12-2023

Safe Navigation of Quadruped Robots Using Density Functions

Andrew Zheng
Clemson University, azheng@clemson.edu

SAFE NAVIGATION OF QUADRUPED ROBOTS USING DENSITY FUNCTIONS

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

---

by
Andrew Zheng
December 2023

Accepted by:
Dr. Umesh Vaidya, Committee Chair
Dr. Ardalan Vahidi
Dr. Ian Walker

Abstract

Safe navigation of mission-critical systems is of utmost importance in many modern autonomous applications. Over the past decades, the approach to the problem has consisted of using probabilistic methods, such as sample-based planners, to generate feasible, safe solutions to the navigation problem. However, these methods use iterative safety checks to guarantee the safety of the system, which can become quite complex. The navigation problem can also be solved in feedback form using potential field methods. Navigation function, a class of potential field methods, is an analytical control design to give almost everywhere convergence properties, but under certain topological constraints and mapping onto a sphere world.

Alternatively, the navigation problem can be formulated in the dual space of density. Recent works have shown the use of linear operator theory on density to convexly approach the navigation problem. Inspired by those works, this work uses the physical-based interpretation of occupation through density to synthesize a safe controller for the navigation problem. Moreso, by using this occupation-based interpretation of density, we design a feedback density-based controller to solve the almost everywhere navigation problem.

Furthermore, due to the recent popularity of legged locomotion for the navigation problem, we integrate this analytical feedback density-based controller into the quadruped navigation problem. By devising a density-based navigation architecture, we show in simulation and hardware the results of the density-based navigation.

Acknowledgements

I would like to thank my advisor, Umesh Vaidya, for giving me the opportunity to pursue my M.S. degree at the Distributed Intelligence and Robot Autonomy Lab. His knowledge helped educate me and his ambitions greatly inspired me to pursue my own research.

I would also like to thank Sriram Krishnamoorthy, Joseph Moyalan, and Sarang Sutavani for the close collaboration in our research. It has been a pleasure working together in the lab with many fruitful discussions and inspiring ideas. I would also like to thank Alex Krolicki and Dakota Rufino for working with me during my early days of my graduate program and guiding me in my research. Special thanks to Jake Buzhardt, Prashanth Chivkula, Colin Rodwell, and the rest of the people in my office for a great time during my master's program.

Lastly, I would like to thank my friends and family for the continuous support. You have been a strong pillar in my life and helped me push forward in my research.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Navigation under safety constraints is of utmost importance in mission-critical systems. However, a few critical factors need to be considered when looking to tackle the problem, as shown in Figure 1.1. These factors can potentially be the identification of the safe and unsafe region, consideration of autonomous system constraints, and considerations of uncertainty in the model of the system or the estimated states. These cumulative components make the safe navigation problem challenging. Furthermore, the safe navigation problem may not only be in an environment separated by safe and unsafe regions, but in an environment with degrees of safety – which can be categorized as the navigation problem in unstructured environments [37]. This additional layer of complexity remains a topic of great interest actively being looked at for the navigation problem [9, 17]. Moreover, in the unstructured navigation problem where the degree of safety can be represented as a traversability map, different systems will have different traversability constraints to remain safe, such as mobile vehicles with their rollover stability through the balance of centripetal acceleration and friction force [38] and legged systems with their body stability through foothold placements [18]. These problem-specific constraints make it difficult to pose the navigation problem from a general point of view [14, 50].

Therefore, a particular autonomous system that is of great interest for the navigation problem and has seen a large amount of success is the legged robotic system. Legged robots are systems that utilize their limbs to navigate around the world, allowing these systems to navigate in structured environments, such as planar surfaces and stairs, and unstructured environments, such as ramps, hills, and planks. Examples of these

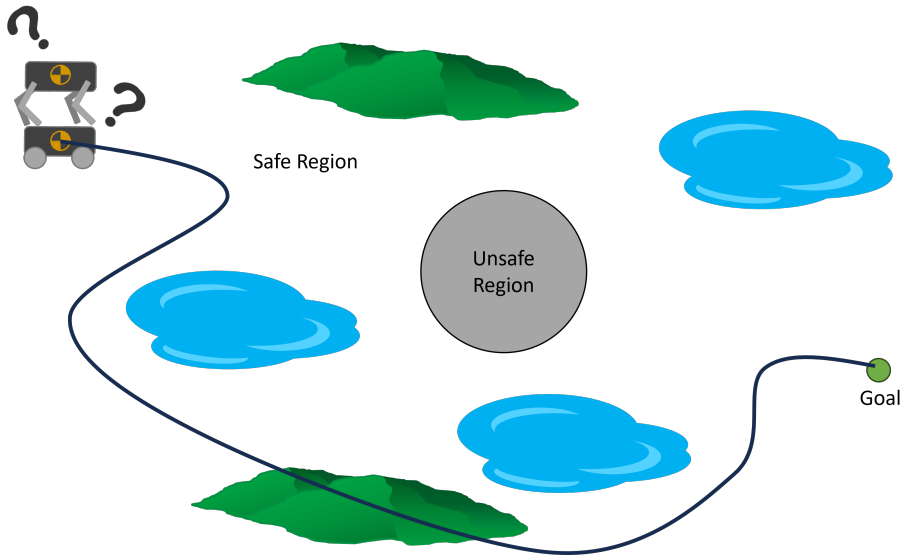

Figure 1.1: General navigation problem in unstructured environment. Diagram showcases the consideration of different dynamic constraints, uncertainty, identification of safe and unsafe region, different traversable areas to tackle the navigation problem of traversing through start to goal.

systems consist of bipeds – two-legged robots [15], quadrupeds – four-legged robots [5], and hexapods – six-legged robots [44]. Naturally, there are more variations to the legged systems, but they all extend towards an increase in legs or an additional appendage, such as an arm or a tail [45]. Although an increase in appendages offers more possibilities in how the robot interacts with the world, which may be beneficial in unorthodox environments, the increase in limbs also increases the complexity of how the robot may navigate the world. Therefore, when considering the navigation problem, one must balance between capability and complexity.

A natural consideration of the balance between the two factors for legged systems is the quadrupedal robots. The recent surge in quadrupedal research within the field [6, 10, 19, 23, 29] and the increase in the commercialization of quadruped robots, such as Unitree Go1, Anybotics Anymal X, and the Boston Dynamic Spot are proof of the capabilities is a good indicator of the balance. Furthermore, the advantages of the four-legged systems were accentuated in a recent subterranean challenge [1, 16, 43, 46], where the objective of the subterranean challenge was to have autonomous systems complete specific tasks while navigating in a subterranean environment. Within this challenge, one of the core difficulties that obstructed the teams was the diverse environmental factors such as rocky terrains, slippery surfaces, staircases, and more. Therefore, the teams' success relied on how their system would handle these various environmental conditions. Thus, it is noted that the successful teams relied on quadrupedal robots to navigate around the subterranean world, as the system naturally accommodates these environmental conditions.

Therefore, inspired by the diverse capabilities of legged robots to traverse through different environments, our interest lies in addressing the safe navigation problem and applying it on quadruped robots.

1.1.1 Contributions

This thesis first presents a novel approach for safe control synthesis using the dual formulation of the navigation problem by providing an analytical construction of density functions for almost everywhere safe navigation with safety constraints. In contrast to the existing approaches, where density functions are used for the analysis of navigation problems, we use the physical interpretation of density functions through occupation for the synthesis of safe controllers. We provide convergence proof using the proposed density functions for navigation with safety. Further, we use these density functions to design feedback controllers capable of navigating in cluttered environments and high-dimensional configuration spaces. The proposed analytical construction of density functions overcomes the problem associated with navigation functions, which are known to exist but challenging to construct, and potential functions, which suffer from local minima.

Second, we present a motion planning architecture to apply the novel navigation density function onto a quadruped. We decompose the locomotion problem into a high-level density planner and a model predictive controller (MPC), where the feedback planner synthesizes safe trajectories for the predictive controller to locally optimize. This proposed method simplifies the model of the quadruped robot into an integrator system, where the high-level plan is in feedback form, formulated through the constructed density function, and the model predictive controller tracks the safe trajectories through locally optimal control trajectories. The overall framework is implemented in both simulation and hardware, demonstrating the effectiveness of the feedback density planner for the legged locomotion problem.

The rest of the work is organized as follows. Chapter 2 discusses background information about density navigation and quadruped dynamics. Next, Chapter 3 presents the work ”Safe Navigation Using Density Functions” at IEEE Robotics and Automation Letters, which showcases the analytical construction of density to synthesize a feedback controller for almost everywhere navigation for integrator systems [54]. Sequentially, Chapter 4 presents the work ”Safe Motion Planning for Quadruped Robot Using Density Functions” accepted to the Indian Control Conference, which gives details about how the synthesized feedback controller is used and applied to the quadruped locomotion problem [35]. Lastly, Chapter 5 summarizes the contribution of this work with navigation using density and discusses future work of extending the work to utilize the legged locomotion capabilities in unstructured terrain.

1.2 Other Contributions

While this thesis is focused on presenting the two papers, ”Safe Navigation Using Density Functions” and ”Safe Motion Planning for Quadruped Robot Using Density Functions”, I wish to note my other research contribution that will not be discussed in this thesis

- Alexander Krolicki, Dakota Rufino, Andrew Zheng, Sriram S.K.S. Narayanan, Joseph Erb, and Umesh Vaidya. ”Modeling Quadruped Leg Dynamics on Deformable Terrains using Data-driven Koopman Operators.” IFAC-PapersOnLine 55.37, 2022: 420-425.

- Sarang Sutavani, Andrew Zheng, Ajinkya Joglekar, Jonathon Smeraka, David Gorsich, Venkat Krovi, and Umesh Vaidya. ”Artificial Neural Network Based Terrain Reconstruction for Off-Road Autonomous Vehicles Using LiDAR”. Ground Vehicle Systems Engineering and Technology Symposium, 2023.

- Joseph Moyalan, Andrew Zheng, Sriram S.K.S. Narayanan, and Umesh Vaidya. ”Off-Road Navigation of Legged Robots Using Linear Transfer Operators.” Modeling, Estimation, and Control Conference, 2023.

- Sriram S.K.S. Narayanan, Andrew Zheng, and Umesh Vaidya. ”Density Functions for Safe Navigation of Robotic Systems in Dynamic Environments.” American Control Conference, 2024.

- Joseph Moyalan, Sriram S.K.S. Narayanan, Andrew Zheng, and Umesh Vaidya. ”Synthesizing Controller for Safe Navigation using Control Density Function.” American Control Conference, 2024.

Chapter 2

Preliminaries

This thesis present a novel approach to safe navigation problem. The novel approach relies on the dual formulation of the navigation problem in the space of density. In this chapter, we provide preliminaries from the linear transfer operator theory that form the mathematical foundation for the dual formulation of navigation problem in the dual space of density. We also provide preliminaries for the reduced order modelling of the quadruped dynamics that we use to demonstrate the application of safe navigation using density function.

2.1 Operator Theory for Safe Navigation

The connection of operator theory is heavily involved in the construction of the density navigation functions through occupation proposed in this thesis. Therefore, to give the appropriate context of this work, we give a brief overview of linear operator theory, its connection to stability theory, and how operator theoretic approach to stability extends to the almost everywhere safe navigation problem. We refer the readers to [26, 48, 49] for more details on this topic.

