excited by the augmented tracking error, $Q[\mathbf{q};\mathbf{r}] := \ddot{\mathbf{r}} - E[\mathbf{e},\dot{\mathbf{e}}]$ where $\mathbf{e} := \mathbf{r} - \mathbf{q} \in \mathcal{Q}$ and E is a force law chosen so that the resulting tracking error system,

$$\ddot{\mathbf{e}} = E(\mathbf{e}, \dot{\mathbf{e}}) - \mathbf{d},\tag{6.2}$$

converges as strongly as possible to zero despite disturbances **d**. Within the controls field, one counterpart to our work is the longstanding anti-windup literature [138] wherein the unexecutably high authority commands of some nominal tracking controller are trimmed back to respect the saturating nature of inputs to the plant (6.1), and the very active

³⁴It is convenient to assume that all our signals are C^{∞} , but physical actuators are generally adequately protected from long term mechanical (albeit not necessarily thermal [34]) harm by C^2 inputs.

reference governor literature [45] provides controllers which do so with formal convergence guarantees. Indeed, any of the variants on these constructions which yield iISS [131] closed loops (6.2) with guaranteed Lyapunov functions [16], suitable for second order systems [100] would be appropriate candidates to generate the posited error tracker (6.2), although for purposes of illustration in this work we use a very much simpler saturating potentialdissipative [78] tracker (6.21).

In contrast, our focus is the question of what benefit can be achieved by modifying the reference trajectory \mathbf{r} in the face of online exposure to the disturbances \mathbf{d} . Specifically, we advance an architecture relevant to the growing class of robots [22, 92, 105, 119, 121, 149] whose reference trajectories are dynamically generated by allowing disturbance induced tracking errors (6.2) to excite a transient *replanner* subsystem that alters the reference generator in a stable manner.

The inevitable inaccuracies in world model, sensor acuity and actuator fidelity represented by \mathbf{d} in (6.1) usually have a systematic (albeit unmodeled) as well as a random component and we believe that such recourse to simple *dynamical replanning* may allow the plant to avoid rather than fight against otherwise intransigent if not adversarial obstacles.

6.2 Background Ideas

In this section we present the conceptual geometric ideas that lead us to our design. The equations in this section are not used in our main result. Instead, they are intended provide the rationale for the more elaborate constructions that follow.

6.2.1 A First Order Graph as a Second Order Attractor

Following [73], we assume a given geometrically defined task [76] encoded as a smooth first order reference dynamical system,

$$\dot{\mathbf{r}} = \mathbf{f}(\mathbf{r}),\tag{6.3}$$

endowed with a known Lyapunov function $\phi_{\mathbf{r}}$. Examples of nontrivial geometrically defined tasks that are nicely amenable to second order lifts of first order dynamical encodings are obstacle avoidance problems [73, 79], group formation coordination [10], and even complex kinodynamic motion planning problems [32, 118]. Define the second order *lift*

$$\ddot{\mathbf{r}} = R_{(\mathbf{f},\phi_{\mathbf{r}})}(\mathbf{r},\dot{\mathbf{r}}) := \dot{\mathbf{f}} - \kappa_{\mathbf{r}}[\dot{\mathbf{r}} - \mathbf{f}(\mathbf{r})] - \nabla\phi_{\mathbf{r}}\mathbf{r}$$
(6.4)

where $\dot{\mathbf{f}} := D_{\mathbf{r}} \mathbf{f}(\mathbf{r}) \dot{\mathbf{r}}, \nabla \phi_{\mathbf{r}} \mathbf{r} := [D_{\mathbf{r}} \phi_{\mathbf{r}}]^T$. Observe [73] $\eta_{\mathbf{r}} := \phi_{\mathbf{r}} + \frac{1}{2} |\dot{\mathbf{r}} - \mathbf{f}(\mathbf{r})|^2$ is a Lyapunov function for (6.4).

In the next section we will replace (6.4) with an augmented construction (6.8),(6.9) that accepts the replanner's transient inputs, respecting which appropriate assumptions on $\phi_{\mathbf{r}}$ insure that $\eta_{\mathbf{r}}$ is an ISS-Lyapunov function as well.

6.2.2 Internal Dynamical Reference Generators

Although there can be great virtue in self-excited designs wherein a copy of (6.3) is placed directly in the plant's feedback path (e.g. [22, 92, 105, 119]) this work focuses on a control scheme that places the reference dynamics in the feed-forward pathway using a design akin to

$$\ddot{\mathbf{r}} = R(\mathbf{r}, \dot{\mathbf{r}}) + \mathbf{u}(\mathbf{e})$$

$$\ddot{\mathbf{q}} = \ddot{\mathbf{r}} - E(\mathbf{e}, \dot{\mathbf{e}}) + \mathbf{d}$$
(6.5)

For example, the original RHex [121] controller adopted a completely open loop version (i.e., with $\mathbf{u} \equiv 0$) of this architecture on the torus, $\mathcal{Q} \approx \mathbb{T}^N$. A compensating feedback term was added and tuned to achieve better performance subsequently in RHex [145], and has proven essential to the RiSE climbing machine [133]. The lift in (6.5) of the reference dynamics (6.3) now constitutes an internal model (a separate *imagined* copy of \mathbf{q} representing the desired plant state and future trajectory) whose value we seek to exploit in recognizing situations of *surprise* and replanning in response.

Toward that end, we now proceed to develop a controller design recipe that augments this internal model with a maneuver-generator/replanner, \mathbf{s} , governed by a smooth timeinvariant vector field, \mathbf{g} , over some Euclidean space, S. This replanner excites the reference dynamics to express recovery maneuvers when an error builds up.

A consequence of *physical* restrictions is that the system cannot reject all bounded disturbances, since adversarial or even blind disturbances larger than the system's force and power budget can always disrupt any controller's attempts at correction. Instead, a more subtle notion of stability is needed, the notion of *Integral Input to State Stability (iISS)* [3, 131], which relates the L_2 norm of the disturbance to a (L_{∞}) bound on the state of the controller. We make additional use of control-theoretic tools from the *Input to State Stability (ISS)* [130] toolbox in demonstrating that our cascaded design has a response to the disturbance that is guaranteed to be bounded, and which will ultimately converge to the desired motion if the disturbance ceases. In the context of persistent state-dependent disturbances such as the unknown terrain obstacles in our examples, the disturbance ceases whenever the system manages to bypass these obstacles. Thus, we are guaranteed that should it succeed in escaping entrapment, the system will resume correct behavior. Notice that trajectories generated through this design recipe are not optimal in any sense. We merely guarantee that the replanner implements a feasible course of action in the face of arbitrary disturbances while respecting force and power limitations.

6.3 Controller Design

Denote the zero section over any submanifold $\mathcal{X} \subseteq \mathcal{Q}$ as $\mathcal{Z}_{\mathcal{X}} := \{(\mathbf{q}, 0) \in T\mathcal{Q} | \mathbf{q} \in \mathcal{X}\}.$

Assume the following design requirements from the component dynamical systems:

- (1) A fully actuated, unit mass, second order plant with state $(\mathbf{q}, \dot{\mathbf{q}}) \in T\mathcal{Q}$.
- (2) A task encoded as a first order dynamical control system

$$\dot{\mathbf{r}} = \mathbf{f}(\mathbf{r}) + \mathbf{v}(\mathbf{r}, \mathbf{s}) \tag{6.6}$$

over $\mathbf{r} \in \mathcal{R} \subseteq \mathcal{Q}$, with input $\mathbf{s} \in \mathcal{S}$.

- (2a) (6.6) is ISS with respect to some compact attractor $\mathcal{G}_{\mathbf{r}}$ and the input s.
- (2b) The coupling term $\mathbf{v}(\mathbf{r}, \mathbf{s})$ is monotonically bounded in $|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}}$ with respect to a \mathcal{K}_{∞} comparison function $\nu(\cdot)$: $|\mathbf{v}(\mathbf{r}, \mathbf{s})| < \nu(|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}})$
- (2c) The task admits $\phi_{\mathbf{r}}$, a smooth ISS-Lyapunov function (in the sense of [132] Sec.2.1) which also has a saturating gradient. Namely, there exists some $F_{max} \in \mathbb{R}_{>0}$ such that $|\nabla \phi_{\mathbf{r}}(\mathbf{r})| \leq F_{max}$
- (2d) The replanner excitation function $\mathbf{u} : \mathcal{Q} \to T\mathcal{S}$ is zero at zero, globally bounded $\|\mathbf{u}(\mathbf{e})\| < \mathbf{u}_{max}$ and continuous everywhere except perhaps at zero.
- (3) A replanner encoded as a first order dynamical control system $\dot{\mathbf{s}} = \mathbf{g}(\mathbf{s}) + \mathbf{u}$ which is ISS with respect to some compact attractor $\mathcal{G}_{\mathbf{s}}$ and the input \mathbf{u} .
- (4) A tracker system \vec{e} = E(e, \vec{e}) + d which is iISS with respect to the point attractor \$\mathcal{Z}_0\$ and the input d.

Using these components, selecting a gain $\kappa_{\mathbf{r}} \in \mathbb{R}_{>0}$, and defining $\mathbf{e} := \mathbf{r} - \mathbf{q} \in \mathcal{Q}$, we propose

a control system in the following form:

$$\dot{\mathbf{s}} = \mathbf{g}(\mathbf{s}) + \mathbf{u}(\mathbf{e}) \tag{6.7}$$

$$\dot{\mathbf{w}} = -\kappa_{\mathbf{r}} \mathbf{w} - \nabla \phi_{\mathbf{r}}(\mathbf{r}) \tag{6.8}$$

$$\dot{\mathbf{r}} = \mathbf{w} + \mathbf{f}(\mathbf{r}) + \mathbf{v}(\mathbf{r}, \mathbf{s}) \tag{6.9}$$

$$\ddot{\mathbf{q}} = \ddot{\mathbf{r}} - E(\mathbf{e}, \, \dot{\mathbf{e}}) - \mathbf{d} \tag{6.10}$$

$$\ddot{\mathbf{e}} = E(\mathbf{e}, \, \dot{\mathbf{e}}) + \mathbf{d} \tag{6.10*}$$

Our key theoretical result is expressed as follows:

Theorem 6.3.1. The proposed architecture (6.7)-(6.10), possesses the following stability properties:

- **[iISS]** The combined dynamics of $(\mathbf{e}, \dot{\mathbf{e}}, \mathbf{r}, \mathbf{w}, \mathbf{s})$ is iISS with respect to input \mathbf{d} and the attractor \mathcal{A} where $\mathcal{A} := \mathcal{Z}_0 \times \mathcal{Z}_{\mathcal{G}_{\mathbf{r}}} \times \{0\}$. \mathcal{A} is an attracting invariant submanifold of the unforced system (i.e. $\mathbf{d} \equiv 0$).
- **[ISS]** The projection of the system to $(\mathbf{r}, \mathbf{w}, \mathbf{s})$ is ISS with respect to the attractor $\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}} \times \{0\}$ and the input \mathbf{e} .
- **[BP]** The undisturbed $(d \equiv 0)$ input to the mechanical plant (6.10) and its internal mechanical power are both bounded.

Note that for the range of intended applications the disturbance will be (in part) state dependent and we have not yet established any useful sufficient conditions (e.g., properties of the replanner relative to the obstacles' shapes and placements) guaranteeing that the disturbance will have bounded energy (e.g., that the replanner will succeed in eluding those obstacles). The theorem merely guarantees the replanner will not itself destabilize the internal reference and mechanical plant dynamics assuming the disturbance desists. *Proof.* The proof that follows relies strongly on various properties of ISS systems and iISS systems; see [131] for an excellent tutorial overview of these ideas.

Given Proposition 6.3.2 below, we conclude that the second order system (6.8), (6.9) is ISS with respect to its input **s**. The system (6.7) was assumed to be ISS with respect to its attractor $\mathcal{G}_{\mathbf{r}}$ and the input **u**. The (compact-set)-ISS property is preserved by cascade composition, thus (6.7) into (6.8) into (6.9) is ISS with respect to the input **u**, proving [ISS]

Because **u** is bounded by construction, and the ISS property implies *Bounded Input to Bounded State (BIBS)*, [ISS] also proves that $(\mathbf{r}, \mathbf{w}, \mathbf{s})$ are bounded, and thus [BP] is proven via (6.10).

As per design requirement (4), (6.10^*) is iISS. From proposition 2 of [85], cascade of an iISS system into an ISS system is also iISS proving that cascading (6.10^{*}) into (6.7), (6.8) and (6.9) is iISS and establishing [iISS].

Proposition 6.3.2. The system $(\mathbf{r}, \mathbf{w}) \in T\mathcal{R}$ from (6.8), (6.9) is ISS with respect to input **s** and compact attractor $\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}}$.