2.1.1 Linear Operator Theory

Consider a continuous-time dynamical system in the form of

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}(t)) \tag{2.1}$$

where the following nonlinear flow map system $\mathbf{f}$ describes how the state $\mathbf{x}(t) \in \mathbb{R}^n$ propagated forward in time. Alternatively, the dynamics of the system can be rewritten through an operator-theoretic approach. The Koopman operator $\mathbb{K} : \mathscr{L}_\infty \to \mathscr{L}_\infty$ is an infinitesimal operator that acts on the space of observables defined as

$$[\mathbb{K}_t \Omega](\mathbf{x}) = \Omega(\mathbf{s}_t(\mathbf{x})), \tag{2.2}$$

where Ω is the space of observable functions. Here, $s_t(\mathbf{x})$ is the solution of (2.1) or the flow map of the system under a dynamic given initial conditions $\mathbf{x}$ and a given t. Given that that $\mathbf{f}$ and Ω are continuously differentiable, then $\Omega(t,\mathbf{x}) = \mathbb{K}\Omega_0(\mathbf{x})$ satisfies the partial differential equation (see [26])

$$\frac{\partial \Omega}{\partial t} = \mathbf{f} \cdot \nabla \Omega := \mathscr{L}_{\mathbb{K}} \Omega, \tag{2.3}$$

where $\mathfrak{L}_{\mathbb{K}}$ is the infinitesimal generator of $\mathbb{K}$ i.e.,

$$\mathfrak{L}_{\mathbb{K}} \Omega = \lim_{t \to 0} (\mathbb{K}_t \Omega - \Omega) / t.$$

Correspondingly, there is a dual relation of the Koopman operator with the Perron-Frobenius (P-F) operator (see [26])

$$\int_{\mathbb{R}^n} [\mathbb{K}_t \Omega](\mathbf{x}(t)) g(\mathbf{x}(t)) d\mathbf{x} = \int_{\mathbb{R}^n} [\mathbb{P}_t g](\mathbf{x}(t)) \Omega(\mathbf{x}(t)) d\mathbf{x}, \tag{2.4}$$

where the P-F operator $\mathbb{P}_t$ describes the transport of densities $\mathbf{g}$ under dynamics through the following $g(t,\mathbf{x}) = \mathbb{P}_t g_0(\mathbf{x})$. Likewise, given that $\mathbf{f}$ and g are continuously differentiable, then the following satisfies the transport equation (see [26]).

$$\frac{\partial g}{\partial t} = -\nabla \cdot (\mathbf{f} g) := \mathfrak{L}_{\mathbb{P}} g. \tag{2.5}$$

where $\mathfrak{L}_{\mathbb{P}}$ is the Perron-Frobenius generator i.e.,

$$\mathfrak{L}_{\mathbb{P}} g = \lim_{t \to 0} (\mathbb{P}_t g - g) / t.$$

2.1.2 Duality in Stability Theory from Linear Operator Theoretic Perspective

The duality in stability theory was discovered in [49] from linear operator theoretic perspective. In particular, it was shown that the traditional notion of point-wise stability as captured using Lyapunov function is connected to the Koopman operator. On the other hand, the weaker notion of almost everywhere stability

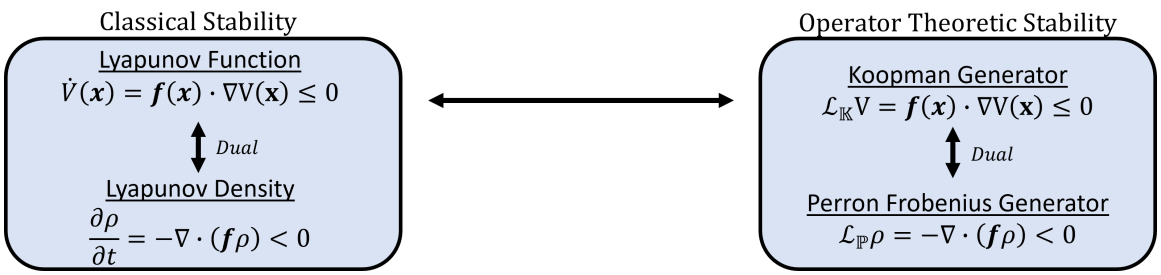

Figure 2.1: Stability analysis through classical method and operator-theoretic framework. The duality between Lyapunov function and Lyapunov density is highlighted in [40, 49].

as discovered in [40] is connected to Perron-Frobenius operator. This weaker notion of almost everywhere stability can be verified using density function for a continous-time system and Lyapunov measure for a discrete-time system. Given that the Koopman and the Perron-Frobenius operator are dual to each other, the dual between the Lyapunov function and the density function or Lyapunov measure follows.

Lyapunov's direct method has been a long-standing tool for the analysis (and control synthesis) of stability for nonlinear dynamical systems. However, many works have shown alternative notions of the stability problem as shown in 2.1. We first discuss the Lyapunov stability criterion. It makes use of a candidate Lyapunov function $V(\mathbf{x})$ and a corresponding stability criterion

$$\mathbf{f}(\mathbf{x}) \cdot \nabla V < 0 \tag{2.6}$$

to verify the stability of the nonlinear system $\mathbf{f}$, given that $V(\mathbf{0}) = 0$ and $V(\mathbf{x}) > 0 \ \ \forall \mathbf{x} \neq 0$. We note the close relation of Lyapunov function and the Koopman generator (and likewise the operator) as shown in (2.3) and (2.6). Therefore, as the Koopman operator has a dual relation to the P-F operator, which acts on the space of density, there is a notion for a dual to the Lyapunov stability criterion, which acts on the space of density. This dual notion of stability was first presented in [40], where the author introduced a weaker notion of stability, almost everywhere (a.e.) stability through the usage of nonnegative density functions $\rho \in \mathscr{C}^1(\mathbb{R}^n \setminus \{0\}, \mathbb{R})$ that satisfies the following condition

$$\nabla \cdot (\mathbf{f}\rho) > 0. \tag{2.7}$$

The density function $\rho(\mathbf{x})$ should be integrable on $\mathbb{R}^n \setminus N_\varepsilon$, where N_ε is ε neighborhood of the equilibrium point at the origin.

It is important to notice that the Lyapunov inequality in (2.6) can be expressed using the Koopman

generator and the density inequality on (2.7) can be expressed using the P-F generator. In particular, we have the following

$$\mathbf{f}(\mathbf{x}) \cdot \nabla V < 0 \iff \mathfrak{L}_{\mathbb{K}} V < 0 \tag{2.8}$$

$$\nabla \cdot (\mathbf{f}(\mathbf{x})\rho) > 0 \iff \mathfrak{L}_{\mathbb{P}} \rho < 0 \tag{2.9}$$

2.1.3 Safe Almost Everywhere Navigation using Operator Theory

The notion of stability and safe navigation are connected. In particular, the use of Lyapunov function for stability can be extended to navigation with safety constraints in the form of navigation function or potential function [21,42]. Similarly, the use of density function used for almost everywhere stability verification can be extended for almost everywhere navigation with safety.

The use of density functions for navigation is based on the occupancy-based physical interpretation of the density function [48]. In particular, consider a dynamical system (2.1). For the dynamical system, assume that there exists a density function $\rho(\mathbf{x})$ that is positive and integrable and satisfies the following equation

$$\nabla \cdot (\mathbf{f}(\mathbf{x})\rho) = g(\mathbf{x}) > 0, \tag{2.10}$$

for some positive function $g(\mathbf{x})$. For the purpose of concreteness, we assume that $g(\mathbf{x}) = 1_D(\mathbf{x})$ i.e., indicator function of set D. Let μ be the measure corresponding to density ρ i.e.,

$$\mu(A) = \int_A \rho(\mathbf{x}) d\mathbf{x}$$

for any set $A \subset \mathbb{R}^n$. $\mu(A)$ signifies the occupancy of system trajectories starting from the set D in the set A. In particular, if $\mu(A) = 0$, then almost all system trajectories starting from set D will not enter the set A i.e., zero occupancy in set A. The physical significance of the density function as a measure of occupancy is critical in providing systematic and analytical construction of density function for solving safe navigation problems. Consider the problem of safe navigation where the objective is to steer the system with initial conditions in set, say D, to some target set, say O, while avoiding the unsafe set U. This safe navigation problem using the

density function can be formulated as follows.

$$\nabla \cdot (\mathbf{f}(\mathbf{x})\rho) = 1_D(\mathbf{x}) \tag{2.11}$$

$$\int_U \rho(\mathbf{x}) d\mathbf{x} = 0 \tag{2.12}$$

The objective is to find a density ρ that is positive and integrable outside the set O satisfying the above property. The above is a convex but infinite-dimensional optimization problem to be solved for the density function $\rho(\mathbf{x})$.

2.2 Quadruped Dynamics

To understand how the navigation problem can be formulated for quadrupeds, one must look at the mathematical model that describes how the system moves around the world. We begin by looking at the dynamics of the quadruped by a high-fidelity full-order rigid body model. We then consider assumptions that can be made for the dynamics of the quadruped, which introduces the corresponding centroidal model, single rigid body model, and linear inverted pendulum model. For the full derivation of the model simplifications, see [52].

We note that the notations defined in this section are defined only for quadrupeds, as well as the sections later pertaining to quadrupeds, and should not be confused with notations defined for density navigation. There should be little to no overlaps between the notations, but the appropriate section in this thesis should ease the differentiation in notations if any do conflict.

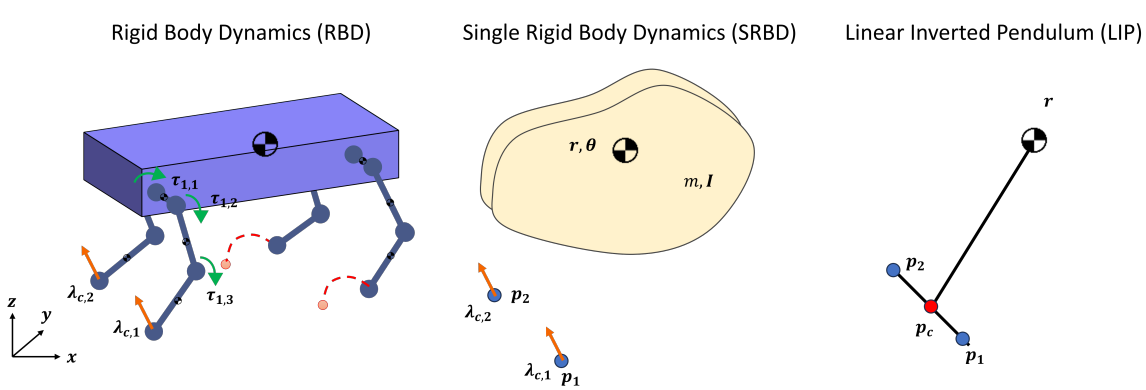

Figure 2.2: Visual representation of the dynamic modeling equations for quadruped pushing off its front hind legs.

2.2.1 Rigid Body Dynamics

Assumption: *Rigid bodies do not deform.*

To model a four-legged robot, one can model each of its rigid bodies. The generalized coordinates of the quadruped system are noted as $\mathbf{q} = [\mathbf{q}_b \;\; \mathbf{q}_j] \in SE(3) \times \mathbb{R}^{12}$, where $\mathbf{q}_b$ are the position and orientation of the body and $\mathbf{q}_j$ are the joint angles of the leg. Therefore, the dynamics of the quadruped is through the following equation of motion

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}(\mathbf{q})^\top \boldsymbol{\lambda}_c, \tag{2.13}$$

where $\mathbf{M} \in \mathbb{R}^{18\times 18}$ is the joint-space inertia matrix, $\mathbf{H} \in \mathbb{R}^{18}$ describes the potential and nonlinear effect (e.g. gravity, Centrifugal, and Coriolis), $\mathbf{S}^\top = [\mathbf{0} \;\; \mathbf{I}_{12\times 12}]$ is a matrix that applies torques to row corresponding to the joint angles, $\boldsymbol{\tau} \in \mathbb{R}^{12}$ is the torque exerted from the actuators, and the Jacobian $\mathbf{J}$ maps the contact forces $\boldsymbol{\lambda}_c \in \mathbb{R}^{12}$ from the end effectors to the generalized coordinates.

The dynamics of the quadruped can also be separated into the unactuated system and actuated system governed by the following equation

$$\mathbf{M}_u(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}_u(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_u(\mathbf{q})^\top \boldsymbol{\lambda}_c \tag{2.14}$$

$$\mathbf{M}_a(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}_a(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} + \mathbf{J}_a(\mathbf{q})^\top \lambda_c, \tag{2.15}$$

where the subscript a in (2.15) describes the 12 torque actuated rows and the subscript u describes the 6 unactuated rows. Note, the 6 unactuated rows shown in (2.14) suggest that no direct torque actuation can be applied onto the unactuated system. Therefore, the unactuated system (i.e. base) can only evolve through the effects of the joints and the resulting forces from contact.

2.2.2 Centroidal Dynamics

Centroidal dynamics can be described as the projection of the quadruped dynamics into the center of mass frame, expressed as follows [36]

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{A}}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} m\mathbf{g} + \sum_{i=1}^{4} \boldsymbol{\lambda}_{c,i} \\ \sum_{i=1}^{4} (\mathbf{r}_{com}(\mathbf{q}) - \mathbf{p}_i(\mathbf{q})) \times \boldsymbol{\lambda}_{c,i} \end{bmatrix}. \tag{2.16}$$

where $\mathbf{A} \in \mathbb{R}^{6\times 12}$ is known as the Centroidal Momentum Matrix that maps the velocities into the center of mass frame, m is the mass of the robot, $\mathbf{g}$ is the gravitational acceleration, $\boldsymbol{\lambda}_{c,i}$ is the contact force generated by end effector i, $\mathbf{r}_{com}$ is the center of mass, and $\mathbf{p}_i$ is the position of the end effector. (2.16) can essentially be viewed through the Newton-Euler equation of change of momentum of the rigid bodies projected onto the base ($\frac{d(\mathbf{A}\dot{\mathbf{q}})}{dt} = \mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{A}}(\mathbf{q})\dot{\mathbf{q}}$), resulting in the right-hand side to be the sum of forces and moments.

2.2.3 Single Rigid Body Model

To reduce the complexity of (2.16), the dependency of the model on the joints is removed:

Assumption: *Momentum produced by the joint velocities is negligible*

Assumption: *Full inertia remains similar to the inertia in nominal joint configuration.*

As a large amount of the mass for most of the legged robot design (e.g. MIT Cheetah [5], Spot-Mini, Anymal) is distributed onto the base, especially the case for MIT Cheetah and Spot (the robots utilized a belt drive such the motors are located closer to the base), the assumption that the momentum and inertia produce by the legs is negligible is quite valid. Moreover, these assumptions are validated even further when the robots move slowly or have very little change in joint configuration due to small movements.

Therefore, (2.16) can be rewritten such that the linear and angular components are decoupled as follows

$$\ddot{m}\ddot{\mathbf{r}}_{com} = m\mathbf{g} + \sum_{i=1}^{4} \boldsymbol{\lambda}_{c,i} \tag{2.17}$$

$$\mathbf{I}(\boldsymbol{\theta})\dot{\omega} + \omega \times \mathbf{I}(\boldsymbol{\theta})\omega = \sum_{i=1}^{4} (\mathbf{r}_{com} - \mathbf{p}_i) \times \boldsymbol{\lambda}_{c,i}, \tag{2.18}$$

where $\mathbf{I}(\boldsymbol{\theta}) \in \mathbb{R}^{3\times 3}$ is the full inertia projected onto the inertial aligned base frame, ω is the body angular velocity between the inertial and body frame, expressed in the inertial frame.

Note, one key component to this model is the loss of joint information. This makes the model independent of joint states, which at the expense of some accuracy, reduces the complexity of the model in both dimension and nonlinearity. This allows for simpler formulation for the legged locomotion problem.

2.2.4 Linear Inverted Pendulum Model

Although (2.17) is a reduced order model, the complexity of the model is still present in the nonlinearity through the cross products. Therefore, to eliminate the nonlinearities, a few assumptions are made:

Assumption: *Angular velocity ω and $\dot{\omega}$ of the base are negligible.*

Assumption: *Center of mass height r_z is constant.*

Assumption: *Footholds are at a constant height p_z.*

The following dynamic equation under these assumptions simplifies to the following equation:

$$m\ddot{r}_x = \sum_{i=1}^{4} \lambda_{c,i,x} = \frac{mg}{r_z - p_z}\left(r_x - \frac{\sum_{i=1}^{4} \lambda_{c,i,z} p_{i,x}}{\sum_{i=1}^{4} \lambda_{c,i,z}}\right) = \frac{mg}{h}(r_x - p_{c,x}) \tag{2.19}$$

where x notates the forward motion (y the horizontal), h is the walking height of the robot, and $p_{c,x}$ is the center of pressure affected by the weighted average from the vertical contact force $\lambda_{c,i,z}$ and the position of contact $p_{i,x}$.

Although strict assumptions limit the model's utility, the linear model is still helpful in generating planar legged locomotion plans.

Chapter 3

Safe Navigation Using Analytically Constructed Density Functions

This chapter is adapted from a paper in *IEEE Robotics and Automation Letters*:

> Andrew Zheng, Sriram S. K. S. Narayanan, and Umesh Vaidya, "Safe Navigation Using Density Functions," in IEEE Robotics and Automation Letters.

3.1 Introduction

The navigation problem is usually decomposed into planning and control, where safety must be ensured at each sublevel [27]. Within the planning problem, the objective usually involves defining a collision-free trajectory in the feasible configuration space given an initial and final configuration.

A standard method to solve the planning problem involves using sample-based planners such as rapidly-exploring random tree search (RRT) and probabilistic roadmaps (PRM) [2, 28]. These sample-based methods are observed to be probabilistically complete through iterative samples of locally safe and feasible paths. Asymptotically optimal variations of these planners have been developed in [20], where the convergence rate for optimality is improved in [12, 17].

Another method involves designing feedback controllers that jointly solve the convergence and collision-free avoidance problem. Artificial potential field based methods have attempted to solve the joint

problem by the sum of attractive and repulsive potentials [22]. However, the existence of local minima is a well-known issue [25, 42]. In [24, 42], a class of analytical potential functions, known as navigation functions (NFs), are introduced, which guarantees almost everywhere (a.e.) convergence while adhering to safety constraints. This method relies on a range of problem-specific tuning parameters to guarantee a.e. convergence. Moreso, complex safety constraints arising from arbitrarily shaped obstacles are generally limited by the possible mapping to a model sphere world.

The navigation problem can alternatively be formulated in the dual space of density. In [48], a navigation measure was introduced to provide a convex formulation for synthesizing safe controllers. In the continuous-time setting, the density function was used as a safety certificate for the analysis and synthesis using the sum of squares optimization method [41]. Similarly, density-based approaches are also used for the convergence analysis of existing navigation algorithms [8, 31]. More recently, convex data-driven approaches based on the linear transfer Perron-Frobenius and Koopman operators are used for solving the optimal navigation problem with safety constraints [34, 53]. In contrast to using the convex dual formulation for navigation, we provide an analytical construction of density functions for navigation. In particular, the analytical construction of navigation density can be viewed as the dual construction of the classical NFs from [42]. However, unlike [42], the construction is not restricted to navigation in the sphere world environment.

The main contribution of this chapter is in providing analytical construction of density functions used for solving the safe navigation problem. The density function has a physical interpretation, where the measure associated with the density is a measure of occupancy of the system trajectories in any set of the state space as shown in Figure 3.1. We exploit this occupancy-based physical interpretation of the density function in the construction of the navigation density functions. Unlike NFs, the density formulation can represent arbitrary shapes of the obstacle sets. We prove that the proposed density function can navigate almost all initial conditions from the initial set to the target set while avoiding the obstacle set. We show navigation results for simple integrator dynamics in complex environments as well as high-dimensional configuration spaces. Similarly, navigation results for obstacle avoidance involving robotics systems such as the two-link planar robotic arm manipulator are presented.

The rest of the chapter is organized as follows. Section 3.2 discuss the preliminaries and the problem formulation. Section 3.3 discusses the construction of density functions, and Section 3.4 discusses the properties of density functions for the navigation problem. This is followed by application to robotic systems in section 3.5 and conclusive remarks in section 3.6.

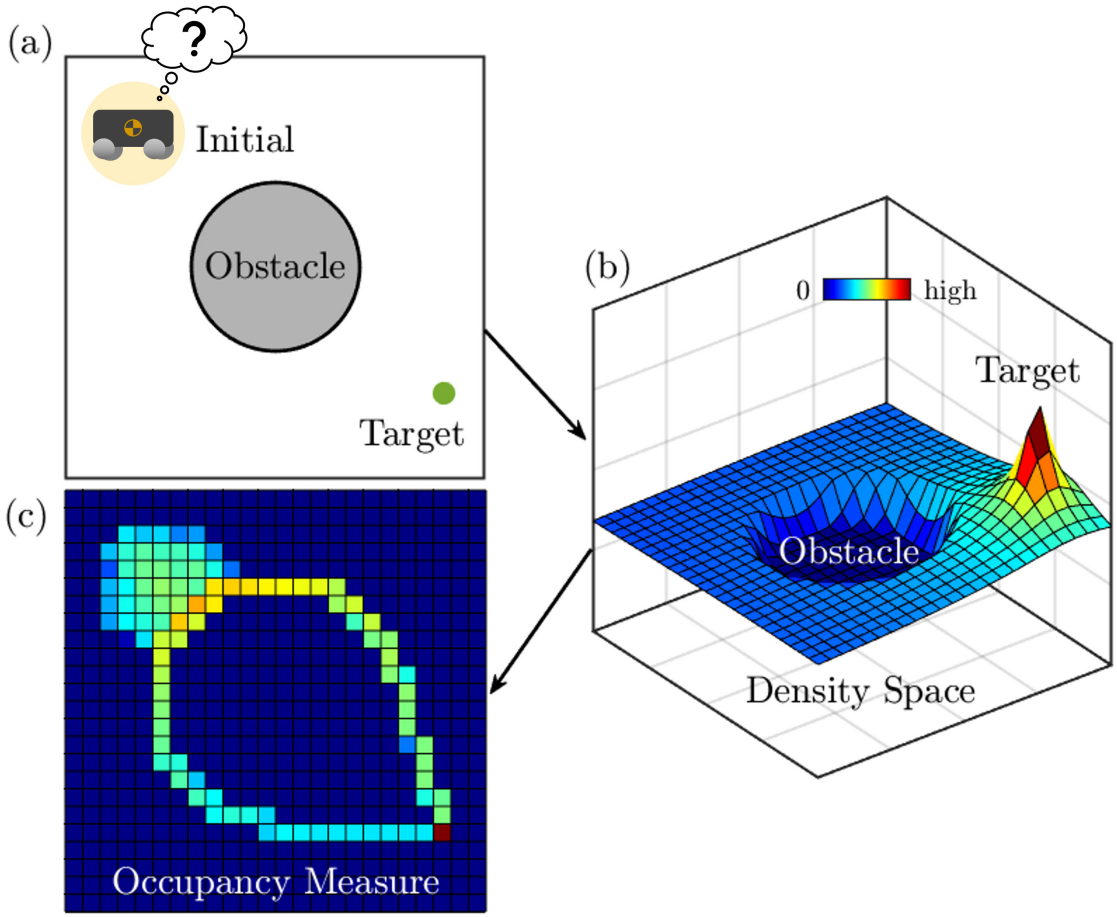

Figure 3.1: Navigation framework using density where (a) defines the navigation problem, (b) shows the density for navigation, and (c) shows occupancy measure, which physically denotes the duration of system trajectories occupying the set.

3.2 Problem Statement

Notations: The following notations will be used in this paper. $\mathbb{R}^n$ denotes the n dimensional Euclidean space, $\mathbf{x} \in \mathbb{R}^n$ denotes a vector of system states, $\mathbf{u} \in \mathbb{R}^n$ is a vector of control inputs. Let $\mathbf{X} \subset \mathbb{R}^n$ be a bounded subset that denotes the workspace for the robot. $\mathbf{X}_0$, $\mathbf{X}_T$, $\mathbf{X}_{u_k} \subset \mathbf{X}$, for $k = 1, \ldots, L$ denote the initial, target, and unsafe sets, respectively. With no loss of generality, we will assume that the target set is a single point set and located at the origin, i.e., $\mathbf{X}_T = \{0\}$. $\mathbf{X}_u = \cup_{k=1}^{L} \mathbf{X}_{u_k}$ defines the unsafe set and $\mathbf{X}_s := \mathbf{X} \setminus \mathbf{X}_u$ defines the safe set. We will denote by $\mathbf{X}_1 := \mathbf{X} \setminus \mathscr{B}_\delta$, where $\mathscr{B}_\delta$ is the δ neighborhood of the origin for arbitrary small δ. We use $\mathscr{C}^k(\mathbf{X})$ to denote the space of all k-times differentiable functions of $\mathbf{X}$. We use $\mathscr{M}(\mathbf{X})$ to denote the space of all measures on $\mathbf{X}$ and $m(\cdot)$ to denote the Lebesgue measure. $\mathbb{1}_A(\mathbf{x})$ denotes the indicator function

for set $A \subset \mathbf{X}$. The formal statement of the navigation problem that we solve in this paper is stated as follows.

Problem 1 *(Almost everywhere navigation) The objective of this problem is to design a smooth feedback control input* $\mathbf{u} = \mathbf{k}(\mathbf{x})$ *to drive the trajectories of the dynamical system*

$$\dot{\mathbf{x}} = \mathbf{u}, \tag{3.1}$$

from almost every initial condition (w.r.t. Lebesgue measure) from the initial set $\mathbf{X}_0$ *to the target set* $\mathbf{X}_T$ *while avoiding the unsafe set* $\mathbf{X}_u$.

There is an implicit assumption that the feedback controller exists and that Problem 1, as stated above, is solvable.

3.3 Construction of Density Function

The a.e. navigation problem, as stated in Problem 1, is solved using the navigation density function. The construction of the navigation density is inspired by the work of [40, 48, 49]. The navigation measure, as introduced in [48], has a physical interpretation of occupancy, where the measure of any set is equal to the occupancy of the system trajectories in the set, as shown in Figure 3.1. Hence, zero occupancy in a set implies system trajectories not occupying that particular set. So by ensuring that the navigation measure is zero on the obstacle set and maximum on the target set, it is possible to induce dynamics whereby the system trajectories will reach the desired target set while avoiding the obstacle set. We exploit this occupancy-based interpretation in the construction of analytical density functions.