Proof. The zero section $\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}}$ is compact from the previously assumed compactness of $\mathcal{G}_{\mathbf{r}}$. Sontag and Wang [132] Section 2.1 provide two equivalent definitions for an *ISS-Lyapunov* function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ whose existence with respect to some compact goal set $\mathcal{H} \subseteq \mathbb{R}^n$ is equivalent to the ISS property with respect to \mathcal{H} . V must be proper and positive definite with respect to \mathcal{H} , and there must exist comparison functions $\alpha_1, \alpha_2, \chi \in \mathcal{K}_{\infty}$ such that for all $\xi \in \mathbb{R}^n$:

$$\alpha_1(|\xi|_{\mathcal{H}}) \le V(\xi) \le \alpha_2(|\xi|_{\mathcal{H}}) \tag{[132] eqn. 5}$$

$$\xi \neq 0 \land |\xi|_{\mathcal{H}} \ge \chi(|v|) \Rightarrow V(\xi) < 0 \qquad ([132] \text{ eqn. 8})$$

We proceed to show that $\eta_{\mathbf{r}}(\mathbf{r}, \mathbf{w}) := \frac{1}{2}\mathbf{w}^2 + \phi_{\mathbf{r}}(\mathbf{r})$ is an ISS-Lyapunov function.

From the assumption that $\phi_{\mathbf{r}}(\mathbf{r})$ is ISS-Lyapunov we conclude that it is smooth, proper, positive, and vanishes precisely on the set $\mathcal{G}_{\mathbf{r}}$, and thus $\eta_{\mathbf{r}}$ is also smooth, proper, positive and vanishes precisely on the set $\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}}$. Now taking the Lie derivative of $\phi_{\mathbf{r}}$ along the motions of system (6.8), (6.9) we have

$$\begin{aligned} \dot{\eta}_{\mathbf{r}} &= \dot{\mathbf{w}} \cdot \mathbf{w} + \nabla \phi_{\mathbf{r}} \cdot \dot{\mathbf{r}} \\ &= -\kappa_{\mathbf{r}} \left| \mathbf{w} \right|^2 + \nabla \phi_{\mathbf{r}} \cdot (\mathbf{f} + \mathbf{v}) \\ &= -\kappa_{\mathbf{r}} \left| \mathbf{w} \right|^2 + \nabla \phi_{\mathbf{r}} \cdot \mathbf{f}(\mathbf{r}) + \nabla \phi_{\mathbf{r}} \cdot \mathbf{v}(\mathbf{r}, \mathbf{s}) \end{aligned}$$
(6.11)

From [132] eqn. (8) applied to $\phi_{\mathbf{r}}$, we conclude the existence of a comparison function $\chi \in \mathcal{K}_{\infty}$ that satisfies

$$|\mathbf{r}|_{\mathcal{G}_{\mathbf{r}}} > \chi(|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}}) \Rightarrow \nabla \phi_{\mathbf{r}}(\mathbf{r}) \cdot (\mathbf{f} + \mathbf{v}) < 0.$$
(6.12)

As an ISS-Lyapunov function with respect to input \mathbf{v} , $\phi_{\mathbf{r}}$ is also perforce a Lyapunov function for the zero input system $\dot{\mathbf{r}} = \mathbf{f}(\mathbf{r})$, and we conclude $\nabla \phi_{\mathbf{r}} \cdot \mathbf{f} \leq 0$ everywhere except $\mathcal{G}_{\mathbf{r}}$.

With these observations in hand, we define a comparison function $\beta(\cdot)$

$$\beta^2(x) := (F_{max}/\kappa_{\mathbf{r}})\,\nu(x) + \chi^2(x), \tag{6.13}$$

and note that $\nu, \chi \in \mathcal{K}_{\infty}$ ensure $\beta \in \mathcal{K}_{\infty}$.

We wish to show that $|\mathbf{r}, \mathbf{w}|_{\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}}} > \beta(|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}})$ implies $\dot{\eta}_{\mathbf{r}} < 0$, and so as to satisfy [132] eqn. (8). Consider two cases: $|\mathbf{r}|_{\mathcal{G}_{\mathbf{r}}} > \chi(|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}})$ and its complement. In the first case, because the term $-\kappa_{\mathbf{r}} |\mathbf{w}|^2$ in (6.11) is negative definite we have $\nabla \phi_{\mathbf{r}}(\mathbf{r}) \cdot (\mathbf{f} + \mathbf{v}) < 0$ from (6.12) and therefore $\dot{\eta}_{\mathbf{r}} < 0$ is satisfied. It remains to handle the complementary case $|\mathbf{r}|_{\mathcal{G}_{\mathbf{r}}} \leq \chi(|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}})$. By definition, $|\mathbf{r}, \mathbf{w}|^2_{\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}}} := |\mathbf{w}|^2 + |\mathbf{r}|^2_{\mathcal{G}_{\mathbf{r}}}$, motivating the derivation

$$\begin{split} |\mathbf{r}, \mathbf{w}|_{\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}}}^2 &= |\mathbf{w}|^2 + |\mathbf{r}|_{\mathcal{G}_{\mathbf{r}}}^2 > \beta^2 (|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}}) \\ |\mathbf{w}|^2 &> \beta^2 (|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}}) - \chi^2 (|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}}), \end{split}$$

with the last step using the assumption of this case. Substituting from (6.13), obtain

$$\kappa_{\mathbf{r}} |\mathbf{w}|^2 > F_{max} \nu(x) > |\nabla \phi_{\mathbf{r}}(\mathbf{r})| \cdot |\mathbf{v}(\mathbf{r}, \mathbf{s})|$$
(6.14)

$$> \nabla \phi_{\mathbf{r}}(\mathbf{r}) \cdot \mathbf{v}(\mathbf{r}, \mathbf{s}) + \nabla \phi_{\mathbf{r}} \cdot \mathbf{f}(\mathbf{r})$$
(6.15)

with the (6.14) from design requirements (2b) and (2c); and (6.15) from $\nabla \phi_{\mathbf{r}} \cdot \mathbf{f} \leq 0$.

From (6.15), we see that in both cases considered the RHS of (6.11) is negative definite with respect to the compact set $\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}}$. This RHS vanishes only on $\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}}$ itself. We conclude that with the comparison function $\beta(|\mathbf{s}|_{\mathcal{G}_{\mathbf{s}}})$, $\eta_{\mathbf{r}}$ is proven to be a (compact-set) ISS-Lyapunov function for the input \mathbf{s} and the attractor $\mathcal{Z}_{\mathcal{G}_{\mathbf{r}}}$.

6.4 Application of the Construction

In these examples, the configuration space is the Euclidean plane \mathbb{R}^2 , and thus vector spaces \mathcal{R} , **s** and \mathcal{Q} are all copies of \mathbb{R}^2 . Denote by J the antisymmetric matrix $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, and (by abuse of notation) define a matrix valued $J(x, y) := \begin{bmatrix} x & -y \\ y & x \end{bmatrix}$ that takes each point $(x, y) \in \mathcal{Q}$ to an orthogonal basis whose first vector is $(x, y)^T$. A useful constituent in the constructions to follow is the function $\mu(\tau) := (\tau + \alpha^2)^{\frac{1}{2}}$, where τ is a positive scalar and α is a positive scalar and α is a positive scale parameter to be selected.

6.4.1 Reference Generator

The reference (6.6) must be ISS with respect to the input **s** coupled via $\mathbf{v}(\cdot, \cdot)$, and with respect to a compact goal $\mathcal{G}_{\mathbf{r}}$. We would like the replanner to backtrack along the most recent motions of the plant and then try to move around the obstacle, and therefore maneuvers should act in a direction opposite to the most recent motion. If we assume that tracking error is small, the most recent motion would have been in the direction of $\mathbf{f}(\mathbf{r})$. We therefore coupled the maneuver into the reference taking the direction of the reference vector field as the first axis

$$\mathbf{v}(\mathbf{r}, \mathbf{s}) := \frac{c_{\mathbf{s}\mathbf{r}}}{\mu(\mathbf{f}^T \mathbf{f})} \mathbf{J}(\mathbf{f}) \,\mathbf{s}.$$
(6.16)

Observe that for any $\tau > 0$, $\mu(\tau) \ge \tau$, giving $\mu(\mathbf{f}) \ge ||\mathbf{J}(\mathbf{f})||$ and thus $|\mathbf{v}(\mathbf{r}, \mathbf{s})| \le c_{\mathbf{sr}}|\mathbf{s}|$, satisfying requirement (2b). Given a smooth Lyapunov function $\phi_{\mathbf{r}}(\mathbf{r})$ for the system $\dot{\mathbf{r}} = \mathbf{f}(\mathbf{r})$, there exists a comparison function $\xi \in \mathcal{K}_{\infty}$ such that $\nabla \phi_{\mathbf{r}}(\mathbf{r}) \cdot \mathbf{f} \le -\xi(|\mathbf{r}|_{\mathcal{G}_{\mathbf{r}}})$, and using the same comparison function

$$\nabla \phi_{\mathbf{r}}(\mathbf{r}) \cdot (\mathbf{f} + \mathbf{v}) \le -\xi(|\mathbf{r}|_{\mathcal{G}_{\mathbf{r}}}) + c_{\mathbf{sr}}|\mathbf{s}|, \qquad (6.17)$$

establishing that $\phi_{\mathbf{r}}(\mathbf{r})$ is an ISS-Lyapunov function for the system (6.6) as per requirement (2a).

6.4.1.1 Point attractor reference system

One of the reference systems we study below models a *flowbox*; a region of a vector field that is constant, or nearly so, by virtue of being en-route to a distant point attractor. We take as our attractor the point $\mathbf{r}_0 := [1000, 0]^T$, and define our Lyapunov function $\phi_{\mathbf{r}}(\mathbf{r}) := \mu((\mathbf{r} - \mathbf{r}_0)^T \begin{bmatrix} 1 & 0\\ 0 & 0.01 \end{bmatrix} (\mathbf{r} - \mathbf{r}_0))$. Our reference system is chosen to be $\mathbf{f}(\mathbf{r}) := \nabla \phi_{\mathbf{r}}(\mathbf{r})$, giving what is effectively a flow-box in the region $\mathbf{r}_x < 0$. Our choice of $\phi_{\mathbf{r}}$ is asymptotically linear in $|\mathbf{r}|_{\mathcal{G}_{\mathbf{r}}},$ causing \mathbf{f} to satisfy requirement $(\mathbf{2c})$.

6.4.1.2 Saturated Hopf oscillator reference system

The second reference system we examine models recurrent tasks, which may encounter a persistent disturbance multiple times. This reference is defined in terms of $\phi_{\mathbf{r}}(\mathbf{r}) := (|\mathbf{r}|^2 - R_0^2)^2/\mu(|\mathbf{r}|^3)$, with the constant α of the saturation function set so that $\alpha^3 \gg R_0^4$, and thus the dynamics near the R_0 radius disc are close to those of the unsaturated system, while the linear asymptotic growth ensures that $\nabla \phi_{\mathbf{r}}(\mathbf{r})$ is bounded. The state space is $\mathcal{Q} := \mathbb{R}^2 - \{\mathbf{0}\}$, and the reference dynamics on this space are given by $\mathbf{f}(\mathbf{r}) := \nabla \phi_{\mathbf{r}}(\mathbf{r}) + \omega_0 \mathbf{J} \mathbf{r}/\mu(\mathbf{r}^T \mathbf{r})$, generating a constant angular rotation rate ω_0 in combination with the Hopf oscillator-like convergence to the circle at radius R_0 and also satisfying requirement (2c).

6.4.2 ISS Replanner

The replanning vector field \mathbf{g} is a stable focus [7]:

$$\mathbf{g}(\mathbf{s}) := -k_g \left(\mathbf{I} + w_s \mathbf{J}\right) \mathbf{s} \tag{6.18}$$

where scalar gain k_g adjusts the recovery rate from any perturbation on the transient dynamics, and the gain w_s adjusts the rate of rotation as expressed in the imaginary part of the eigenvalues. We excite maneuvers along the first coordinate of **s**, driven by the magnitude of tracking error

$$\mathbf{u}(\mathbf{e}) := \frac{|\mathbf{e}|}{\mu(\mathbf{e}^T \mathbf{e})} \begin{bmatrix} 1\\ 0 \end{bmatrix}$$
(6.19)

where $|\mathbf{u}| < 1$ since $\mu(\mathbf{e}^T \mathbf{e}) = \sqrt{|\mathbf{e}|^2 + \alpha^2} > |\mathbf{e}|$, providing requirement (2d).