We start with the construction of the unsafe set, where the boundary of the unsafe set is described in terms of the zero-level set of a function. Let $h_k(\mathbf{x})$ be a continuous scalar-valued function for $k = 1, \ldots, L$ such that the set $\{\mathbf{x} \in \mathbf{X} : h_k(\mathbf{x}) \leq 0\}$, is connected with only one component. Thus, the unsafe set $\mathbf{X}_{u_k}$ is defined using the function $h_k(\mathbf{x})$ as follows

$$\mathbf{X}_{u_k} := \{\mathbf{x} \in \mathbf{X} : h_k(\mathbf{x}) \leq 0\}. \tag{3.2}$$

Next, we define a transition region $\mathbf{X}_{s_k}$, which encloses the unsafe set $\mathbf{X}_{u_k}$. Let $s_k(\mathbf{x})$ be a continuous scalar-valued function for $k = 1, \ldots, L$ such that the set $\{\mathbf{x} \in \mathbf{X} : s_k(\mathbf{x}) = 0\}$ defines the boundary of this

transition region. Then the transition region can be defined by the following set

$$\mathbf{X}_{s_k} := \{\mathbf{x} \in \mathbf{X} : s_k(\mathbf{x}) \leq 0\} \setminus \mathbf{X}_{u_k}. \tag{3.3}$$

We propose a navigation density function of the form

$$\rho(\mathbf{x}) = \frac{\prod_{k=1}^{L} \Psi_k(\mathbf{x})}{V(\mathbf{x})^{\alpha}}. \tag{3.4}$$

Here, the function $V(\mathbf{x})$ is the distance function that measures the distance from state $\mathbf{x}$ to the target set, (i.e., the origin), and α is a positive scalar. In this paper, we assume $V(\mathbf{x})$ to be of the form $V(\mathbf{x}) = \|\mathbf{x}\|^2$. Additionally, $\Psi_k(\mathbf{x})$ is a smooth $\mathscr{C}^{\infty}$ function that captures the geometry of the unsafe set $\mathbf{X}_{u_k}$ and can be constructed using the following sequence of functions. We first define an elementary $\mathscr{C}^{\infty}$ function f as follows

$$b(\tau) = \begin{cases} \exp\left(\frac{-1}{\tau}\right), & \tau > 0 \\ 0, & \tau \leq 0 \end{cases}, \tag{3.5}$$

where $\tau \in \mathbb{R}$ [47]. Next, we construct a smooth version of a step function $\bar{f}$ from f as follows

$$\bar{b}(\tau) = \frac{b(\tau)}{b(\tau) + b(1-\tau)}. \tag{3.6}$$

Here, $\bar{b}$ serves as the elementary function for representing zero and nonzero occupation through density. Furthermore, the form of the elementary function, $\bar{b}$, is chosen to ensure that the gradient of the density function is well-defined. To incorporate more general geometric information about the environment, we define a change of variables such that $\phi_k(\mathbf{x}) = \bar{b}\left(\frac{h_k(\mathbf{x})}{h_k(\mathbf{x}) - s_k(\mathbf{x})}\right)$. The resulting function $\Phi_k(\mathbf{x})$ take the following form,

$$\Phi_k(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \mathbf{X}_{u_k} \\ \phi_k(\mathbf{x}), & \mathbf{x} \in \mathbf{X}_{s_k} \\ 1, & \text{otherwise.} \end{cases} \tag{3.7}$$

Finally, the function $\Psi_k(\mathbf{x})$ is defined as

$$\Psi_k(\mathbf{x}) = \Phi_k(\mathbf{x}) + \theta, \tag{3.8}$$

where $\theta > 0$ is some positive parameter. The parameters θ and α are introduced in the construction of the navigation density. The physical significance of these parameters and the assumption made on these parameters and functions are stated in the following remark.

Remark 1

- *The distance function $V(\mathbf{x})$ can be modified to adapt to the geometry of the underlying configuration space. For a Euclidean space with $\mathbf{x} \in \mathbb{R}^n$, we pick $V(\mathbf{x}) = \|\mathbf{x}\|^2$.*
- *The parameter α is used to control the sharpness of the distance function and is used in the proof of the main convergence results.*
- *The function $\Psi_k(\mathbf{x})$ is a θ shifted version of inverse bump function $\Phi_k(\mathbf{x})$ and hence strictly positive i.e., $\Psi_k(\mathbf{x}) \geq \theta > 0$ for $k = 1, \ldots, L$.*
- *$\Psi_k(\mathbf{x})$ makes a smooth transition from θ to $1+\theta$ in the transition region $\mathbf{X}_{s_k}$.*
- *The transition region, $\mathbf{X}_{s_k}$, acts as a sensing region for system trajectories where they start to react to the unsafe set. We refer to the transition region as the sensing region for the rest of this paper.*
- *$h_k(\mathbf{x}) = 0$ defines the boundary of the unsafe set and $s_k(\mathbf{x}) = 0$ defines the boundary of the sensing region. Refer to Figure 3.2 for an illustrative example. In the simplest case, the function $s_k(\mathbf{x})$ can be chosen to be synonymous to $h_k(\mathbf{x})$, such that $h_k(\mathbf{x}) - s_k(\mathbf{x}) = \sigma$ (where $\sigma > 0$ is a constant) uniformly scales the unsafe set to form a sensing region.*

We assume explicit bounds on the functions Ψ_k, V, and their derivatives which follow from the construction of the density function in equation (3.4). It is important to emphasize that it is not necessary to estimate these bounds, but the existence of these bounds is used as part of the proof of the main results of this paper.

Assumption 1

1. *We assume that the distance between the initial set, the target set, and the unsafe sets are all bounded away from zero by some positive constant, say ζ.*

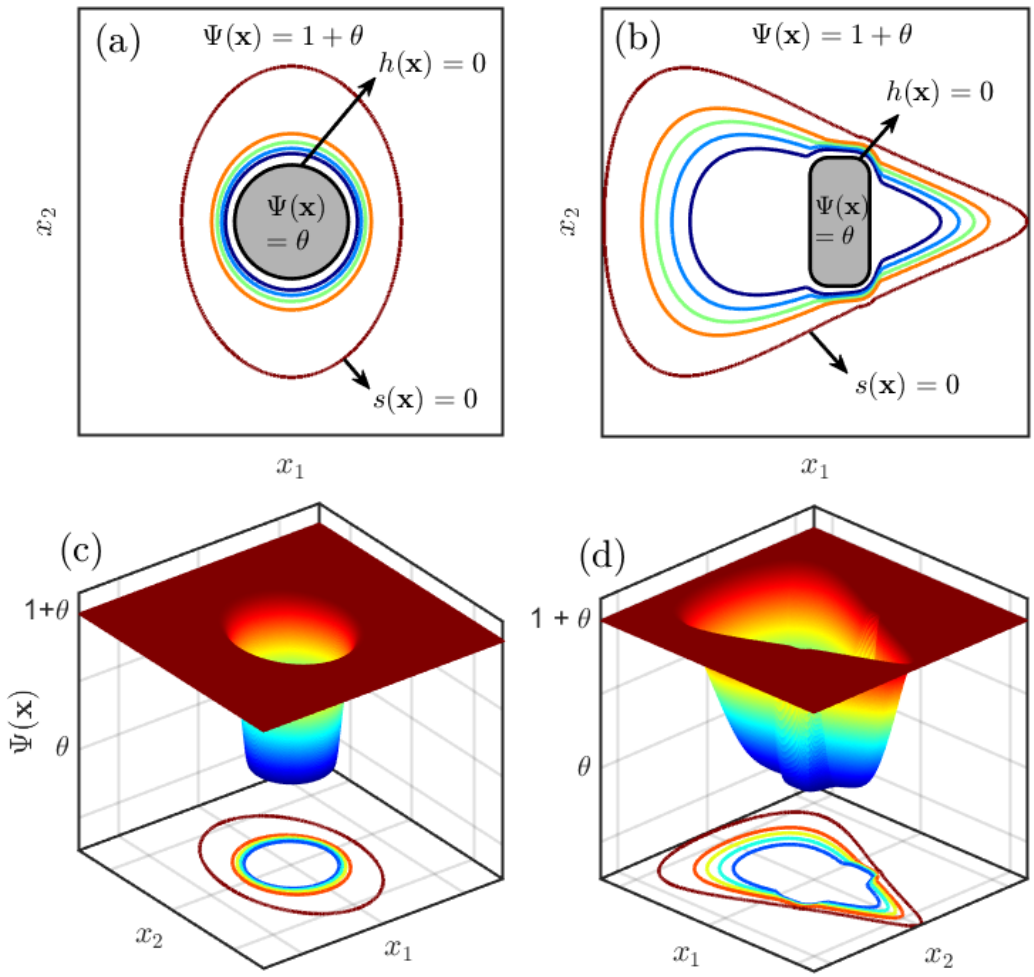

Figure 3.2: Examples of inverse bump function $\Psi(x)$ for (a) $\mathbf{X}_u$ as a circle ($h(\mathbf{x}) = ||\mathbf{x}||^2 - r_1^2 \leq 0$) and transition region boundary as an ellipse ($s(\mathbf{x}) = ||\mathbf{a}\mathbf{x}||^2 - r_2^2 = 0$ where $r_2 > r_1$ and $\mathbf{a}$ is a scaling vector), (b) $\mathbf{X}_u$ as a rounded square ($h(\mathbf{x}) = ||\mathbf{x}||^4 - r_1^4 \leq 0$) and a transition region ($s(\mathbf{x}) = a^2x_1^2 + b^2x_2^2c^{x_1} - r_2^2$ where $r_2 > r_1$; a, b, c are parameters) defined using equation (3.4), (c-d) 3D view of (a) and (b) respectively.

2. *For* $\mathbf{x} \in \mathbf{X}_{u_k}$, *let*

$$V_{min}^k = \min_{\mathbf{x} \in \mathbf{X}_{u_k}} V(\mathbf{x}) > 0. \tag{3.9}$$

Since the distance between the unsafe set and the target set is bounded away from zero, the above quantity is well-defined and greater than zero.

3. *Furthermore,* $m(\mathbf{X}_{u_k})$, *i.e., the Lebesgue measure of the unsafe set, is assumed to be finite, with* θ

satisfying the following inequality for any given $\varepsilon > 0$,

$$\theta \leq \frac{V_{min}^k}{m(\mathbf{X}_{u_k})}\varepsilon, \quad k = 1, \ldots, L. \tag{3.10}$$

4. *In the transition region, i.e., $0 < h_k(\mathbf{x}) \leq s_k(\mathbf{x})$, we assume following bounds*

$$\underline{c}_V \leq V(\mathbf{x}) \leq \bar{c}_V, \quad \underline{c}_{V_x} \leq \left|\frac{\partial V}{\partial x_j}\right| \leq \bar{c}_{V_x}, \quad \left|\frac{\partial^2 V}{\partial x_j^2}\right| \leq \bar{c}_{V_{x^2}}$$

$$\left|\frac{\partial \Psi}{\partial x_j}\right| \leq \bar{c}_{\Psi_x}, \quad \left|\frac{\partial^2 \Psi}{\partial x_j^2}\right| \leq \bar{c}_{\Psi_x^2}, \quad j = 1, \ldots, n.$$

Further, by construction, both the first and second derivatives of Ψ w.r.t. x_j are zero outside the transition region.

5. *Outside the transition region and in $\mathbf{X}_1$, we assume*

$$\frac{\partial^2 V}{\partial x_j^2} \leq \bar{d}_{V_x^2}, \; V \leq \bar{d}_V \|\mathbf{x}\|^2, \; \left|\frac{\partial V}{\partial x_j}\right| \geq \underline{d}_{V_x} \|\mathbf{x}\| \; j = 1, \ldots, n.$$

We have used lower bar, $\underline{c}$, $\underline{d}$, and upper bar, $\bar{c}$, $\bar{d}$, notations to help define the lower and upper positive bounds on functions. The subscripts for c and d signify the corresponding functions.