Let $\phi_{\mathbf{s}}(\mathbf{s}) := \mu(\mathbf{s}^T \mathbf{s})$, giving $\nabla \phi_{\mathbf{s}}(\mathbf{s}) = \mathbf{s}/\mu(\mathbf{s}^T \mathbf{s})$. This gradient's norm monotonically grows

to 1 as $|\mathbf{s}|$ grows to infinity. The Lie derivative of $\phi_{\mathbf{s}}$ for the system (6.7) given by $\dot{\mathbf{s}} = \mathbf{g}(\mathbf{s}) + \mathbf{u}(\mathbf{e})$ is

$$\dot{\phi}_{\mathbf{s}}(\mathbf{s}, \mathbf{e}) = \frac{\mathbf{g}(\mathbf{s}) \cdot \mathbf{s} + \mathbf{u}(\mathbf{e}) \cdot \mathbf{s}}{\mu(\mathbf{s}^{T}\mathbf{s})}$$

$$= \frac{1}{\mu(|\mathbf{s}|^{2})} \left(-k_{g} |\mathbf{s}|^{2} + \frac{|\mathbf{e}|}{\mu(|\mathbf{e}|^{2})} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \mathbf{s} \right)$$

$$\leq -k_{g} \frac{|\mathbf{s}|^{2}}{\mu(|\mathbf{s}|^{2})} + \frac{|\mathbf{s}|}{\mu(|\mathbf{s}|^{2})} \frac{|\mathbf{e}|}{\mu(|\mathbf{e}|^{2})}$$

$$\leq -k_{g} |\mathbf{s}| + \frac{|\mathbf{e}|}{\mu(|\mathbf{e}|^{2})}$$

$$\leq -k_{g} |\mathbf{s}| + \frac{|\mathbf{e}|}{\alpha} \qquad (6.20)$$

From [132] eqn. (7), $\phi_{\mathbf{s}}$ is an ISS-Lyapunov function for the replanner, (6.7) is ISS with respect to attractor **0** and input **e**, and requirement (3) is satisfied³⁵.

6.4.3 Integral-ISS Tracking Error Dynamics

We implement a simple potential-dissipative [75, 78] tracking controller (in this case, a generalized spring-damper) with saturated terms, where $\phi_{\mathbf{e}}(\mathbf{e}) = k_{\mathbf{e}}\mu(\mathbf{e}^{T}\mathbf{e})$ and

$$E(\mathbf{e}, \dot{\mathbf{e}}) := -\nabla \phi_{\mathbf{e}}(\mathbf{e}) - \frac{m_{\mathbf{e}}}{\mu(\dot{\mathbf{e}}^T \dot{\mathbf{e}})} \dot{\mathbf{e}}$$
(6.21)

Proposition 6.4.1. The system

$$\ddot{\mathbf{e}} = E(\mathbf{e}, \dot{\mathbf{e}}) = -\nabla\phi_{\mathbf{e}}(\mathbf{e}) - \frac{m_{\mathbf{e}}}{\mu(\dot{\mathbf{e}}^T \dot{\mathbf{e}})}\dot{\mathbf{e}}$$
(6.22)

is GAS.

 $^{^{35}}$ Note that we have formulated our theory allowing the replanner's attractor to be a general compact set, rather than zero; this allows for *memory*; the maneuver state within the attractor can persist between excitations.

Proof. Consider the function

$$\eta_{\mathbf{e}} := \phi_{\mathbf{e}}(\mathbf{e}) + \frac{1}{2} \dot{\mathbf{e}}^T \dot{\mathbf{e}}$$
(6.23)

whose Lie derivative under (6.22) satisfies,

$$\begin{split} \dot{\eta}_{\mathbf{e}} &= \nabla \phi_{\mathbf{e}}(\mathbf{e}) \cdot \dot{\mathbf{e}} + \dot{\mathbf{e}} \cdot \ddot{\mathbf{e}} \\ &= \dot{\mathbf{e}} \cdot \left(\nabla \phi_{\mathbf{e}}(\mathbf{e}) - \nabla \phi_{\mathbf{e}}(\mathbf{e}) - \frac{m_{\mathbf{e}}}{\mu(|\dot{\mathbf{e}}|^2)} \dot{\mathbf{e}} \right) \\ &= -\frac{m_{\mathbf{e}}}{\mu(\dot{\mathbf{e}}^T \dot{\mathbf{e}})} |\dot{\mathbf{e}}|^2 \\ &\leq 0. \end{split}$$

For $\dot{\mathbf{e}} = \mathbf{0}$ we note that $\ddot{\mathbf{e}} = -\nabla \phi_{\mathbf{e}}(\mathbf{e})$, and thus $(\ddot{\mathbf{e}} \cdot \mathbf{e}) |_{\dot{\mathbf{e}}=\mathbf{0}} < 0$, satisfying LaSalle's condition and therefore ensuring that $\eta_{\mathbf{e}} \to 0$.

Proposition 6.4.2. $\ddot{\mathbf{e}} = E(\mathbf{e}, \dot{\mathbf{e}}) + \mathbf{d}$ is iISS with respect to the attractor 0 and the input \mathbf{d} .

Proof. We show that $\eta_{\mathbf{e}}$ of (6.23) satisfies the conditions of an iISS storage function, as per [3] equation (11).

$$\dot{\eta}_{\mathbf{e}} = \frac{m_{\mathbf{e}}}{\mu(\dot{\mathbf{e}}^T \dot{\mathbf{e}})} \left(\dot{\mathbf{e}} \cdot \mathbf{d} - \dot{\mathbf{e}}^T \dot{\mathbf{e}} \right) < \frac{m_{\mathbf{e}}}{\mu(\dot{\mathbf{e}}^T \dot{\mathbf{e}})} \left| \dot{\mathbf{e}} \right| \left| \mathbf{d} \right|$$
(6.24)

By construction $\mu(x^2) > \max(x, \alpha)$, and thus $M := \sup\{x/\mu(x^2)|x \in \mathbb{R}_{>0}\}$ is finite. We may choose for [3] equation (11) to have $\sigma(|\mathbf{d}|) := M |\mathbf{d}|$. As we have already shown 0-GAS, the requirements of [3] theorem 1 case 4 are met, satisfying our design requirement (4). \Box

6.4.4 Disturbance

We model persistent disturbances by taking $\mathbf{d} := \nabla h(q)$ for a scalar function h defined in terms of manually placed square tiles. This height-like disturbance potential h will be referred to as the *terrain*, although the magnitudes were chosen such that the terrain obstacles could not be surmounted with the force available to the controller. Each tile is endowed with a cubic mapping height from a displacement measured either radially from a corner or as a Cartesian distance from one of the edges of the tile. This collection of tiles allows the construction of C^2 smooth terrains, by appropriate selection of neighboring tiles; all simulated terrains are smooth.

6.5 Simulation Studies

6.5.1 Simulations and Quality Metrics

The controller architecture we propose lies on a continuum determined by the coupling gain $c_{\mathbf{sr}}$ of (6.16), at one end of which $c_{\mathbf{sr}} = 0$ and the system simplifies to a classical trajectory tracker with the trajectory starting at $\mathbf{r}(0)$ as its reference. As $c_{\mathbf{sr}}$ grows, maneuvers induced in t have larger effects on \mathbf{r} . We demonstrate that for our example systems, an interval of $c_{\mathbf{sr}}$ values provides noticeably better system performance by several quality metrics: (1) tracking quality as represented by the Lyapunov function $\eta_{total} := \eta_{\mathbf{r}} + \eta_{\mathbf{e}} + \phi_{\mathbf{s}}$; (2) reference convergence as represented by the Lyapunov function of the reference $\phi_{\mathbf{r}}$; (3) power expenditure as expressed by the integral $E_{total} := \int |\mathbf{\ddot{q}} \cdot \mathbf{\dot{q}}| dt$. For the point attractor example, this integral is taken until the state variable \mathbf{q}_x lies to the right of the terrain obstacle. For the Hopf examples, the integral is normalized by dividing by the number of rotations around the origin.

As can be observed in the accompanying figures, the proposed architecture results in considerable perturbation away from the trajectories of the undisturbed reference generator. Indeed, as discussed above, these deformations cannot be claimed optimal in any sense. Rather, they are feasible courses of action that respect the plant's power and energy limitations. All simulations were integrated using code derived from the dopri5 code from [49], with the output interfaced to the SciPy open-source scientific Python environment³⁶.

6.5.1.1 Point Attractor with comb obstacle

The first example shows the interaction of our controller with a *comb* obstacle punctuated by regularly spaced cul-de-sac traps (Figure 6.1), and demonstrates how c_{sr} relates performance to the geometry of obstacles. Success at this task constitutes reaching a state with \mathbf{q}_x to the right of the obstacle. The change in total energy consumption with varying c_{sr} is presented in Figure 6.2, and shows that while the interval $3.9 < c_{sr} < 18.0$ provides good performance, at larger values repeated resonance-like bands of degraded performance appear (e.g. at $c_{sr} = 99.0$). Apparently these bands correspond to maneuver spatial scales that take the state out of one trap into another.



Figure 6.1: Plant evolution for the point attractor reference meeting a comb obstacle with different transient to reference (c_{sr}) coupling gains((red) plant, (black) reference). (a) For a small value of c_{sr} the particle remains blocked by the obstacle, (b) For a moderate c_{sr} value plant escapes with very low costs, (c),(d) For higher values of c_{sr} energy cost grows again with resonance peaks when the replanner induces escape maneuvers whose spatial frequencies couple strongly to the geometric features of the particular obstacle.



Figure 6.2: Energy consumed over the course of the point attractor reference with comb obstacle depicted in Fig. 1 as a function of the transient to reference coupling gain, $c_{\rm sr}$ (of (6.16)). (a) Magnified view of small values of $c_{\rm sr}$; (b) Larger values of $c_{\rm sr}$ showing optimum, an approximately linear increase in cost with increased $c_{\rm sr}$, and occasional resonance peaks where cost is larger over a narrow range.

6.5.1.2 Hopf reference with two obstacles

In these two examples, the task encoded by the reference system continually brings the plant back into interaction with an obstacle that blocks the limit cycle, and includes a trap that would completely block a simple reference tracker.

For both the simple obstacle A (Figure 6.3) and the more elaborate obstacle B (Figure 6.4), our controller manages to escape the traps. For a range of $c_{\rm sr}$ values, the system then exhibits a modified cycle which accomplishes the task with moderate energy consumption (Figure 6.5 obstacle A; Figure 6.6 obstacle B).

 $^{^{36}}$ Scientific Tools for Python, www.scipy.org. Using our code this provided an extremely fast ODE integrator. In our tests it gave $1.02 \cdot 10^6$ trajectory points a second of a Rossler system's chaotic orbit on a single core of an Intel i5 CPU at 2.67 GHz — an order of magnitude faster than the commonly used MatLab ode45 integrator on the same platform.



Figure 6.3: Plant evolution for the Hopf cycle attractor reference meeting a simple obstacle with different transient to reference coupling gains ((red) plant, (black) reference). (a) Classical trajectory tracker (Zero or small c_{sr}) gets trapped until the reference sweeps back behind it – at which point it is pulled out and proceeds to cycle hitting the obstacle again at a different position, effectively trapped in place, (b) At a sufficiently large c_{sr} a qualitative change appears – the plant hits the obstacle exactly once every cycle and then back-tracks and circles the obstacle, achieving a deformation on the reference trajectory cycle, (c) At even larger c_{sr} this regular trajectory deforms more and more.

From the point of view of iISS theory, it should be noted that in these simulations $|\mathbf{d}|_2$ is unbounded since the obstacle interaction has support in every cycle. Thus we can not expect convergence to $\mathcal{G}_{\mathbf{r}}$, nor should we anticipate η_{total} and $\phi_{\mathbf{r}}$ to go to zero (see Figure 6.7).

6.6 Unicycle Extension

In this section, we describe a unicycle controller implementation on the RHex [42, 121] platform that loosely follows the methods presented in Section 6.4, specifically the point attractor replanner system. The major departure comes from the modeling decisions presented in Chapter 3: the robot is only equipped with local measurements about its environment, and thus, it can not facilitate an internal reference system. Instead, we seek to couple the internal transient system directly with the kinematic unicycle model. Since we have no reference to measure error against, we utilize the obstacle function gradient to excite the transient system.



Figure 6.4: Plant evolution for the Hopf cycle attractor reference meeting an elaborate obstacle with different transient to reference coupling gains ((red) plant, (black) reference). (a) The classical trajectory tracker (small or zero c_{sr}) gets trapped in the cul-de-sac as expected, and unlike the previous case, even though the reference trajectory gets behind the plant at each period, the plant can not leave the trap. (b),(c) At a sufficiently large c_{sr} a qualitative change appears whereby the initial hit excites a successful escape recovery trajectory which returns along the unblocked portion of the cycle to repeat the same pattern, cycle after cycle.

The resulting system is as follows:

$$\dot{\mathbf{q}} = B(\theta) \left[\mathbf{u}_{ku} + \frac{1}{\mu(\nabla \varphi^T \nabla \varphi)} \mathbf{J}(\nabla \varphi) \mathbf{s} \right]$$
(6.25)

$$\dot{\mathbf{s}} = -k_g \left(\mathbf{I} + w_s \mathbf{J}\right) \mathbf{s} + \frac{|\mathbf{d}|}{\mu(\mathbf{d}^T \mathbf{d})} \begin{bmatrix} 1\\ 0 \end{bmatrix}$$
(6.26)

with \mathbf{u}_{ku} as in (3.62), and $\mathbf{d} := \nu_{\psi} \sum_{i=1}^{d} \nabla \psi_i(\mathbf{p})$ combined obstacle function gradients as utilized in Section 3.2.1.2. To analyze whether the resulting system exhibits anything akin to an ISS or iISS system is beyond the scope of this thesis. There exists the notion of an almost-ISS system [4]. Unfortunately, as stated in Remark 3.2.12, we can't prove that the kinematic unicycle system is AGAS under (3.62), and thus, we can not follow up with the corresponding analysis.



Figure 6.5: Energy consumed over the course of the Hopf cycle attractor reference with simple obstacle depicted in Fig. 3 as a function of the transient to reference coupling gain. (a) Magnified view of small values of c_{sr} ; (b) Larger values of c_{sr} showing optimum, an approximately linear increase in cost with increased c_{sr} , and occasional resonance peaks where cost is larger over a narrow range.



Figure 6.6: Energy consumed over the course of the Hopf cycle attractor reference with elaborate obstacle depicted in Fig. 4 as a function of the transient to reference coupling gain. (a) Magnified view of small values of $c_{\rm sr}$; (b) Larger values of $c_{\rm sr}$ showing optimum, an approximately linear increase in cost with increased $c_{\rm sr}$, and occasional resonance peaks where cost is larger over a narrow range.



Figure 6.7: Contributions to total Lyapunov function η_{total} for one cycle of the Hopf system. The tracking error Lyapunov $\eta_{\mathbf{e}}$ (red) comprising potential (cyan) and kinetic terms grows rapidly when the obstacle is hit, causing a growth of the transient $\phi_{\mathbf{s}}$ ($\eta_{\mathbf{e}} + \phi_{\mathbf{s}}$ in green). The ISS Lyapunov function η_{total} (blue) continues to grow until the transient becomes sufficiently small, and then it too decays exponentially.

Chapter 7

A drift-diffusion model for robotic obstacle avoidance

In Chapter 6, we demonstrated a method that can be used for negotiating with more complex obstacles by utilizing an internal model capable of inferring and reacting to the presence of an unexpected obstacle by exciting special behaviors that promote escape. Unfortunately, the problem representation suitable to sound reasoning about the dynamical implications of these methods leaves a substantial gap with respect to the implications relating to knowledge about the geometric properties of the environment–most crucially, the obstacle loci and shape.

In this chapter, we present another extension to our task execution strategies previously published in [114], where we take the very earliest steps toward a fundamentally stochastic approach to reasoning about the interaction between such a system and the geometric properties of its state space that shows promise for meeting up usefully with the deterministic properties of the underlying dynamics. For now, as a first step toward a stochastically-enhanced version of the deterministic replanner [115], we simply replace it and introduce stochastic noise into the otherwise deterministic robot dynamics and reason about the statis-
tics of the resulting interaction with the uncertain local geometric environment. Unsurprisingly, this approach allows a more natural representation of that uncertainty. However, at the same time, less obviously, it invites a representation of the deterministic aspects of the obstacle avoidance control strategy in terms of boundary interaction models treated by a growing body of literature on stochastic differential equations (SDEs).

In Section 7.1 we present the problem statement motivating this work, where the robot and the task are modeled together as a stochastic dynamical system. In Section 7.2, we investigate the robot's interaction with a single obstacle under this formulation. In Section 7.3, we present a loose interpretation of this method on the RHex robot, where the control law of Section 3.2.2.1 is not tuned properly and the robot can be *trapped* by an undesired fixed point in front of a single obstacle. We document experimental results (Section 7.3.3), where the introduction of our approach not only improves the probability of avoiding this undesired fixed point from 50% to 100% but also reduces the average time the robot spends interacting with the obstacle.

7.1 Problem statement

The starting point for our framework is the navigation function method originally proposed in [73]. We model the robot as a point mass traveling in a domain $\mathcal{D} \subseteq \mathbb{R}^2$, so its configuration at time $t \in \mathbb{R}$ is given by $\mathbf{x}(t) \in \mathcal{D}$. The domain is cluttered with obstacles, which we model as closed curves in \mathcal{D} . We assume the existence of a navigation function $\phi : \mathcal{D} \to \mathbb{R}$, which is a differentiable function with a unique maximum. The navigation function encodes the robot's task, which is to find maxima of ϕ . The robot achieves its task if its trajectory $\mathbf{x}(t)$ obeys

$$\lim_{t \to +\infty} \mathbf{x}(t) = \operatorname*{argmax}_{\mathbf{x}} \phi(\mathbf{x}). \tag{7.1}$$

The robot carries out its task by climbing the spatial gradient $\nabla \phi$ of the task function ϕ ,

so its idealized dynamics are given by

$$\dot{\mathbf{x}} = u\nabla\phi, \ u \in \mathbb{R}_+,$$

where the quantity u controls the speed at which the robot climbs the gradient. However, there are disturbances to these idealized dynamics due to, e.g., issues measuring the gradient $\nabla \phi$, interactions with the environment, as well as disturbances introduced as part of the control scheme. Denote the coordinates on \mathcal{D} as $(x, y) = \mathbf{x}$. We model the disturbance in each coordinate as a Wiener process of strength $D(\mathbf{x}) \in \mathbb{R}_+$, and assume that the two processes are independent. The process noise intensity is the sum of two terms: $D(\mathbf{x}) =$ $D_a(\mathbf{x}) + D_c(\mathbf{x})$, where $D_a(\mathbf{x}) \in \mathbb{R}_+$ is the ambient noise due to the environment and $D_c(\mathbf{x}) \in$ \mathbb{R}_+ is the control noise added added as part of the control strategy.

The noise-corrupted dynamics are described by the following Itô stochastic differential equation (SDE)

$$\mathsf{D}\mathbf{x} = \begin{bmatrix} \mathsf{D}x\\ \mathsf{D}y \end{bmatrix} \mathsf{D}t = u \begin{bmatrix} \frac{\partial\phi}{\partial x}\\ \frac{\partial\phi}{\partial y} \end{bmatrix} \mathsf{D}t + D(\mathbf{x}) \begin{bmatrix} \mathsf{D}W_t\\ \mathsf{D}V_t \end{bmatrix},\tag{7.2}$$

where $D(\mathbf{x})$ is the strength of the disturbance at $\mathbf{x} \in \mathcal{D}$ and $\mathsf{D}W_t$ and $\mathsf{D}V_t$ are independent Wiener increments. Dependencies in the disturbances can be modeled by making $D(\mathbf{x})$ a positive-definite matrix-valued function of \mathbf{x} . Standard references for the SDE methods used in this work are [117] and [43].

Solving Equation (7.2) generates trajectories of a single particle. Solving the equation repeatedly from the same initial conditions generates different trajectories due to the stochastic nature of the dynamics. Alternatively, one can consider the probability distribution $p(\mathbf{x}, t)$ of the state $\mathbf{x}(t)$ as a function of time t. The probability distribution is a function that gives the probability of finding the robot in a set of states:

$$\Pr\left[\mathbf{x}(t) \in S\right] = \int_{S} p(\mathbf{x}, t) \mathsf{D}\mathbf{x},\tag{7.3}$$

where $S \subseteq \mathcal{D}$ is a subset of the state space.

The time evolution of the distribution p induced by the dynamics (7.2) is given by the following partial differential equation:

$$\frac{\partial p}{\partial t} = \frac{1}{2} \nabla \cdot (D(\mathbf{x}) D(\mathbf{x})^T \nabla p) - u \nabla \phi \cdot \nabla p.$$
(7.4)

Equation (7.4) is known as the *Fokker-Planck equation* [43, 117]. Equations of this form are studied in the literature on scalar transport phenomena under various names such as the advection-diffusion equation and the *drift-diffusion equation*.

The following physical analogy is illustrative. Consider a drop of dye in a fluid flow. The function $p(\mathbf{x}, t)$ measures the concentration of dye at the spatial location \mathbf{x} at time t. If the dye is initially concentrated at \mathbf{x}_0 , the initial condition for the equation (7.4) is the Dirac delta function $p(\mathbf{x}, t_0) = \delta(\mathbf{x} - \mathbf{x}_0)$. As time elapses, the dye moves with the fluid, which flows with local velocity $\nabla \phi(\mathbf{x})$ and diffuses with coefficient $D(\mathbf{x})$. Transport due to the local velocity is called *advection*, or *drift*, while the spreading due to the $D(\mathbf{x})$ term is called *diffusion*, and the two terms of the equation are referred to accordingly.

The equations (7.2) and (7.4) define a stochastic dynamical system where u is a control parameter. In future work, we will leverage tools from the stochastic geometry literature to derive ways to tune u such that the robot can navigate through a spatially-distributed obstacle field. A key next step to developing this theory will be the extension of our model to the case of multiple obstacles.

7.2 Single obstacle

In this section, we analyze the interaction of a particle obeying the stochastic dynamics (7.2) and a single obstacle, which we model as a closed curve in \mathcal{D} . We develop a set of assumptions under which we can calculate the probability of escaping a single obstacle in

closed form as a function of a single dimensionless parameter.

7.2.1 Assumptions

We make the following assumptions to develop analytical tools to study informative cases of the obstacle escape problem.

- (1) The coordinates are aligned with the local gradient ∇φ, such that ∂φ/∂x = 1 is a constant and ∂φ/∂y = 0. In other words, traveling in the +x direction is equivalent to climbing the (constant) local gradient.³⁷
- (2) The diffusion tensor $D(\mathbf{x})$ is diagonal and constant in \mathbf{x} : $D(\mathbf{x}) = D_i \delta_{ij}$.
- (3) Diffusion only acts in the dimension orthogonal to the gradient, so $D(\mathbf{x})$ has x component $D_1 = 0$ and y component $D_2 = D$.
- (4) The particle interacts with obstacles through specular reflection: if, prior to the interaction it has momentum \mathbf{p} , after the interaction it will have momentum $\mathbf{p}' = \mathbf{p} 2\hat{\mathbf{n}}(\hat{\mathbf{n}} \cdot \mathbf{p})$, where $\hat{\mathbf{n}}$ is the outward normal vector to the surface of the obstacle at the point of contact. This is equivalent to assuming that the obstacles have infinite mass and that the particle-obstacle interaction is an elastic collision.

Specular reflection constitutes one of the two canonical types of boundary conditions generally specified for stochastic differential equations (with absorption being the other [43]). More recent work, e.g., [126, 136], has considered more general boundary conditions that could model inelastic collisions with a coefficient of restitution $\epsilon \in (0, 1)$. However, the interpretation of these boundary conditions is more complicated and would require more detailed modeling of the physical obstacle interaction. Therefore, in this preliminary study,

³⁷This analytical simplification (guaranteed by the "flowbox" theorem of dynamical systems to be possible in the neighborhood of any non-singular orbit) would not have any impact on the actual physical implementation.

we adopt the abstract reflecting boundary condition as the most appropriate for developing analytical results with the particle model considered here.

With these assumptions, the dynamics (7.2) reduce to

$$\mathsf{D}\mathbf{x} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \mathsf{D}t = \begin{bmatrix} u \\ 0 \end{bmatrix} \mathsf{D}t + \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} dW_t \\ dV_t \end{bmatrix}.$$
 (7.5)

The drift-diffusion equation (7.4) induced by (7.5) is

$$\frac{\partial p}{\partial t} = \frac{D^2}{2} \frac{\partial^2 p}{\partial y^2} - u \frac{\partial p}{\partial x}.$$
(7.6)

We assume that the initial location of the particle is known with certainty to be $\mathbf{x}_0 = (x_0, y_0) \in \mathcal{D}$, so the initial condition for (7.6) is $p(\mathbf{x}, t = 0) = \delta(\mathbf{x} - \mathbf{x}_0)$. Finally, we assume that the speed u is constant.

In the absence of boundary conditions, the solution of (7.6) can be found in closed form, and is (cf. [117, (5.20)])

$$p(\mathbf{x},t) = \frac{1}{\sqrt{2\pi D^2 t}} \exp\left(-\frac{(y-y_0)^2}{2D^2 t}\right) \delta(x - (x_0 + ut)).$$
(7.7)

The solution can be interpreted as follows. The particle moves deterministically along the x coordinate with a constant velocity u and moves stochastically along the y coordinate according to a random walk. At time t, the particle distribution is Gaussian with center $(x_0 + ut, y_0)$ and standard deviation $D\sqrt{t}$.

For a given evolution time t the distribution has two characteristic lengths:

- (1) Advection length scale: ut
- (2) Diffusion length scale: $D\sqrt{t}$.

Their ratio, $ut/D\sqrt{t}$, is a form of the Péclet number [55], which is a dimensionless quantity

that measures the relative strength of advection and diffusion. This ratio is a function of evolution time t; if we specify an evolution time, we get a characteristic number that captures all the dimensional variables governing the dynamical behavior.

7.2.2 Probability of escaping a single obstacle

The dynamics (7.5) have a clear flow in the positive x direction. We take advantage of this behavior to characterize obstacles according to their geometry relative to the flow. Intuitively, the reflecting boundary condition specified in assumption 4) allows the particle to bounce off of obstacles and in some cases escape an obstacle by moving around it. However, a particle will clearly not be able to escape all obstacles in this fashion. Consider, for example, the crescent-shaped obstacle shown in Figure 7.1-B. If the advection term dominates in the dynamics (7.5), then the particle will tend to get trapped by the obstacle.