3.4 Almost Everywhere Navigation Using Density Functions

Given the construction of $\rho(\mathbf{x})$ in (3.4), we design a controller for navigation as the positive gradient of $\rho(\mathbf{x})$, i.e.,

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{k}(\mathbf{x}) = \nabla\rho(\mathbf{x}) \\ &= \left(-\frac{\alpha}{V^{\alpha+1}}\frac{\partial V}{\partial \mathbf{x}}\prod_{k=1}^{L}\Psi_k(\mathbf{x}) + \frac{1}{V^{\alpha}}\frac{\partial}{\partial \mathbf{x}}\prod_{k=1}^{L}\Psi_k(\mathbf{x})\right)^{\top}. \end{aligned} \tag{3.11}$$

Remark 2 *We make following modification to (3.11) to ensure that the vector field is well-defined and the origin is locally asymptotically stable in $\mathscr{B}_\delta$. $\dot{\mathbf{x}} = \left[1 - \bar{b}(\tau)\right]\nabla\rho(\mathbf{x}) - \bar{b}(\tau)\mathbf{x}$ where, $\bar{f}$ is as defined in (3.6).With this modification we will continue to work with (3.11) with the assumption that the origin is locally asymp-*

totically stable in $\mathscr{B}_\delta$ for (3.11).

The main result of the paper is given in the following theorem.

Theorem 1 *Under Assumption 1, the dynamical system (3.11) will solve the a.e. navigation problem as stated in Problem 1.*

Proof of this main theorem is differed to Appendix A. The feedback controller design for the a.e. navigation problem is illustrated in pseudo-code in Algorithm 1[1].

Algorithm 1 Density-based Navigation Algorithm

Input: $\mathbf{X_0}, \mathbf{X_u}, \mathbf{X_T}$
$\Psi(\mathbf{x}) \leftarrow 1$
Define $V(\mathbf{x})$ according to configuration
for X_{u_k} in $\mathbf{X_u}$ **do**
  Define $h_k(\mathbf{x})$ and $s_k(\mathbf{x})$ (see Remark 1 and 3)
  Form $\Psi_k(\mathbf{x})$ from $h_k(\mathbf{x})$ and $s_k(\mathbf{x})$ (see equation 3.7)
  $\Psi(\mathbf{x}) \leftarrow \Psi(\mathbf{x}) \times \Psi_k(\mathbf{x})$
end for
$\rho(\mathbf{x}) = \frac{\Psi(\mathbf{x})}{V(\mathbf{x})^\alpha}$
$u = \nabla\rho(\mathbf{x})$

The rest of the section showcases the navigation results using the controller designed from the analytical density function. We first show the characteristics of the proposed controller, which validates the a.e. navigation properties. Then, we extend our feedback controller to a more complex environment. Lastly, a comparison of our algorithm to NFs is presented.

3.4.1 Characteristics of Density Functions

In this example, we demonstrate the a.e. navigation properties of the proposed controller. The navigation problem is defined with the target set at $\mathbf{X}_T = (4, -3)$ and the unsafe set $\mathbf{X}_u$, which is constructed using a circular inverse bump function with $h(\mathbf{x}) = ||\mathbf{x}||^2 - r_1^2$ and $s(\mathbf{x}) = ||\mathbf{x}||^2 - r_2^2$ with $r_1 = 2$ and $r_2 = 3$. Hence, $\mathbf{X}_{s_k}$ for the inverse bump function is defined on the domain $2 < \|\mathbf{x}\| < 3$.

Figure 3.3a illustrates the a.e convergence of the proposed controller with initial conditions set defined by a line at the top left of the environment boundary. The blue contour lines represent the level sets of the density function. For this example, all the initial conditions starting on the set $\{\mathbf{X}_0 \subset \mathbf{X} : m(\mathbf{X}_0) = 0\}$, which is polar opposite of the target set, cannot converge. This set of initial conditions constitutes a measure

[1]Code implementation details can be seen through the following link: `https://github.com/clemson-dira/density_feedback_control`

zero set. Furthermore, these initial conditions are attracted to a saddle point, implying the existence of local maxima (shown in Figure 3.3b). Note that the existence of a saddle point will imply the existence of local maxima. Any other trajectory starting from an initial condition perturbed from the zero-measure set converges to the target set $\mathbf{X}_T$ while avoiding the obstacle set $\mathbf{X}_u$. Furthermore, we look at the characteristics of initial conditions starting outside the sensing region, defined as a state $\mathbf{x}$ such that $s(\mathbf{x}) \geq 0$ (trajectory A), and within the sensing region, defined as a state $\mathbf{x}$ such that $0 < h(\mathbf{x}) < s(\mathbf{x})$ (trajectory B), shown in Figure 3.3c. The gradients of the density function $\rho(\mathbf{x})$ are such that trajectory A starts to react as it enters the sensing region while trajectory B is repelled outward towards the boundary of the sensing region before converging to the target set (see Figure 3.3d).

3.4.2 Complex environment

One of the main features of our proposed navigation density is that it can incorporate complex shapes of the obstacle set, which is captured in terms of the unsafe set by some appropriate function $h_k(\mathbf{x})$. The unsafe set $\mathbf{X}_u \in \mathbb{R}^2$ in Figure 3.4a is constructed using an implicit function that geometrically represents a circle, an ellipse, an oval, and a bowtie. We show that the initial conditions starting along the boundary converge to the goal at the center while safely avoiding obstacles. The proposed controller can also satisfy a.e navigation in complex maze-like environments. Figure 3.4b shows a trajectory finding a tight feasible region between two obstacles while navigating to the target set. Furthermore, this can be easily extended to navigation problems in higher dimensions. Figure 3.4c shows all trajectories starting from a plane converging to the target set while avoiding obstacles represented as 3D spheres. Figure 3.4d shows navigation with unsafe sets composed of two tori, an unbounded cylinder, and a sphere. We note that unlike [11,30], the construction of the density function naturally admits any complex shapes.

3.4.3 Comparison to Navigation Functions

In this section, we compare the a.e. convergence property of artificial potential field NF to the proposed density functions in a complex environment as shown in Figure 3.5. More specifically, we compare the tuning of $s(\mathbf{x})$ for a.e. convergence in the density function formulation shown in equation (3.4) to the tuning of $\kappa \in \mathbb{R}$ for a.e. convergence in NFs proposed in [42, Ch. 3, p. 36],

$$\psi_k(\mathbf{x}) = \frac{||\mathbf{x}-\mathbf{x}_g||_2^2}{\left(||\mathbf{x}-\mathbf{x}_g||_2^{2\kappa} + \beta(\mathbf{x})\right)^{1/\kappa}}, \tag{3.12}$$

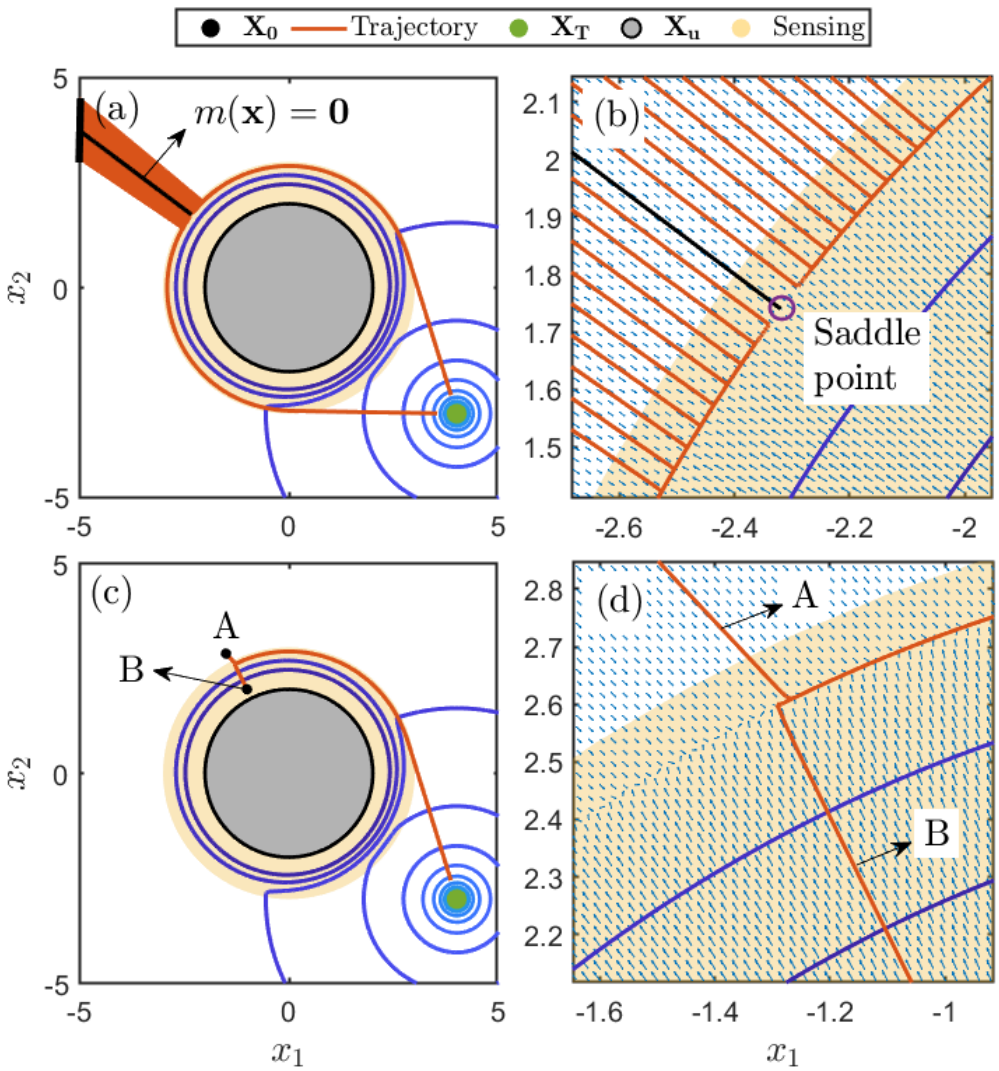

Figure 3.3: (a) Trajectories converge to the target (green) while avoiding the unsafe set (gray), (b) Initial conditions along the zero-measure set (black) converge to a saddle point (purple), (c) Trajectories starting at A ($s(\mathbf{x}) > 0$) and B (in $\mathbf{X}_{s_k}$) converge to the target set, (d) Trajectories follow the same path near the boundary of $s(\mathbf{x})$.

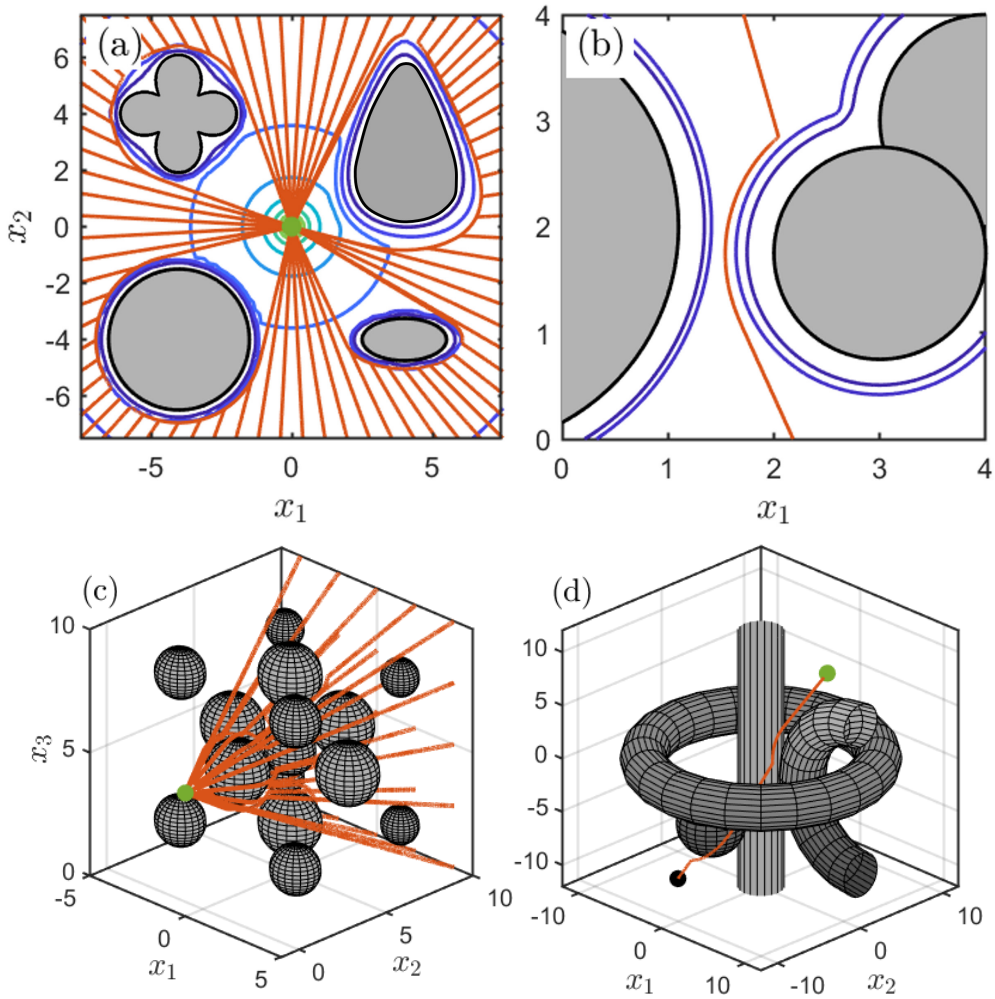

Figure 3.4: (a) Trajectories converge to the target set (green) while avoiding obstacles (gray), (b) Trajectory finding a narrow feasible region around obstacles, (c) Navigation in a spherical grid, (d) Navigation through two tori, cylinder, and sphere.

where $\mathbf{x}_g$ is the desired goal location, $\beta(\mathbf{x})$ is an obstacle function and κ is a tuning parameter.