The examples in Figure 7.1 illustrate that the important characteristic of an obstacle in this framework is the convexity of its footprint with respect to the local advective flow. Loosely speaking, an obstacle is *convex* with respect to the flow (7.5) if the obstacle appears convex to an observer looking at it from the upstream direction. An obstacle *concave* with respect to the flow is defined analogously.



Figure 7.1: Example obstacles placed in the flow described by the dynamics (7.5). Panel A: an obstacle that is convex with respect to the flow, and will not trap a particle with D > 0. Panel B: An obstacle that is concave with respect to the flow, and may trap a particle regardless of the value of D.

The definition of obstacle convexity can be made more precise in the following way. Define a section Σ transverse to the flow upstream of the obstacle. Consider the noise-free dynamics, i.e., (7.5) with D = 0. For each point $\mathbf{x} \in \Sigma$, define $g(\mathbf{x})$ as the time at which the solution to the noise-free dynamics with initial condition \mathbf{x} first touches the obstacle. The convexity property of the obstacle can now be formally defined as being inherited from that of the time-to-impact function, g.

If a particle following the dynamics (7.5) with D = 0 evolves from an initial condition upstream of a concave obstacle, it will eventually hit the obstacle and remain close to the point of impact. Conversely, we define a particle to have escaped an obstacle if its trajectory passes downstream of the obstacle. On the basis of physical intuition and numerical experiments, we argue that a particle following (7.5) with D > 0 will escape convex obstacles with probability one. This statement follows from the asymptotic behavior of solutions of (7.6), but that degree of formal development lies beyond the scope at present.

In contrast to convex obstacles, concave obstacles can trap particles with positive probability. Therefore, we explore in somewhat greater detail the interplay between controlled drift and diffusion in the face of concavity. Figure 7.2 defines the quantities relevant to the interaction with a concave obstacle. The advection and diffusion length scales defined in the previous section appear, as well as two geometric length scales: d is the distance between the initial position of the particle and the front of the obstacle located downstream, and ℓ is the width of the concave section of the obstacle. Note that ℓ can be less than the width of the obstacle itself. For simplicity of exposition we assume that the obstacle has a mirror symmetry over the $y = y_0$ axis. The probability of escape thus computed is a lower bound for the probability associated with a non-symmetric obstacle with the same width ℓ .

The geometric length scales allow us to compute the probability that a particle obeying (7.5) will escape a given concave obstacle. We evolve the probability distribution (7.7) of the location of the particle until it reaches the front of the obstacle. This requires the particle to travel a distance d at a constant speed u, which takes time $t_d = d/u$. This sets

the evolution time for the advection and diffusion length scales. The particle's location follows a Gaussian distribution with mean y_0 and standard deviation $\sigma = D\sqrt{t_d} = D\sqrt{d/u}$.

The particle will move into the concave region of the obstacle and get trapped if it is at the edge of the concave region at time t_d , i.e., if its y coordinate is in the range $(-\ell/2, \ell/2)$. Since the particle's location is Gaussian distributed, the probability that y is in this range, and therefore that the particle will become trapped, can be calculated in closed form. This yields π , the probability that the particle will avoid the obstacle:

$$\pi = \Pr\left[\text{Avoid obstacle}\right] = 2\left(1 - \Phi\left(\frac{\text{Pe}}{2}\right)\right),\tag{7.8}$$

where $Pe = \sqrt{\frac{\ell^2 u}{D^2 d}} \ge 0$ is the Péclet number for the given interaction and $\Phi : \mathbb{R} \to [0, 1]$ is the cumulative distribution function for the standard normal (i.e., Gaussian) distribution.

Figure 7.3 compares the analytical avoidance probability (7.8) with the simulated avoidance probability computed from 100 numerical simulations of the particle interaction depicted in Figure 7.2. The two avoidance probabilities match well up to moderate values of the diffusion coefficient D; for large values of D, there is more of a discrepancy, but this is likely due to approximation effects in the simulation code.

7.2.3 Escape time

The primary objective in the single obstacle problem is escaping the obstacle, for which the probability of avoidance (7.8) gives a quantitative metric. Given that the particle escapes the obstacle, a secondary objective is to do so quickly. For this objective a quantitative metric is time to escape, which can also be analyzed in our stochastic framework.

Consider again an obstacle interaction with geometry as in Figure 7.2. Define the random variable T as the first time at which the particle passes beyond the face of the obstacle.



Figure 7.2: The geometry of interaction with a concave obstacle. There are three characteristic lengths involved: d, $D\sqrt{t_d}$, and ℓ . The particle starts at location \mathbf{x}_0 , which is at a distance d from the obstacle, and travels at a constant speed u. This defines the time to interaction t_d through the relationship $d = ut_d$. At the interaction time, the effects of diffusion mean the particle distribution has characteristic width $D\sqrt{t_d}$. When the particle interacts interacts with the obstacle, it will get trapped if its location falls inside the concave footprint, which has width ℓ .

That is,

$$T = \inf_{\tau \ge 0} \{ x(\tau) > 0 \},$$

where $x(\tau)$ is the *x* coordinate of the particle at time τ . The quantity *T* is a random variable because of the stochastic nature of the dynamics. In general, *T* can have a complicated distribution, which depends on the initial location of the particle. Let $T(\mathbf{x})$ represent the mean of *T* conditional on the initial location being equal to \mathbf{x} .

The function $T(\mathbf{x})$ (and the higher-order moments of T) can be computed using a partial differential equation that is closely related to the Fokker-Planck equation (7.4) [43, Section 6.6]. The partial differential equation can be solved analytically only in special cases,



Figure 7.3: Analytical vs. simulated obstacle avoidance probability for the concave obstacle depicted in Figure 7.2. The theoretic analytical probability is given by (7.8), while the simulated probability (with approximate 95% confidence interval) is computed as the empirical avoidance probability from 100 numerical simulations. The two quantities match well up to moderate values of the diffusion coefficient D.

corresponding to obstacles with simple geometries. In more general cases, it can be solved numerically using finite element methods. An alternative method for finding the distribution of T is direct simulation of individual trajectories. This method is completely general and can be thought of as a type of particle filter method. In the following, we use direct simulation to study escape probability and escape time.

Figure 7.4 shows the probability of escape π and mean escape time $T(\mathbf{x}_0)$ as a function of diffusion coefficient D for a particle obeying dynamics (7.5) interacting with a circular obstacle with the geometry depicted in Figure 7.5. This geometry can be considered a special case of the geometry depicted in Figure 7.2 with the length ℓ of the concave footprint being $\ell = 0$. As argued above, the details of the obstacle geometry outside the concave section of the footprint do not matter so long as they are convex with respect to the drift flow $\nabla \phi$.

When D = 0, the particle hits the obstacle at the point (x, y) = (0, 0) and reflects directly along the direction in which it came, thereby getting trapped with probability one. For D > 0, the particle eventually escapes the obstacle, though the time to escape can be arbitrarily long. The figure depicts probability of escape in less than 10 time units; for $D > 10^{-3}$, the probability of escape is effectively one. The time to escape, conditional on escaping in less than 10 time units, decreases with increasing D until it appears to reach an asymptotic value of approximately 2.5 for large D. The asymptotic value is similar to the value that would be seen if there were no obstacle and the particle were simply traversing the distance d.

7.2.4 Implications for control

The result (7.8) and the escape time shown in Figure 7.4 have direct implications for obstacle avoidance control for a mobile robot, for which the particle model serves as a control target. We assume that the robot can control its speed u and the amount of process noise in its dynamics D by manipulating D_c . When there is an obstacle, on-board sensors, e.g., a LIDAR unit, will provide the robot with estimates of the distance d to the obstacle and its width, which serves as an upper bound for ℓ . If the sensor is sufficiently precise, it may be able to classify the obstacle as convex or concave, and provide an estimate of ℓ in the latter case. If the obstacle is convex, any positive noise will guarantee that the robot escapes the obstacle with certainty, i.e., probability one.

If the obstacle is concave, (7.8) implies that there is a non-zero probability that the robot will get stuck. However, we can make π , the probability of avoiding the obstacle, take any value in (0, 1) by appropriately manipulating the control parameters u and D. This result provides a point of contact to recent work in the robotics literature that makes use of results from percolation theory, e.g., [68]. In this literature, the obstacle-strewn domain



Figure 7.4: Probability of escape (line with circles, right scale) and expected time to escape conditional on escaping (solid line, left scale) a circular obstacle of radius R = 5, as a function of diffusion coefficient D. For D = 0, the particle gets trapped with probability one, while for D greater than 10^{-3} , the probability of escape is effectively one. The drop in probability of escape for D greater than 1 is due to the finite time of simulation. The obstacle was centered at $\mathbf{x} = (0, 0)$ and the initial location of the particle was $\mathbf{x}_0 = (0, -20)$. The drift speed u was equal to 10. The dashed lines indicate one standard deviation above and below the mean expected time to escape. All quantities were computed based on 1,000 simulations for each set of parameter values.

is modeled as a lattice graph and the probability of avoiding an obstacle is represented in terms of probabilities associated with the edges and vertices of the graph. Percolation theory then provides tools to find conditions under which it is feasible to travel extended distances through the graph.

For an individual obstacle, there will be trade-offs between the control parameters because high avoidance probabilities are associated with small speeds u and large diffusion parameters D. Large values of D result in fast escape times, as seen in Figure 7.4. However, such large diffusion parameters result in large deviations from the desired flow along $\nabla \phi$. These



Figure 7.5: The geometry of interaction with a circular obstacle. This can be thought of as a case of the geometry in Figure 7.2 with $\ell = 0$, as explained in detail in Section 7.2.3. A trajectory of the particle dynamics (7.5) is said to escape from the obstacle if the trajectory crosses the plane x = 0 denoted by the dashed horizontal line.

deviations result in occasional large escape times T, which produce the dip in probability of escape and increased spread of escape time observed for D > 1. For a given interaction geometry, (7.8) shows that the two control parameters trade off in an inverse manner. This gives us a first step towards understanding the optimal way to trade off the parameters, which we intend to continue in future work.

7.3 Experimental results

Consider a particle interacting with a convex obstacle with geometry as in Figure 7.5. The qualitative prediction of the theory developed in the previous section is that in the noise-free case D = 0, the particle will get trapped behind the obstacle. This can be seen from

Equation (7.8): a convex obstacle corresponds to the limit $\ell \to 0$, while the noise-free case corresponds to $D \to 0$. For the case of noise-free motion with a convex obstacle, D goes to 0 more quickly than ℓ , so this case corresponds to a Péclet number $\text{Pe} \to +\infty$. In contrast, in the case with noise D > 0, the Péclet number obeys $\text{Pe} \to 0$ and Equation (7.8) predicts that the particle will eventually escape the obstacle. As shown in Figure 7.4, the theory also predicts that in this case the mean time to escape decreases sharply with increasing noise. In this section, we present results of robot experiments that corroborate these qualitative predictions.

7.3.1 Implementation on RHex

To verify the qualitative predictions of the theory developed above in the context of a physically interesting robot (rather than a more literal instantiation of the abstract point particle for which the theory and simulation are literally applicable), we implemented a version of the stochastic dynamics (7.5) on an X-RHex hexapedal robot [42]. The X-RHex robots have non-trivial dynamics [146] whose coarse horizontal plane motion in slow gaits (e.g., up to two body lengths per second) can be reasonably well approximated by a kinematic unicycle [88] and by a second order generalization of such nonholonomically constrained models when moving at higher speeds [33]. For purposes of this work, we used a gait slow enough to be usefully abstracted by the kinematic unicycle, and applied a variant of the controller in [88] whose anchoring relation [41] to the notional point-particle gradient dynamics posited in this work can be established [57].

However, because we are disinclined to allow our robot to actually collide and bounce off a physical obstacle, our point particle gradient controller is rather more complicated than the simple constant-flow-with-elastic-collision model underlying the theoretical results presented above. Rather, we posit that the modified navigation function controller [115] implemented in these experiments introduces local deterministic interactions with obstacles that would be better modeled by the case of a plastic collision — i.e., the case $\epsilon = 0$ in Section II-A, Assumption 4. Looking ahead to assessing the efficacy of more sophisticated local replanners [115], we are pursuing the analysis of the more general *scattering* collision models discussed in that section. However, these more sophisticated analyses all lie beyond the scope of the present state of this work. In sum, the discrepancies between our actual robot control strategy and the abstraction used to develop the theory of Section II preclude any likelihood that quantitative predictions from the stochastic theory could be directly comparable to these experimental results. However, as we now report, the qualitative predictions are encouragingly reflected in these early empirical trials.

The implementation used for the robot experiments follows an approach that was first introduced by Khatib [71] where the task to be executed is represented by an artificial potential field composed of an attractive pole representing the goal state and repelling regions representing the obstacles. An extension to this approach was developed by Borenstein and Koren [17] where the obstacles are represented by certainty grids which enables a temporal filtering approach to obstacle detection. An alternative approach introduced by Borenstein and Koren [18] stems from some limitations of the previous method and focuses on moving to empty regions rather than being repelled by obstacle regions. A previous implementation on the RHex platform [62] utilizes a similar approach. Our assumptions regarding obstacle shape and distribution let us disregard the limitations described by Borenstein and Koren and implement a simple local repelling field around obstacles where, with proper choice of control parameters, any spurious fixed points introduced to the force field are guaranteed to be unstable [57].