Although a domain is not necessary in the density formulation, NFs do require a radially bounded sphere world. Hence, we define an appropriate bounded sphere world of radius 25. The authors note that NFs do not make any claims about tuning κ for a.e. convergence other than the sphere world and its extensions [11,42], but for the sake of comparison, we look at an environment with a C-shaped unsafe set. We then look at initial conditions that lie inside the C-shaped unsafe set with the target set defined outside the cavity of the unsafe set.

Figures 3.5a and 3.5b show that the trajectories do not converge to the goal for all random initial

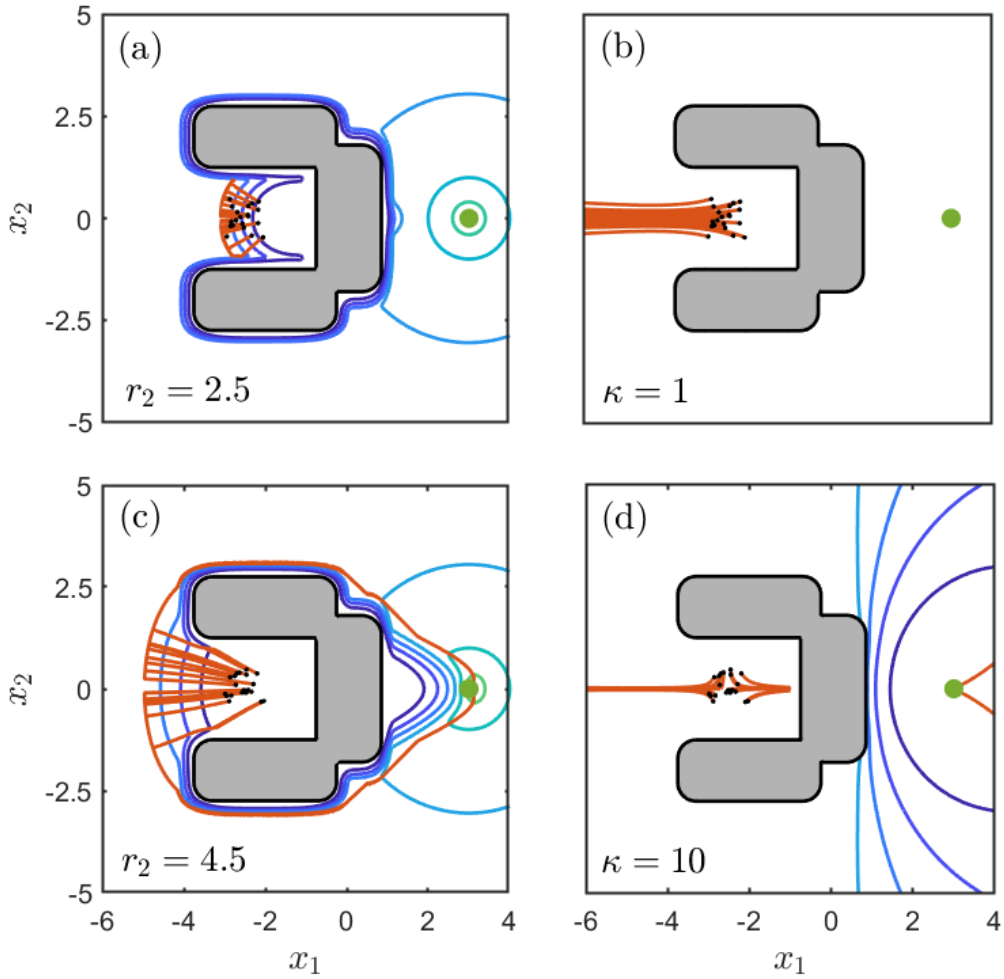

Figure 3.5: Comparison of density functions and NFs for random initial conditions. The sensing region for the density function is defined by $s(\mathbf{x}) = a^2x_1^2 + b^2x_2^2c^{x_1} - r^2$ (r, a, b, c are parameters). For (a) $r = 2.5$, trajectories don't converge, while setting (c) $r = 4.5$ leads to all trajectories converging. NFs with their corresponding tuning parameter for convergence (b) $\kappa = 1$ and (d) $\kappa = 10$ lead to trajectories not converging.

conditions for small values in tuning parameter in either the density function formulation or the artificial potential field NF formulation. This is expected in NF as only large κ in a sphere world guarantees a.e. convergence. Likewise, the density formulation sees the same results. However, tuning $s(\mathbf{x})$ such that the density function formulation has a.e. convergence property is intuitive, as stated below in Remark 3. This is shown in Figure 3.5c, where tuning $s(\mathbf{x})$ to be larger than the C-shaped unsafe sets results in all system trajectories converging to the target set. Note, no explicit mapping to a simplistic unsafe set (e.g., circle) is required, where the same cannot be stated for NFs (even with high κ), which does not give a.e. convergence results for complex unsafe sets. This can be seen in Figure 3.5d, where some trajectories exit the unsafe set

and converge to the goal (by taking a large curvature path) while others get trapped inside the cavity of the unsafe set.

Remark 3 *The tuning parameter in the design of the navigation density functions are α, and $s_k(\mathbf{x})$. The tuning of α depends on the rate of convergence of the trajectories. The convergence proof of the main theorem relies on the argument of using large value of α but in practice small value of alpha works in simulation. The tuning of $s_k(\mathbf{x})$ is physically intuitive, as it signifies the sensing region. Hence, a sensing region that encompasses the unsafe set with a sufficiently curved convex set has worked in the simulations.*

3.4.4 Comparison to CLF-CBF-QP

Similarly to the analytical density function that guarantees safety and convergence, control Lyapunov function with control barrier function using quadratic programming (CLF-CBF-QP) guarantees safety and convergence by enforcing both a control Lyapunov stability criterion (control counterpart to Lyapunov stability) for convergence and a forward-invariance criterion for safety [4]. This forward invariance of safety is enforced by defining a barrier function on a safe set C, which we denote here as $B(\mathbf{x})$, that satisfies the following conditions,

$$\begin{aligned} C &= \{x \in D \subset \mathbb{R}^n : B(\mathbf{x}) \geq 0\}, \\ \partial C &= \{x \in D \subset \mathbb{R} : B(\mathbf{x}) = 0\}, \\ \mathrm{Int}(C) &= \{x \in D \subset \mathbb{R}^n : B(\mathbf{x}) > 0\}. \end{aligned} \tag{3.13}$$

The barrier function $B(\mathbf{x})$ must also satisfy safety invariance by enforcing the minimally restrictive linear constraint

$$\dot{B}(\mathbf{x}, \mathbf{u}) = L_f B(\mathbf{x}) + L_g B(\mathbf{x})\mathbf{u} \geq -\iota B(\mathbf{x}), \tag{3.14}$$

where $\iota \in \mathbb{R}^+$. See [3] for more information.

We compare our feedback controller to the CBF-CLF-QP with the appropriate distance function as shown in Figure 3.6 and 3.7. We note that for small ι, (3.14) is minimally restrictive and, therefore, smoothly diverges from the obstacle set while converging to the target. However, for larger ι, i.e. more restrictive barrier certification $\dot{B} \geq 0$, the trajectories shown are similar to the analytical density function solution. However, the difference lies in that the navigation density controller abruptly reacts in the transition region while the CBF controller reacts at the boundary of the unsafe set.

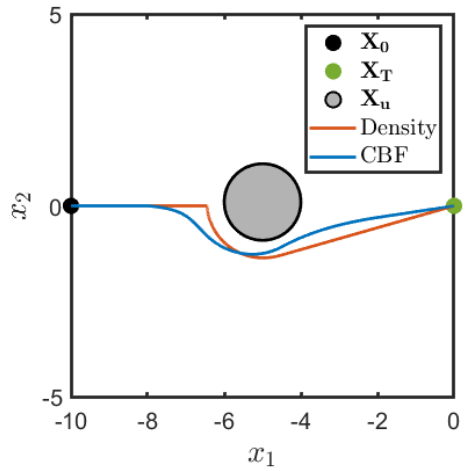

Figure 3.6: Comparison of CLF-CBF-QP and Analytical Density Functions for $\mathbf{X}_u$ w/ center at $(-5,0)$, $\mathbf{X}_0 = (-10,0)$ and $\mathbf{X}_T = (0,0)$. Here our density function is defines $h(\mathbf{x})$ and $s(\mathbf{x})$ as circles with $r_1 = 1$ and $r_2 = 1.5$. For the CBF, $B(\mathbf{x})$ is defined as a distance function with $\iota = 1$.

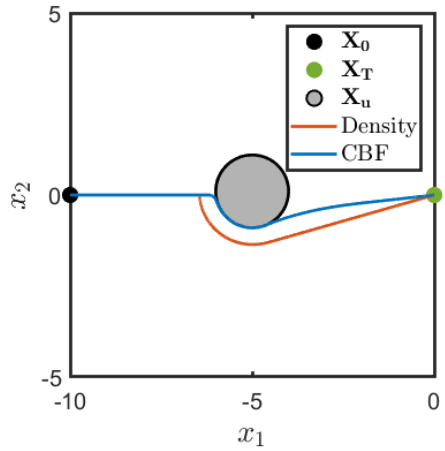

Figure 3.7: Comparison of CLF-CBF-QP and Analytical Density Functions for $\mathbf{X}_u$ w/ center at $(-5,0)$, $\mathbf{X}_0 = (-10,0)$ and $\mathbf{X}_T = (0,0)$. Here our density function is defines $h(\mathbf{x})$ and $s(\mathbf{x})$ as circles with $r_1 = 1$ and $r_2 = 1.5$. For the CBF, $B(\mathbf{x})$ is defined as a distance function with $\iota = 10$.

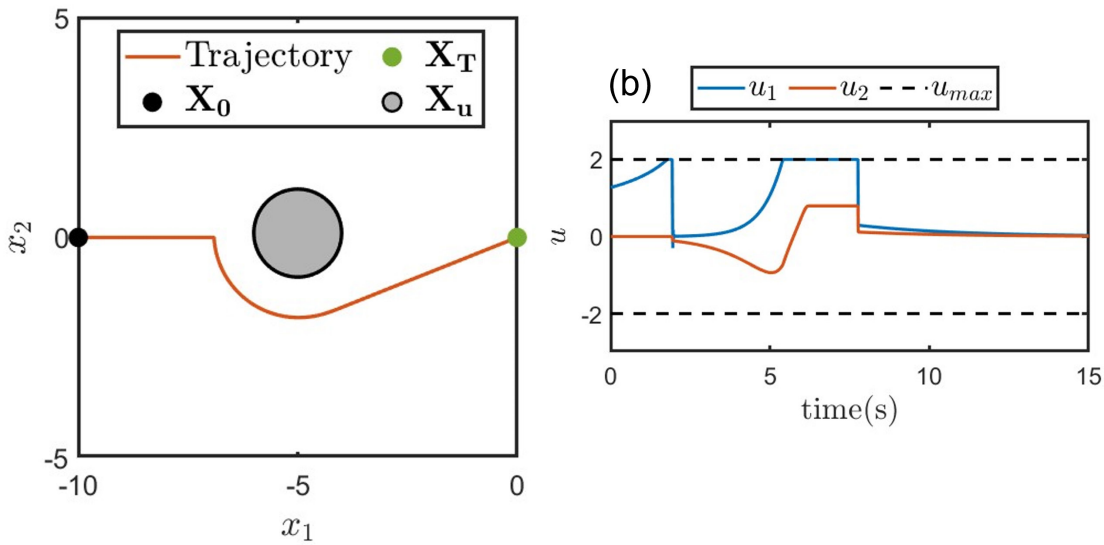

Figure 3.8: (a) States converge to the target and (b) control trajectories remain within the constraints

3.5 Application to Robotic Systems

In this section, we show applications of the proposed density functions where the system is under control constraints, the system has process noise, and the navigation results for a fully actuated robotic systems.