7.3.2 Experimental setup

In our experiments we used a circular obstacle in the geometry depicted in Figure Figure 7.5. The effective radius of the obstacle was approximately R = 0.75 m, and the initial location of the center of the robot was at $\mathbf{x}_0 = (-2.0, 0.0)$ m, which is equivalent to an initial distance d = 2.0 m. The gradient field $\nabla \phi$ was generated by placing a point attractor in the far distance directly in front of the robot's initial position, along with an immediate repeller located in the obstacle. The effective radius of the repeller was small, and is included in the effective radius of the obstacle. The resulting gradient field approximates the constant field assumed in the dynamics (7.5) to a degree of precision comparable to the other experimental uncertainties. Timing for runs was performed through manual control of logging software, which resulted in measurements of the time to escape that were accurate to within 1 s.

As defined above, the process noise D was modeled as the sum of two terms: $D = D_a + D_c$, where D_a was the ambient noise due to the environment and D_c was the control noise added as part of the control strategy. The ambient noise D_a is due to noise in the robot's perceptual systems and various control loops. We manipulated the control noise D_c to have two values: either $D_c = 0$ or $D_c = \sqrt{40} \approx 6.3 \text{ m} \cdot \text{s}^{-1/2}$. We did not directly measure nor manipulate the ambient noise D_a , but it can reasonably be assumed to have been small and constant across the series of experiments. Importantly, the experimental results presented below imply that D_a was non-zero.

7.3.3 Results

The experiments consist of a number of obstacle interactions for the two control noise cases: the noise-free case $D_c = 0$ and the noisy case $D_c = \sqrt{40}$. For these first efforts, we focused exclusively on the single circular convex obstacle, rather than a family of obstacles including both convex and concave examples; such a family will be the subject of future work. Again, the noise values are not directly comparable to the diffusion coefficient D defined in Section II because of the details of the control strategy used on the robot. The results presented in Table 7.1 show, however, that the experiments match the qualitative trend predicted in Figure 7.4: adding control noise results in a higher probability of escaping the obstacle and a shorter mean time to escape for those runs that do escape.

In the noise-free case where $D_c = 0, 50\%$ of the runs resulted in the robot escaping the

	Noise-free, $D_c = 0$	Noisy, $D_c > 0$
	N = 8 runs	N = 10 runs
Probability of escape	50%	100%
Mean time to escape	$45.08 \mathrm{\ s}$	$8.860 \mathrm{\ s}$
Standard deviation	$13.94~\mathrm{s}$	$0.5393~{\rm s}$

Table 7.1: Experimental results. The noisy control strategy results in avoiding the obstacle much more quickly and with significantly higher probability.

obstacle. In view of the results presented in Figure 7.4, this implies that the ambient noise D_a is small, resulting in an overall noise $D = D_a$ that is comparable to the value of 10^{-5} that one can interpolate from Figure 7.4. Adding noise ensures that 100% of the empirical runs resulted in the robot escaping the obstacle. This corresponds to pushing the system into the regime on the right hand side of Figure 7.4. The other benefit of the noise can be seen in the mean time to escape: adding noise results in decreasing the mean time to escape by a factor of approximately five. This represents a substantial increase in obstacle avoidance performance.

Clearly the theory does not account for all of the empirical trends: for example, the empirical standard deviation of time to escape decreases with increasing noise, while the simulations based on the particle model presented in Figure 7.4 show a standard deviation that is increasing with increasing noise intensity. The intuition behind this trend is as follows. In the model, when a particle interacts with an obstacle, it can be reflected into the direction opposite the desired direction of motion. When this occurs, the particle takes longer to escape the obstacle. The control noise injects momentum into the particle, so larger noise intensities result in more energetic reflecting interactions with the obstacle and larger standard deviations of the escape time. This energetic reflecting behavior does not occur in the physical experiment, showing a limitation of the reflecting boundary condition model. More detailed modeling of the robot-obstacle interaction is required to better match theory and experiment which will be a subject of future work.

Chapter 8

Conclusions

In this thesis, we have explored methodologies for executing autonomous behaviors on a legged robot. For both behaviors demonstrated in our work, we started with simplified models of the robot's surroundings, its perceptual capabilities and restrictions on its mobility. Task encoding, behavior development, and the physical implementation closely followed these modeling decisions, which resulted in the empirical successes we have reported throughout this dissertation.

We have presented the autonomous hill ascent behavior, whose empirically demonstrated robustness in unstructured natural environments rests upon simple physical, sensory, and environmental models, combined with the underlying motor competencies of the host platform. Linear superposition of vestibular-sensed hill gradients and body-centric exteroceptive sensed obstacle gradients yields convergence and obstacle avoidance guarantees in simple environments—ones punctured by sufficiently sparse convex obstacles—for fully actuated point particle agents. Appropriately extended models enable control strategies that correctly embed this construction into kinematic and dynamic unicycle agents. Implementation of these sensorimotor schemes on a legged physical platform achieves highly successful autonomous hill-climbing performance at both walking and jogging speeds. The comparison studies between the two unicycle models presented in Section 4.2.3 did not provide enough evidence in favor of one model over the other. This lack of clarity likely stems from the limited sample size for the various scenarios. Roughly speaking, the trends emerging in our analysis agree with our intuition that the kinematic unicycle agent is more suitable for slow-pace behavior, whereas the dynamic unicycle agent is a better fit for fastpace behavior. Considering that the control policy for the dynamic unicycle agent is simply a low-pass-filtered version of the kinematic unicycle agent, if we had to choose one approach over the other, we would recommend the dynamic unicycle agent (3.71) combined with the control law (3.72). This choice can closely approximate the kinematic unicycle agent (3.59) equipped with the control law (3.62) via a steep increase in the dynamic unicycle control gain, $\nu_{\mathbf{r}}$.

We have also presented the autonomous stairwell ascent behavior, a rudimentary form of guarded autonomous locomotion. Its empirically demonstrated robustness in unstructured synthetic environments rests upon the underlying motor competencies of the host platform, stitched together with very simple perceptually triggered switches. The behavior implementation is arranged in a manner idealized by the formal notion of sequential composition [24].

We are convinced that a number of readily available extensions and improvements to the controllers presented in Chapter 3 would still further raise the level of practical autonomy suggested by the experiments in Chapter 4, thereby conferring still greater applications-worthy utility upon the RHex platform. We could use hill slope angle measurements to cue a greater diversity of better hill climbing gaits [146] in order to climb hills as steep as 45° . To better interact with obstacles that are not detectable by the current sensory implementation, the robot could rely on its legs to *feel* such disturbances [63] and temporarily modify its control policies, as in Chapter 6, if they persist. Similar deformations could be utilized to circumvent detectable obstacles violating our simple world model, an elementary version of which is presented in Chapter 7.

We believe that several further modest extensions and improvements to the execution of

the autonomous stairwell ascent task would considerably close the gap to full autonomy still revealed by the tables, thereby conferring true applications-worthy utility upon the X-RHex platform. The stair climbing behavior can be endowed with descent capability (as in [52] via [25]), as well as more reactive obstacle avoidance (as in Chapter 4). We also suspect this behavior could be completed using no exteroceptive sensors at all [63]. Instead the robot would rely on proprioceptive sensors and use the legs to *feel* obstacles. We could use a *virtual contact sensor* to feel the walls and a *missing ground* sensor as a cliff detector. Lastly, our approach to task encoding and execution could be combined with the perceptual capabilities developed in [148] for faster exploration of multiple floor buildings.

More broadly, as discussed in Section 1.4, while we only address the set stabilization problem here, the horizontal unicycle control policies presented in Section 3.2.2.1 and Section 3.2.2.2 are not limited to the problem of physical terrain ascent. This greedy, reactive methodology extends to trajectory tracking and path following settings by utilizing the reference tracking formulation of Chapter 6 or, for abstract quadratic hills, the more general reference governor approaches to reactive navigation of [9]. Finally, these methods can be utilized as a baseline for more complex (not purely uphill and respecting other objectives) behavior planning, as in [64].

This dissertation focuses on real-time control strategies for reactive motion planning. On the one hand, our controllers rely on the presumed mechanical capabilities of the platform to relieve many detailed real-time responsibilities. On the other hand, they rely on a naïve world model as a means of delimiting the navigational competence of their resulting closed loop behaviors. Of course, such motion planning strategies merely postpone—but, by dint of their known conservative guarantees, we contend, can simplify—the role of deliberative *task planning* on which we now speculate.

As a concrete setting, consider a geosciences robot field assistant that could help study the process of desertification [112], hill-slope erosion [21], or the structural geology of fault zones [98]. Researchers in these areas require repeated measurements of both ground and atmospheric quantities to be made in a manner largely determined by the topography (crests and troughs, windward and lee faces of dunes, etc) of the local terrain. We posit an aggressively hierarchical deliberative layer (i.e. featuring lazy execution at the task level) [66] whose semantics entail topographic features only in so far as they relate to the underlying scientific hypotheses (e.g., as in [26]). The affordances of this layer in the workspace require exactly the goal states of physically embodied primitives (such as those presented here) that handle the myriad of topographical features from the physical environment that are *not* related to the task semantics. Deliberation succeeds only because of the primitives' formal guarantees (here, ridge ascent), predicated upon conservative guard conditions (here, sparse and convex obstacles). These conditions can then be explicitly reasoned about (by high level postponement or recursive refinement [66]), e.g., here, perhaps via a geometry engine to subdivide or merge those troubling but otherwise uninteresting obstacles encountered at real-time execution.

Thus, the sufficient conditions we have established to achieve the task at hand (that we speculate, as in [8] and [109], may be close to necessary) can provide a valuable primitive for deliberative navigation task planners by handing off an otherwise overwhelming set of detailed and task-irrelevant responsibilities to lower motion planning layers. Considering the increased complexity (mission time span, disparity between physical and sensory capabilities, and environmental factors) in deployment scenarios for mobile autonomous systems, we speculate that passing some of these responsibilities to lower layers of control will become crucial for the overall success of the mission.

Bibliography

- Harold Abelson, Gerald Jay Sussman, and Julie Sussman. Structure and interpretation of computer programs. MIT Press, 1996.
- [2] Michele Aicardi, Giuseppe Casalino, Antonio Bicchi, and Aldo Balestrino. Closed loop steering of unicycle like vehicles via lyapunov techniques. *IEEE Robotics & Automation Magazine*, 2(1):27–35, 1995.
- [3] D. Angeli, E. D. Sontag, and Y. Wang. A characterization of integral input-to-state stability. *IEEE Transactions on Automatic Control*, 45(6):1082–1097, 2000.
- [4] David Angeli. An almost global notion of input-to-state stability. *IEEE Transactions* on Automatic Control, 49(6):866–874, 2004.
- [5] David Angeli and Denis Efimov. On input-to-state stability with respect to decomposable invariant sets. In *IEEE 52nd Annual Conference on Decision and Control* (CDC), pages 5897–5902. IEEE, 2013.
- [6] Ronald C Arkin and Warren F Gardner. Reactive inclinometer-based mobile robot navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 936–941. IEEE, 1990.
- [7] V. I. Arnold. Ordinary Differential Equations. MIT Press, 1973.
- [8] Omur Arslan and Daniel E. Koditschek. Sensor-based reactive navigation in convex sphere worlds. In *The 12th International Workshop on Algorithmic Foundations of Robotics (WAFR)*, December 2016.
- [9] Omur Arslan and Daniel E. Koditschek. Smooth extensions of feedback motion planners via reference governors. In *IEEE International Conference on Robotics and Au*tomation (ICRA), pages 4414–4421. IEEE, May 2017.
- [10] Nora Ayanian, Vinutha Kallem, and Vijay Kumar. Synthesis of feedback controllers for multiple aerial robots with geometric constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3126–3131. IEEE, 2011.
- [11] Abraham Bachrach, Samuel Prentice, Ruijie He, and Nicholas Roy. RANGE Robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*, 28(5):644–666, 2011.