3.5.1 Constrained Control w/ Density Function

We consider a system with constraints in the following form

$$\dot{\mathbf{x}} = \mathbf{u} = \nabla\rho(\mathbf{x}), \quad \mathbf{u} \in [-u_{max}, u_{max}], \tag{3.15}$$

where u_{max} is the bound on control. More specifically, we constrain the control when $||\mathbf{u}||_\infty > u_{max}$ by normalizing the control

$$\bar{\mathbf{u}} = \frac{\mathbf{u}}{||\mathbf{u}||_\infty} u_{max}, \tag{3.16}$$

where $\bar{\mathbf{u}}$ is the constrained control. Figure 3.8 shows that the system trajectories converge to the target while satisfying control constraints.

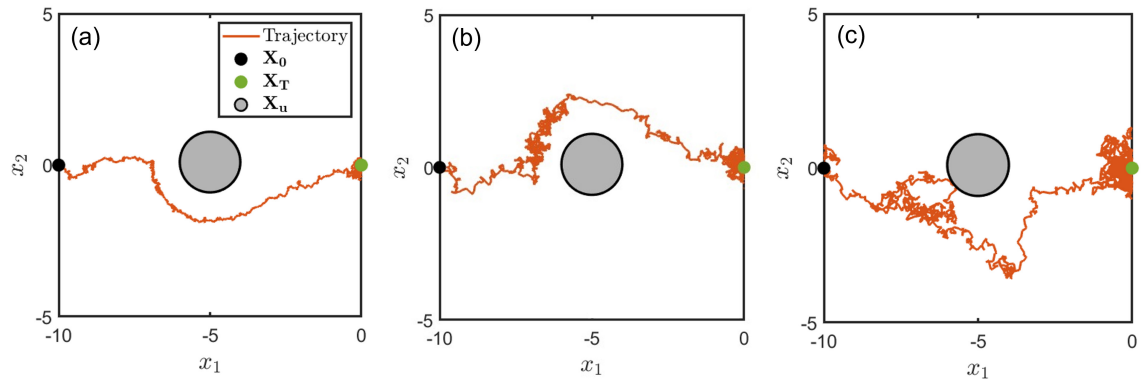

Figure 3.9: Density function w/ gaussian white noise of $\mu = 0$ and (a) $\Sigma = 10^{-3} \times I_2$ (b) $\Sigma = 5 \times 10^{-3} \times I_2$ (c) $\Sigma = 10^{-2} \times I_2$

3.5.2 Performance of Density Function w/ Noise

We consider the performance of our controller in a stochastic setting where noise is entered through the control input

$$\dot{\mathbf{x}} = \mathbf{u} + \mathbf{w} \quad \mathbf{u} \in [-u_{max}, u_{max}], \tag{3.17}$$

where $\mathbf{w} \in \mathcal{N}(\mu, \Sigma)$ is the gaussian white noise with mean $\mu = 0$ and covariance Σ. Figure 3.9 showcases the navigation problem with control noise for varying levels of covariance.

We see that the feedback controller is capable of invariance while converging towards the goal with noise. Although the invariance of our control law is not guaranteed, we see that up to a certain bound on the noise, the control performs reasonably well.

3.5.3 Fully Actuated Robotic System

We extend the density function presented in Section 3.3 to a general class of robotic systems. For a robot with n joints and n rigid links, the system's dynamics can be expressed using the Euler-Lagrange equations. Consider an unconstrained system where $\mathbf{M}(\mathbf{q})$ is the inertia matrix and $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$ represents the Coriolis and gravity effects on the system, $\mathbf{q} \in \mathbb{S}^1 \times \mathbb{S}^1$. Then the corresponding system is represented as follows

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u}. \tag{3.18}$$

We then take a similar approach outlined in [24] in which there exists an equivalent "planning" system defined by $\dot{\mathbf{q}} = \nabla\rho(\mathbf{q})$ and a control law given by $\mathbf{u} = \nabla\rho(\mathbf{q}) + d(\mathbf{q}, \dot{\mathbf{q}})$ ($d(\mathbf{q}, \dot{\mathbf{q}})$ is a dissipative term and

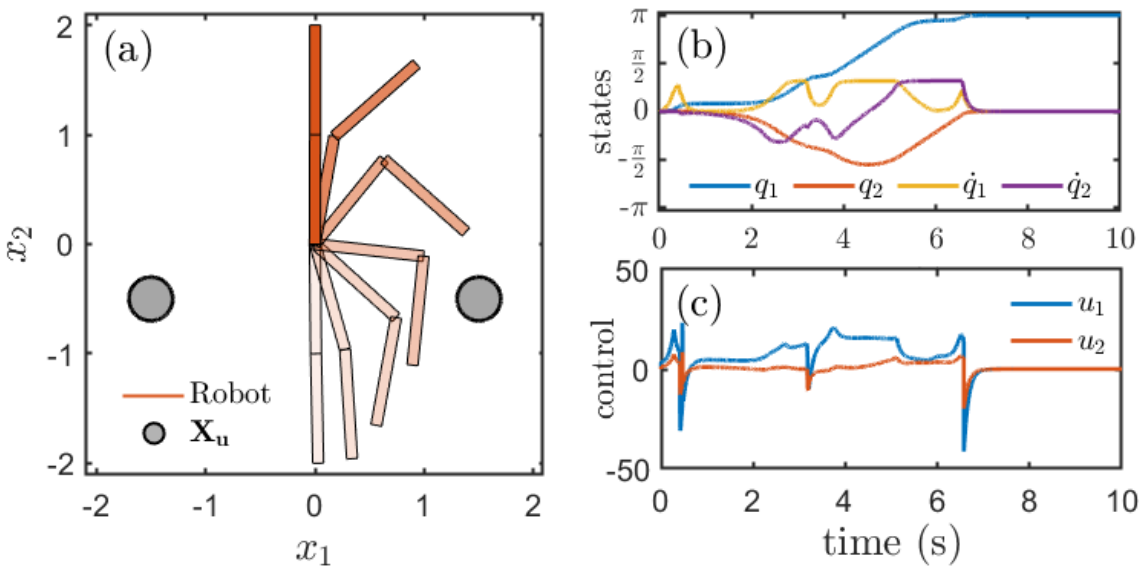

Figure 3.10: (a) Robot (red) converges to the goal $(\pi, 0)$ starting from equilibrium (0,0) while avoiding obstacles (gray). (b) state trajectories of the robot and (c) control inputs for executing the swing-up maneuver.

$\dot{\mathbf{q}}^\top d(\mathbf{q}, \dot{\mathbf{q}}) < \mathbf{0}$), where the system defined in (3.18) tracks the planning system asymptotically [24]. For a general robotic system such as the system defined in equation (3.18), $d(\mathbf{q}, \dot{\mathbf{q}})$ can be selected such that it cancels out the nonlinearities of the system similar to the inverse dynamics approach. Therefore, we define a density-based inverse dynamics controller given by

$$\mathbf{u}_\rho = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_\mathbf{d} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{M}(\mathbf{q})\Big(\mathbf{K_p}\nabla\rho(\mathbf{e}) - \mathbf{K_v}\dot{\mathbf{e}}\Big), \tag{3.19}$$

where $\mathbf{e} := \mathbf{q} - \mathbf{q_d}$, $\dot{\mathbf{e}} := \dot{\mathbf{q}} - \dot{\mathbf{q}}_\mathbf{d}$, $\mathbf{q_d}$ is the desired reference trajectory to follow, and $\mathbf{K_p}$ and $\mathbf{K_v}$ are positive definite gain matrices.

Figure 3.10a shows a fully actuated two-link planar robotic arm executing a swing-up maneuver with $\mathbf{K_p} = \text{diag}([1,1])$, $\mathbf{K_v} = \text{diag}([10,10])$ and $V(\mathbf{q}) = (1-\cos(q_1))(1-\cos(q_2))$. The mass and length of each link are set to unity. The task space obstacles (circular with a radius of 0.2) are mapped to joint space and approximated using inverse bump functions. The reference trajectories are obtained in joint space based on the planning system $\dot{\mathbf{q}} = \nabla\rho(\mathbf{q})$. The corresponding state and control trajectories are shown in Figures 3.10b and 3.10c, respectively. It is seen that the density-based inverse dynamics controller drives the two-link manipulator to the upright position while avoiding the obstacle set.

3.6 Conclusions

This chapter provides an analytical construction for the navigation density. Moreso, we prove that the navigation density solves the almost everywhere navigation problem. The proposed navigation density can be viewed as dual to the popular navigation function and is derived based on the occupancy-based interpretation of the density function. The navigation density has a few advantages compared to navigation functions. Unlike navigation functions, which are hard to construct, navigation density can be easily constructed. Furthermore, the density function formulation can incorporate arbitrary shapes of the unsafe set. We provide simulation results for navigation using density function in complex and high dimensional environments as demonstrated. Lastly, we also demonstrate the application of the density function for control on a robotic system with safety constraints.

Chapter 4

Safe Navigation of Quadruped Using Density Function

This chapter is adapted from a paper in *Indian Control Conference*:

> Sriram S. K. S. Narayanan, Andrew Zheng, and Umesh Vaidya, "Safe Motion Planning for Quadruped Robots Using Density Functions," in Indian Control Conference.

4.1 Introduction

Quadruped research has grown considerably over the past few years [51]. This was highlighted in the DARPA Subterranean Challenge, where top contenders used legged robots as a core component of their underground autonomy challenge [46]. It is noted that the legged locomotion problem for these systems is typically decomposed into smaller subproblems [13, 32] as shown in Figure 4.1. Therefore, under certain assumptions on the model and the motion of the dynamical systems, it is seen that the quadruped dynamics can be reduced to a low-order fidelity model and used as a motion plan [52]. In fact, [32] shows that the motion planning problem for legged locomotion can be utilized in a model-free planning problem under asymptotic assumptions. Naturally, as a consequence, the generated reference trajectory cannot directly be utilized to control the robot's motion. Therefore, the low-fidelity trajectories must be mapped to a higher-fidelity model. A standard method used is utilizing an optimizer in the form of a model predictive controller

to generate high-fidelity solutions that enforce dynamically feasible constraints from a given reference.

Therefore, inspired by [32], this chapter uses the density functions proposed in [54] as a tool to design safe high-level trajectories for quadruped locomotion. More specifically, a modular navigation architecture is designed such that any planning algorithm can be utilized for the legged locomotion problem. Furthermore, the navigation density controller is reformulated as a navigation feedback density planner for the motion planning problem of a quadruped. These low-fidelity trajectories are optimized using a nonlinear model predictive controller (NMPC) to obtain locally optimal reference contact forces to track the trajectories. Finally, the contact forces are converted to joint torques using a whole-body controller.

The rest of the paper is organized as follows. First, we review this chapter's notation and problem statement in Section 4.2. Then, we give an overview of the legged locomotion problem in Section 4.3. We followed this with our navigation framework for our novel density navigation function in Section 4.4, where we show the results of the algorithm in simulation (Section 4.5 and hardware (Section 4.6. Finally, Section 4.7 are concluding remarks about the current limitations and future directions.

4.2 Notations and Problem Statement

Here, we restate the underlying notations that was mentioned in the previous chapter for clarity and independency in this chapter. Furthermore, we give an informal problem statement about navigation applied onto the quadruped. This is a modified version to the original problem statement. See Problem 1 to see how we originally addressed the a.e. navigation problem.

Notations: The following notations will be used in this chapter. $\mathbb{R}^n$ denotes the n dimensional Euclidean space, $\mathbf{x} \in \mathbb{R}^n$ denotes a vector of system states, $\mathbf{u} \in \mathbb{R}^n$ is a vector of control inputs. Let $\mathbf{X} \subset \mathbb{R}^n$ be a bounded subset that denotes the workspace for the robot. $\mathbf{X}_0$, $\mathbf{X}_T$, $\mathbf{X}_{u_k} \subset \mathbf{X}$, for $k = 1, \ldots, L$ denote the initial, target, and unsafe sets, respectively. With no loss of generality, we will assume that the target set is a single point set located at the origin, i.e., $\mathbf{X}_T = \{0\}$. $\mathbf{X}_u = \cup_{k=1}^{L} \mathbf{X}_{u_k}$ defines the unsafe set and $\mathbf{X}_s := \mathbf{X} \setminus \mathbf{X}_u$ defines the safe set. We use $\mathscr{C}^k(\mathbf{X})$ to denote the space of all k-times differentiable functions of $\mathbf{x}$. We use $\mathscr{M}(\mathbf{X})$ to denote the space of all measures on $\mathbf{X}$ and $m(\cdot)$ to denote the Lebesgue measure. $\mathbb{1}_A(\mathbf{x})$ denotes the indicator function for set $A \subset \mathbf{X}$.