- [12] Max Bajracharya, Andrew Howard, Larry H Matthies, Benyang Tang, and Michael Turmon. Autonomous off-road navigation with end-to-end learning for the LAGR program. Journal of Field Robotics, 26(1):3–25, 2009.
- [13] R Barbero, JT Abatzoglou, NK Larkin, CA Kolden, and B Stocks. Climate change presents increased potential for very large fires in the contiguous united states. *International Journal of Wildland Fire*, 24(7):892–899, 2015.
- [14] Pinhas Ben-Tzvi, Shingo Ito, and Andrew A Goldenberg. A mobile robot with autonomous climbing and descending of stairs. *Robotica*, 27(2):171–188, 2009.
- [15] Andreas Birk and Stefano Carpin. Rescue robotics A crucial milestone on the road to autonomous systems. Advanced Robotics, 20(5):595–605, 2006.
- [16] F. Blanchini and S. Miani. Constrained stabilization via smooth lyapunov functions. Systems & control letters, 35(3):155âĂŞ–163, 1998.
- [17] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. IEEE Transactions on Systems, Man and Cybernetics, 19(5):1179–1187, 1989.
- [18] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.
- [19] Ian E Brown and Gerald E Loeb. A reductionist approach to creating and using neuromusculoskeletal models. In Jack M. Winters and Patrick E. Crago, editors, *Biomechanics and neural control of posture and movement*, pages 148–163. Springer, 2000.
- [20] Robert G Brown and Richard F Meyer. The fundamental theorem of exponential smoothing. Operations Research, 9(5):673–685, 1961.
- [21] Rorke B Bryan. Soil erodibility and processes of water erosion on hillslope. Geomorphology, 32(3):385–415, 2000.
- [22] M. Buehler, D. E. Koditschek, and P. J. Kindlmann. Planning and control of robotic juggling and catching tasks. *International Journal of Robotics Research*, 13(12):101– 118, April 1994.
- [23] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. The 2005 DARPA grand challenge: the great robot race, volume 36. Springer Science & Business Media, 2007.
- [24] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.
- [25] D Campbell and Martin Buehler. Stair descent in the simple hexapod'rhex'. In IEEE International Conference on Robotics and Automation (ICRA), volume 1, pages 1380– 1385. IEEE, 2003.

- [26] Alberto Candela, David Thompson, Eldar Noe Dobrea, and David Wettergreen. Planetary robotic exploration driven by science hypotheses for geologic mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, September 2017.
- [27] Claudio Caraffi, Stefano Cattani, and Paolo Grisleri. Off-road path and obstacle detection using decision networks and stereo vision. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):607–618, 2007.
- [28] Fernando Auat Cheein and Gustavo Scaglia. Trajectory tracking controller design for unmanned vehicles: a new methodology. *Journal of Field Robotics*, 31(6):861–887, 2014.
- [29] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. *Principles of robot motion: theory, algorithms, and implementations*. MIT Press, 2005.
- [30] Andrei Ciobanu, Anca Morar, Florica Moldoveanu, Lucian Petrescu, Oana Ferche, and Alin Moldoveanu. Real-time indoor staircase detection on mobile devices. In 21st International Conference on Control Systems and Computer Science (CSCS), pages 287–293. IEEE, 2017.
- [31] Jennie Cochran and Miroslav Krstic. Nonholonomic source seeking with tuning of angular velocity. *IEEE Transactions on Automatic Control*, 54(4):717–731, 2009.
- [32] D. C. Conner, H. Choset, and A. A Rizzi. Flow-through policies for hybrid controller synthesis applied to fully actuated systems. *IEEE Transaction on Robotics*, 25(1):136– 146, 2009.
- [33] A. De, K. S. Bayer, and D. E. Koditschek. Active sensing for dynamic, non-holonomic, robust visual servoing. In *IEEE International Conference on Robotics and Automation* (*ICRA*), pages 6192–6198. IEEE, May 2014.
- [34] A. De, G. Lynch, A. Johnson, and D. Koditschek. Motor sizing for legged robots using dynamic task specification. In *IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, pages 64–69, 2011.
- [35] Avik De and Daniel E. Koditschek. Toward dynamical sensor management for reactive wall following. In *IEEE International Conference on Robotics and Automation* (*ICRA*), pages 2400–2406. IEEE, 2013.
- [36] J. A. Delmerico, D. Baran, P. David, J. Ryde, and J. J. Corso. Ascending stairway modeling from dense depth imagery for traversability analysis. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2283–2290, May 2013.
- [37] Jaydev P Desai, James P Ostrowski, and Vijay Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Au*tomation, 17(6):905–908, 2001.

- [38] Dimos V Dimarogonas and Kostas J Kyriakopoulos. On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on Automatic Control*, 52(5):916– 922, 2007.
- [39] Michael D Flannigan, Brian J Stocks, and B Mike Wotton. Climate change and forest fires. Science of the total environment, 262(3):221–229, 2000.
- [40] Gustavo Freitas, Bradley Hamner, Marcel Bergerman, and Sanjiv Singh. A practical obstacle detection system for autonomous orchard vehicles. In *IEEE/RSJ Interna*tional Conference on Intelligent Robots and Systems (IROS), pages 3391–3398. IEEE, 2012.
- [41] Robert J Full and Daniel E Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202(23):3325– 3332, 1999.
- [42] Kevin C. Galloway, G. C. Haynes, B. Deniz Ilhan, Aaron M. Johnson, Ryan Knopf, Goran Lynch, Benjamin Plotnick, Mackenzie White, and D. E. Koditschek. X-RHex: A highly mobile hexapedal robot for sensorimotor tasks. Technical report, University of Pennsylvania, 2010.
- [43] C. Gardiner. Stochastic Methods: A Handbook for the Natural and Social Sciences, volume 13 of Springer Series in Synergetics. Springer, 4th edition, 2009.
- [44] José Gaspar, Niall Winters, and José Santos-Victor. Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Transactions* on Robotics and Automation, 16(6):890–898, 2000.
- [45] E. Gilbert and I. Kolmanovsky. Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor. *Automatica*, 38(12):2063âĂŞ– 2073, 2002.
- [46] Georges Giralt, Ralph Sobek, and Raja Chatila. A multi-level planning and navigation system for a mobile robot: a first approach to Hilare. In *The 6th International Joint Conference on Artificial intelligence*, volume 1, pages 335–337. Morgan Kaufmann Publishers Inc., 1979.
- [47] Daniel Goldman, Haldun Komsuoglu, and Daniel Koditschek. March of the sandbots: Legged robots are striding more successfully than ever with the help of some fleetfooted desert natives. *IEEE Spectrum*, 46(4):26–26, 2009.
- [48] J Guckenheimer and P Holmes. Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields. Springer-Verlag, 1983.
- [49] E. Hairer, S. P. Nœrsett, and G. Wanner. Solving Ordinary Differential Equations I, Nonstiff Problems, volume publisher of Springer Series in Computational Mathematics. Springer, second edition, 2008.
- [50] G. C. Haynes and D. E. Koditschek. On the comparative analysis of locomotory systems with vertical travel. In *International Symposium on Experimental Robotics* (*ISER*), Delhi, India, December 2010.

- [51] G. Clark Haynes and Alfred A. Rizzi. Gaits and gait transitions for legged robots. In *IEEE International Conference On Robotics and Automation (ICRA)*, pages 1117– 1122, Orlando, FL, USA, May 2006.
- [52] J.A. Hesch, G.L. Mariottini, and S.I. Roumeliotis. Descending-stair detection, approach, and traversal with an autonomous tracked vehicle. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5525–5531. IEEE, 2010.
- [53] Philip Holmes, Robert J Full, Daniel E Koditschek, and John Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Review*, 48(2):207–304, 2006.
- [54] Roger A Horn and Charles R Johnson. Matrix analysis. Cambridge University Press, 1985.
- [55] M. Huysmans and A. Dassargues. Review of the use of Péclet numbers to determine the relative importance of advection and diffusion in low permeability environments. *Hydrogeology Journal*, 13(5-6):895–904, 2005.
- [56] B. D. Ilhan, O. Erkent, and H. I. Bozma. Saccades and fixating using artificial potential functions. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5819–5824, 2006.
- [57] B. Deniz Ilhan, Aaron M. Johnson, and Daniel E. Koditschek. Autonomous legged hill ascent. *Journal of Field Robotics*, 2018 (in press).
- [58] B. Deniz Ilhan, Aaron M. Johnson, and Daniel E. Koditschek. Autonomous stairwell ascent. *Robotica*, 2018 (under review).
- [59] Félix Ingrand and Malik Ghallab. Robotics and artificial intelligence: A perspective on deliberation functions. AI Communications, 27(1):63–80, 2014.
- [60] Larry D Jackel, Eric Krotkov, Michael Perschbacher, Jim Pippine, and Chad Sullivan. The DARPA LAGR program: Goals, challenges, methodology, and phase I results. *Journal of Field robotics*, 23(11–12):945–973, 2006.
- [61] Zhong-Ping Jiang. Robust exponential regulation of nonholonomic systems with uncertainties. Automatica, 36(2):189–209, 2000.
- [62] Aaron M Johnson, Matthew T Hale, GC Haynes, and Daniel E Koditschek. Autonomous legged hill and stairwell ascent. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 134–142. IEEE, 2011.
- [63] Aaron M. Johnson, G. Clark Haynes, and D. E. Koditschek. Disturbance detection, identification, and recovery by gait transition in legged robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5347–5353, Taipei, Taiwan, October 2010. IEEE.
- [64] Aaron M. Johnson, Jennifer E. King, and Siddhartha Srinivasa. Convergent planning. *IEEE Robotics and Automation Letters*, 1(2):1044–1051, 2016.

- [65] Aaron M. Johnson and D. E. Koditschek. Legged self-manipulation. IEEE Access, 1:310–334, May 2013.
- [66] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *IEEE International Conference on Robotics and Automation* (*ICRA*), pages 1470–1477. IEEE, 2011.
- [67] Yutaka Kanayama, Yoshihiko Kimura, Fumio Miyazaki, and Tetsuo Noguchi. A stable tracking control method for an autonomous mobile robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 384–389. IEEE, 1990.
- [68] S. Karaman and E. Frazzoli. High-speed flight in an ergodic forest. In *IEEE Inter*national Conference on Robotics and Automation (ICRA), pages 2899–2906. IEEE, 2012.
- [69] Alonzo Kelly and Anthony Stentz. Rough terrain autonomous mobility-part 2: An active vision, predictive control approach. *Autonomous Robots*, 5(2):163–198, 1998.
- [70] Hassan K Khalil. Nonlinear systems. Prentice Hall, 2nd edition, 1996.
- [71] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research, 5(1):90–98, 1986.
- [72] Richard Knoblauch, Martin Pietrucha, and Marsha Nitzburg. Field studies of pedestrian walking speed and start-up time. *Transportation Research Record: Journal of* the Transportation Research Board, (1538):27–38, 1996.
- [73] D. E. Koditschek. Adaptive techniques for mechanical systems. In Fifth Yale Workshop on Applications of Adaptive Systems Theory, pages 259–265, May 1987.
- [74] D. E. Koditschek. Application of a new lyapunov function: Global adaptive inverse dynamics for a single rigid body. Technical Report 8806, Center for Systems Science, Yale University, 1988.
- [75] D. E. Koditschek. The control of natural motion in mechanical systems. Journal of Dynamic Systems, Measurement, and Control, 113:547–551, 1991.
- [76] D. E. Koditschek. Task encoding: Toward a scientific paradigm for robot planning and control. *Robotics and autonomous systems*, 9(1–2):5–39, 1992.
- [77] D. E. Koditschek, R. J. Full, and M. Buehler. Mechanical aspects of legged locomotion control. Arthropod Structure and Development, 33(3):251–272, 2004.
- [78] Daniel E Koditschek. The application of total energy as a lyapunov function for mechanical control systems. *Contemporary Mathematics*, 97:131, 1989.
- [79] Daniel E Koditschek and Elon Rimon. Robot navigation functions on manifolds with boundary. Advances in Applied Mathematics, 11(4):412–442, 1990.

- [80] Haldun Komsuoglu, Dave McMordie, Uluc Saranli, Ned Moore, Martin Buehler, and Daniel E Koditschek. Proprioception based behavioral advances in a hexapod robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3650–3655. IEEE, 2001.
- [81] Kurt Konolige, Motilal Agrawal, Morten Rufus Blas, Robert C Bolles, Brian Gerkey, Joan Solà, and Aravind Sundaresan. Mapping, navigation, and learning for off-road traversal. *Journal of Field Robotics*, 26(1):88–113, 2009.
- [82] Eric Krotkov, Scott Fish, Larry Jackel, Bill McBride, Mike Perschbacher, and Jim Pippine. The darpa perceptOR evaluation experiments. *Autonomous Robots*, 22(1):19–35, 2007.
- [83] Eric Langmuir. Mountaincraft and leadership: a handbook for mountaineers and hillwalking leaders in the British Isles. Scottish Sports Council, 1984.
- [84] Ti-Chung Lee, Kai-Tai Song, Ching-Hung Lee, and Ching-Cheng Teng. Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. *IEEE Transactions on Control Systems Technology*, 9(2):305–318, 2001.
- [85] D. Liberzon, E. Sontag, and Y. Wang. Universal construction of feedback laws achieving iss and integral-iss disturbance attenuation. Systems& Control Letters, 46(2):111– 127, 2002.
- [86] Zhiyun Lin, Bruce Francis, and Manfredi Maggiore. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transactions on Automatic Control*, 50(1):121–127, 2005.
- [87] Shu-Jun Liu and Miroslav Krstic. Stochastic source seeking for nonholonomic unicycle. Automatica, 46(9):1443–1453, 2010.
- [88] Gabriel AD Lopes and Daniel E Koditschek. Visual servoing for nonholonomically constrained three degree of freedom kinematic systems. *The International Journal of Robotics Research*, 26(7):715–736, 2007.
- [89] T. Lozano-Perez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984.
- [90] Wenjie Lu, Guoxian Zhang, and Silvia Ferrari. An information potential approach to integrated sensor path planning and control. *IEEE Transactions on Robotics*, 30(4):919–934, 2014.
- [91] Ren C Luo, Ming Hsiao, and Tsung-Wei Lin. Erect wheel-legged stair climbing robot for indoor service applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2731–2736. IEEE, 2013.
- [92] G. A. Lynch, J. E. Clark, and D. E. Koditschek. A self-exciting controller for highspeed vertical running. In *IEEE/RSJ International Conference on Intelligent Robots* and Systems (IROS), St. Louis, Missouri, USA, October 2009. IEEE.

- [93] Roberto Manduchi, Andres Castano, Ashit Talukder, and Larry Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. Autonomous robots, 18(1):81–102, 2005.
- [94] Joshua Marshall, Mireille E Broucke, Bruce Francis, et al. Formations of vehicles in cyclic pursuit. *IEEE Transactions on Automatic Control*, 49(11):1963–1974, 2004.
- [95] Bhaskara Marthi, Stuart J Russell, and Jason Andrew Wolfe. Angelic hierarchical planning: Optimal and online algorithms. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 222–231. AAAI Press, 2008.
- [96] Alexey S Matveev, Hamid Teimoori, and Andrey V Savkin. Navigation of a unicyclelike mobile robot for environmental extremum seeking. *Automatica*, 47(1):85–91, 2011.
- [97] Javier Minguez, Florant Lamiraux, and Jean-Paul Laumond. Motion planning and obstacle avoidance. In Bruno Siciliano and Oussama Khatib, editors, Springer handbook of robotics, pages 1177–1202. Springer, 2016.
- [98] Matty Mookerjee, Daniel Vieira, Marjorie A. Chan, Yolanda Gil, Charles Goodwin, Thomas F. Shipley, and Basil Tikoff. We need to talk: Facilitating communication between field-based geoscience and cyberinfrastructure communities. *GSA TODAY*, 25(11):34–35, 2015.
- [99] EZ Moore, D. Campbell, F. Grimminger, and M. Buehler. Reliable stair climbing in the simple hexapod 'RHex'. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2222–2227. IEEE, 2002.
- [100] F. Morabito, A. R Teel, and L. Zaccarian. Nonlinear antiwindup applied to eulerlagrange systems. *IEEE Transactions on Robotics and Automation*, 20(3):526–537, Jun 2004.
- [101] Rafael Morales, J Somolinos, and J Cerrada. Dynamic control of a reconfigurable stair-climbing mobility system. *Robotica*, 31(2):295–310, 2013.
- [102] Nicholas Morozovsky and Thomas Bewley. Stair climbing via successive perching. IEEE/ASME Transactions on Mechatronics, 20(6):2973–2982, 2015.
- [103] A.I. Mourikis, N. Trawny, S.I. Roumeliotis, D.M. Helmick, and L. Matthies. Autonomous stair climbing for tracked vehicles. *The International Journal of Robotics Research*, 26(7):737, 2007.
- [104] Robin R Murphy, Satoshi Tadokoro, Daniele Nardi, Adam Jacoff, Paolo Fiorini, Howie Choset, and Aydan M Erkmen. Search and rescue robotics. In Bruno Siciliano and Oussama Khatib, editors, Springer Handbook of Robotics, pages 1151–1173. Springer, 2008.
- [105] J. Nakanishi, T. Fukuda, and D. E. Koditschek. A brachiating robot controller. IEEE Transactions on Robotics and Automation, 16(2):109–123, 2000.

- [106] Petter Ogren and Naomi Ehrich Leonard. Obstacle avoidance in formation. In IEEE International Conference on Robotics and Automation (ICRA), volume 2, pages 2492– 2497. IEEE, May 2003.
- [107] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, 2002.
- [108] Dimitra Panagou and Herbert Tanner. Modeling of a hexapod robot; kinematic equivalence to a unicycle. Technical report, Department of Mechanical Engineering, University of Delaware, 2009.
- [109] Santiago Paternain, Daniel E Koditschek, and Alejandro Ribeiro. Navigation functions for convex potentials in a space with convex obstacles. *IEEE Transactions on Automatic Control*, 2017.
- [110] Alejandro Perez-Yus, Daniel Gutiérrez-Gómez, Gonzalo Lopez-Nicolas, and JJ Guerrero. Stairs detection with odometry-aided traversal from a wearable rgb-d camera. *Computer Vision and Image Understanding*, 154:192–205, 2017.
- [111] James Pippine, Douglas Hackett, and Adam Watson. An overview of the defense advanced research projects agency's learning locomotion program. *The International Journal of Robotics Research*, 30(2):141–144, 2011.
- [112] Feifei Qian, Douglas Jerolmack, Nicholas Lancaster, George Nikolich, Paul Reverdy, Sonia Roberts, Thomas Shipley, R Scott Van Pelt, Ted M Zobeck, and Daniel E Koditschek. Ground robotic measurement of aeolian processes. *Aeolian Research*, 27:1–11, 2017.
- [113] Ranjit Ray, Bikash Bepari, and Subhasis Bhaumik. On design and development of an intelligent mobile robotic vehicle for stair-case navigation. In Dilip Pratihar and Lakhmi Jain, editors, *Intelligent Autonomous Systems*, volume 275 of *Studies in Computational Intelligence*, pages 87–122. Springer Berlin / Heidelberg, 2010.
- [114] P. Reverdy, Berkay D. Ilhan, and D. E. Koditschek. A drift-diffusion model for robotic obstacle avoidance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6113–6120. IEEE, 2015.
- [115] Shai Revzen, B Deniz Ilhan, and Daniel E Koditschek. Dynamical trajectory replanning for uncertain environments. In *IEEE Conference on Decision and Control* (CDC), pages 3476–3483. IEEE, 2012.
- [116] Elon Rimon. A navigation function for a simple rigid body. In IEEE International Conference on Robotics and Automation, pages 546–551. IEEE, 1991.
- [117] H. Risken. The Fokker-Planck Equation. Springer, 1984.
- [118] A. A. Rizzi. Hybrid control as a method for robot motion programming. In IEEE International Conference on Robotics and Automation (ICRA), volume 1, page 832âĂŞ837. IEEE, 1998.

- [119] A. A. Rizzi and D. E. Koditschek. An active visual estimator for dexterous manipulation. *IEEE Transactions on Robotics and Automation*, 12(5):697–713, Oct 1996.
- [120] Erick J. Rodriguez-Seda, Chinpei Tang, Mark W. Spong, and Dusan M. Stipanovic. Trajectory tracking with collision avoidance for nonholonomic vehicles with acceleration constraints and limited sensing. *The International Journal of Robotics Research*, 33(12):1569–1592, 2014.
- [121] Uluc Saranli, Martin Buehler, and Daniel E Koditschek. RHex: A simple and highly mobile hexapod robot. The International Journal of Robotics Research, 20(7):616– 631, 2001.
- [122] E. Sayginer, T. Akbey, Y. Yazicioglu, and A. Saranlı. Task oriented kinematic analysis for a legged robot with half-circular leg morphology. In *IEEE Internation Conference* on Robotics and Automation (ICRA). IEEE, May 2009.
- [123] Paul S Schenker, Terry L Huntsberger, Paolo Pirjanian, Eric T Baumgartner, and Eddie Tunstel. Planetary rover developments supporting mars exploration, sample return and future human-robotic colonization. *Autonomous Robots*, 14(2–3):103–126, 2003.
- [124] Stephen Se and Michael Brady. Vision-based detection of staircases. In Fourth Asian Conference on Computer Vision (ACCV), volume 1, pages 535–540, 2000.
- [125] Jean-Antoine Seon, Brahim Tamadazte, and Nicolas Andreff. Decoupling path following and velocity profile in vision-guided laser steering. *IEEE Transactions on Robotics*, 31(2):280–289, 2015.
- [126] A Singer, Z. Schuss, A. Osipov, and D. Holcman. Partially reflected diffusion. SIAM Journal on Applied Mathematics, 68(3):844–868, 2008.
- [127] Sanjiv Singh and Paul Keller. Obstacle detection for high speed autonomous navigation. In *IEEE Internation Conference on Robotics and Automation (ICRA)*, pages 2798–2805. IEEE, 1991.
- [128] Sarjoun Skaff, George Kantor, David Maiwand, and Alfred Rizzi. Inertial navigation and visual line following for a dynamical hexapod robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1808–1813. IEEE, 2003.
- [129] D Soetanto, L Lapierre, and A Pascoal. Adaptive, non-singular path-following control of dynamic wheeled robots. In *IEEE Conference on Decision and Control (CDC)*, volume 2, pages 1765–1770. IEEE, 2003.
- [130] E. D. Sontag. The ISS philosophy as a unifying framework for stability-like behavior, pages 443–468. 2000.
- [131] E. D. Sontag. Input to state stability: Basic concepts and results, volume 1932 of Lecture Notes in Mathematics. Springer Verlag, 2008.

- [132] Eduardo D. Sontag and Yuan Wang. On characterizations of input-to-state stability with respect to compact sets. In *IFAC Non-Linear Control Systems Design Sympo*sium, pages 226–231. IFAC, 1995.
- [133] M. J. Spenko, J. A. Saunders, G. C. Haynes, M. R. Cutkosky, A. A. Rizzi, R. J. Full, and D. E. Koditschek. Biologically inspired climbing with a hexapedal robot. *Journal* of Field Robotics, 25(4–5):223–242, 2008.
- [134] Solomon Steplight, Geoffrey Engal, Sang-Hack Jung, Daniel B. Walker, Camillo J. Taylor, and James P. Ostrowski. A mode-based sensor fusion approach to robotic stairclimbing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), pages 1113–1118. IEEE, 2000.
- [135] Gilbert Strang. Linear Algebra and Its Applications. Harcourt Brace Jovanovic, Inc., 3rd edition, 1988.
- [136] P. Szymczak and A. J. C. Ladd. Boundary conditions for stochastic solutions of the convection-diffusion equation. *Physical Review E*, 68:036704, 2003.
- [137] Y Tan, WH Moase, C Manzie, D Nešić, and IMY Mareels. Extremum seeking from 1922 to 2010. In *Chinese Control Conference (CCC)*, pages 14–26. IEEE, 2010.
- [138] S. Tarbouriech and M. Turner. Anti-windup design: an overview of some recent advances and open problems. *IET Control Theory & Applications*, 3(1):1–19, Jan 2009.
- [139] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. MIT Press, 2005.
- [140] S. H. Turlapati, M. Shah, S. P. Teja, A. Siravuru, S. V. Shah, and Madhava Krishna K. Stair climbing using a compliant modular robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3332–3339. IEEE, 2015.
- [141] Maarten K Van Aalst. The impacts of climate change on the risk of natural disasters. Disasters, 30(1):5–18, 2006.
- [142] Vasileios Vasilopoulos, William Vega-Brown, Omur Arslan, Nicholas Roy, and Daniel E. Koditschek. Sensor-based reactive symbolic planning in partially known environments. In *IEEE Internation Conference on Robotics and Automation (ICRA)*, 2018.
- [143] William Vega-Brown and Nicholas Roy. Asymptotically optimal planning under piecewise-analytic constraints. In *The 12th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.
- [144] Theodore Von Karman and G Gabrielli. What price speed? specific power required for propulsion of vehicles. *Mechanical Engineering*, 72:775–781, 1950.
- [145] J. D. Weingarten, R. E. Groff, and D. E. Koditschek. A framework for the coordination of legged robot gaits. In *IEEE Conference on Robotics, Automation and Mechatronics*, volume 2, 2004.

- [146] Joel D Weingarten, Gabriel AD Lopes, Martin Buehler, Richard E Groff, and Daniel E Koditschek. Automated gait adaptation for legged robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2153–2158. IEEE, 2004.
- [147] Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.
- [148] Garrett J. Wenger, Aaron M. Johnson, Camillo J. Taylor, and Daniel E. Koditschek. Semi-autonomous exploration of multi-floor buildings with a legged robot. In Unmanned Systems Technology XVII, volume 9468, pages 94680B–8, Baltimore, MD, 2015. SPIE.
- [149] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(1):42–56, 2003.
- [150] Thomas Westfechtel, Kazunori Ohno, Bärbel Mertsching, Daniel Nickchen, Shotaro Kojima, and Satoshi Tadokoro. 3d graph based stairway detection and localization for mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 473–479. IEEE, 2016.
- [151] David Wettergreen, Scott Moreland, Krzysztof Skonieczny, Dominic Jonak, David Kohanbash, and James Teza. Design and field experimentation of a prototype lunar prospector. *The International Journal of Robotics Research*, 2010.
- [152] Yalin Xiong and L. Matthies. Vision-guided autonomous stair climbing. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1842–1847. IEEE, 2000.
- [153] Jung-Min Yang and Jong-Hwan Kim. Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 15(3):578–587, 1999.
- [154] Chunlei Zhang, Daniel Arnold, Nima Ghods, Antranik Siranosian, and Miroslav Krstic. Source seeking with non-holonomic unicycle without position measurement and with tuning of forward velocity. Systems & control letters, 56(3):245–252, 2007.