Problem Statement: We wish to design a smooth feedback planner from almost every initial condition from the initial set $\mathbf{X}_0$ to the target set $\mathbf{X}_T$ while avoiding the unsafe set $\mathbf{X}_u$. We then wish to apply a feedback controller to track the safe reference trajectory generated by the planner for the quadruped robot.

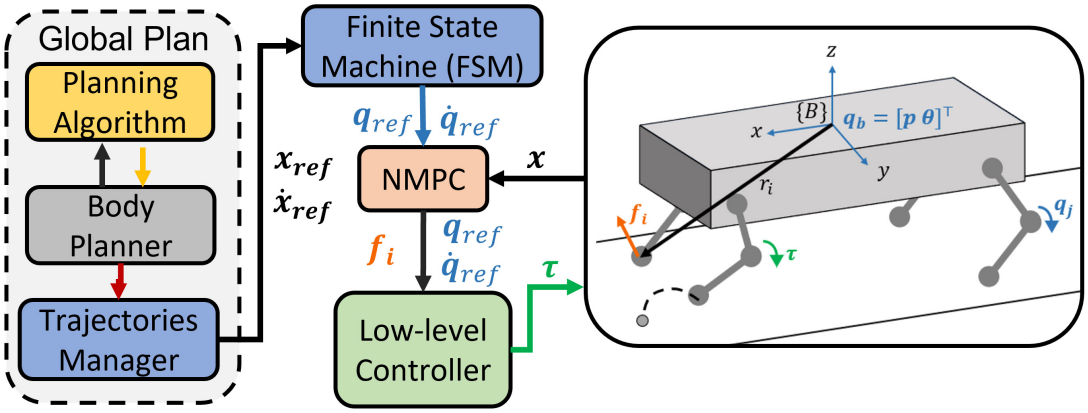

Figure 4.1: Hierarchical Planning and Control Architecture for Quadruped Locomotion.

4.3 Legged Locomotion Overview

Quadruped locomotion is a challenging problem due to its dimensionality and hybrid nature. One standard method of approaching the problem is decomposing the system into smaller subsystems. First, to deal with the hybrid nature, a Finite State Machine (FSM) determines a switching logic that generates the corresponding gait phases. Accordingly, to track a reference trajectory under these different gaits, the robot must exert contact forces from the different feet in stance, propelling the robot forward. This is achieved by finding locally optimal control trajectories $\mathbf{u}^\star(t)$ from a model predictive controller (MPC). Finally, a whole-body controller is used to generate the required joint torques to track the joint torques in stance and swing. The corresponding subsections go into detail about each of the subsystems.

4.3.1 Finite State Machine

To incorporate the hybrid dynamics of a quadruped, apriori knowledge of the contact phases for a quadruped is crucial in reducing the complexity of the problem. Therefore, a finite state machine can be utilized as a generator to define the corresponding gaits of a quadruped. Therefore, the finite state machine defines which of the legs of a system are in contact, which we denote C, and not in contact, which we denote as $\hat{C}$. Knowledge of the contact phases can then be used in the control formulation as shown in the section 4.3.3.

4.3.2 Dynamic Model

Under the assumption that there is adequate control for the fully actuated system (2.15), we can consider (2.14) for our control problem. The unactuated dynamics – or centroidal dynamics – can be rewritten in Newton-Euler form as

$$\dot{\mathbf{L}}_{com} = \begin{bmatrix} \sum_{i=1}^{n_c} \boldsymbol{\lambda}_{c_i} + m\mathbf{g} \\ \sum_{i=1}^{n_c} \mathbf{r}_{com,c_i} \times \boldsymbol{\lambda}_{c_i} \end{bmatrix}, \tag{4.1}$$

where $\mathbf{L}_{com} = [\mathbf{L}_{lin}\ \mathbf{L}_{ang}] \in \mathbb{R}^6$ is the centroidal momentum described through its respective linear and angular components about the centroidal frame[1], $\mathbf{r}_{com,c_i}$ is the position of the contact point c_i of each leg with respect to the center of mass m, $\boldsymbol{\lambda}_{c_i}$ are the contact forces of each leg exerted onto the robot from the environment, and $\mathbf{g}$ is the gravitation vector.

Correspondingly, to model the rate of change of the generalized coordinates, we can use the centroidal momentum matrix (CMM) $\mathbf{A}(\mathbf{q}) := [\mathbf{A}_b(\mathbf{q})\ \mathbf{A}_j(\mathbf{q})] \in \mathbb{R}^{6\times 18}$ as the following

$$\mathbf{L}_{com} = \begin{bmatrix} \mathbf{A}_b(\mathbf{q})\ \mathbf{A}_j(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_b \\ \dot{\mathbf{q}}_j \end{bmatrix}, \tag{4.2}$$

where $\mathbf{A}_b(\mathbf{q})$ and $\mathbf{A}_j(\mathbf{q})$ are the body and joint components of the centroidal momentum matrix. This can then be rewritten into the following form

$$\dot{\mathbf{q}}_b = \mathbf{A}_b^{-1}\Big(\mathbf{L}_{com} - \mathbf{A}_j\dot{\mathbf{q}}_j\Big),$$

where $\mathbf{q}_b = [\mathbf{p}\ \boldsymbol{\theta}] \in \mathbb{R}^6$ is the base pose in the inertial frame and $\mathbf{q}_j \in \mathbb{R}^{12}$ are the joint angles. Thus, with the dynamic relation defined, the equation of motion can be written in state space form

[1]Centroidal frame, here, is the reference frame on the center of mass of the quadruped aligned with the world frame

$$
\begin{aligned}
\dot{\mathbf{L}}_{lin} &= \sum_{i=1}^{n_c} \boldsymbol{\lambda}_{c_i} + m\mathbf{g} \\
\dot{\mathbf{L}}_{ang} &= \sum_{i=1}^{n_c} \mathbf{r}_{com,c_i} \times \boldsymbol{\lambda}_{c_i}. \\
\dot{\mathbf{q}}_b &= \mathbf{A}^{-1}(\mathbf{L}_{com} - \mathbf{A}_j \dot{\mathbf{q}}_j) \\
\dot{\mathbf{q}}_j &= \mathbf{v}_j,
\end{aligned} \tag{4.3}
$$

where we define $\mathbf{x}_r = (\mathbf{h}_{com}, \mathbf{q}_b, \mathbf{q}_j) \in \mathbb{R}^{24}$ as the robot states and $\mathbf{u} = (\boldsymbol{\lambda}_{c_1}, ..., \boldsymbol{\lambda}_{c_{nc}}, \mathbf{v}_j)$ as the input vectors.

4.3.3 Optimal Control Problem Formulation

The optimal control problem for quadruped locomotion can be formulated as a nonlinear MPC proposed in [45] as such

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{u}} \quad & \Phi(\mathbf{x}(T)) + \int_0^T L(\mathbf{x}(t), \mathbf{u}(t), t) dt \\
\text{s.t.} \quad & \dot{\mathbf{x}} = f_h(\mathbf{x}(t), \mathbf{u}(t), t) \\
& \mathbf{p}(\mathbf{x}(t), \mathbf{u}(t), t) = 0 \\
& \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) \geq 0 \\
& \mathbf{x}(0) = \mathbf{x}_0,
\end{aligned} \tag{4.4}
$$

where $f_h(\mathbf{x}(t), \mathbf{u}(t))$ is the hybrid system dynamics in state space form mentioned in (4.3), $\mathbf{x}$ and $\mathbf{u}$ are the vector of state and input decision variables, Φ defines the terminal cost, L defines the stage cost, $\mathbf{p}$ denotes the state and input constraints, $\mathbf{h}$ denotes the inequality constraints, and $\mathbf{x}(0) = \mathbf{x}_0$ denotes the initial conditions. We note that $\mathbf{p}$ and $\mathbf{h}$ are general constraints to the optimization problem, but the core constraints that are expressed in this form for the quadruped problem are the following

$$
\mathbf{p}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{v}_{c_i} = 0 \qquad if\ c_i \in C_i \tag{4.5}
$$

$$
\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) = \mu_s \lambda_{c_i}^z - \sqrt{\lambda_{c_i}^{x^2} + \lambda_{c_i}^{y^2}} \geq 0 \qquad if\ c_i \in C_i \tag{4.6}
$$

where c_i denote the leg in closed contact C, $\mathbf{p}$ denotes the no slip condition in contact, $\mathbf{h}$ denotes the friction constraints.

The constrained optimization problem is solved in closed-loop in a receding horizon fashion using the Sequential Linear Quadratic (SLQ) technique [7].

4.3.4 Low-level Controller

Due to simplifications in the model, a low-level controller is designed to find the closest feasible solution from the model predictive controller through a quadratic program (QP) under full-order model constraints. Information is extrapolated from the solution of the predictive controller to generate control references for the QP. For the leg in swing, joint accelerations are obtained through a transformation of task to joint space accelerations

$$\mathbf{J}_{sw_i}\ddot{\mathbf{q}}_j + \dot{\mathbf{J}}_{sw_i}\dot{\mathbf{q}}_j = \ddot{\mathbf{r}}_{c_i}, \tag{4.7}$$

where $\mathbf{J}_{sw_i}$ is the swing leg Jacobian and $\ddot{\mathbf{r}}$ is the end-effector acceleration. Correspondingly, torque level control is computed to be optimized in the QP from the optimized contact forces:

$$\boldsymbol{\tau}_i^{st} = \mathbf{J}_i^\top \mathbf{R}_i^\top \boldsymbol{\lambda}_{c_i}, \tag{4.8}$$

where $\mathbf{J}_i$ is the stance jacobian, $\mathbf{R}$ is the corresponding rotation matrix. Finally, the solution from the quadratic program is commanded to the driver, where the commanded torque ($\boldsymbol{\tau}_a$) is defined by the following

$$\boldsymbol{\tau}_a = \boldsymbol{\tau}_{ff} + \mathbf{K}_p(\mathbf{q}^\star - \mathbf{q}_j) + \mathbf{K}_d(\dot{\mathbf{q}}^\star - \dot{\mathbf{q}}_j) \tag{4.9}$$

where $\boldsymbol{\tau}_{ff}$ is the feedforward joint torque, $\dot{\mathbf{q}}^\star$ and $\mathbf{q}^\star$ are the optimized joint reference states. More details on the QP formulation can be referenced to [45].

4.4 Feedback Density Planner

To incorporate safe navigation on the legged locomotion problem, we incorporate a hierarchical navigation framework to execute a feedback density plan, where a corresponding model predictive controller (MPC) and whole-body controller find locally optimal joint torques to track the corresponding reference trajectory as seen in Figure 4.1. We again note the usage of simplified model trajectories as a reference to

the control problem, as seen in [33, 52]. Therefore, we first design a motion planning navigation framework that utilizes a bidirectional communication architecture between the planning algorithm and body planner, as shown in the top left image of Figure 4.1, through the ROS messaging interface. This communication interface handles receiving the current observed state from the quadruped sensory information to generate an appropriate plan. Likewise, the plan is sent to the body planner, where the low-fidelity trajectory is transcribed into a centroidal model trajectory. The centroidal model trajectory is then sent to a trajectory manager to handle synchronization conflicts with the FSM and MPC. This navigation architecture allows us the ability to incorporate general motion planning algorithms for the quadruped, such as A*, RRT*, and Zero Moment Point planning. Details of the full implementation are available through code in the GitHub[2].

Furthermore, we highlight the integration details to transcribe the density plan from integrator dynamics to a quadruped. In particular, we incorporate the density feedback controller as a density feedback planner by formulating a discrete time form for our continuous time controller.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \nabla\rho(\mathbf{x})\Delta t \tag{4.10}$$

Furthermore, the discrete time form of the controller (4.10) was forward simulated up to a horizon for reference tracking.

Correspondingly, due to the discretization of a controller in the continuous domain, we smoothen the trajectories through a moving average filter and a first-order filter. The following trajectory is given to the reference tracking MPC and FSM to execute a locally optimal trajectory for legged locomotion.

4.5 Simulation Results

In this section, we show simulation results using Gazebo as the physics engine with RVIZ as the visual interface, as shown in Figure 4.2. The following experiment considers the navigation problem where the target set is at $\mathbf{X}_T = (10, 0)$. Moreover, a circular obstacle $\mathbf{X}_u$ is centered at (5,0.1) with a radius of $r = 0.5 + 0.5$ (the obstacle set incorporates the quadruped geometry by enlarging the radius of the obstacle) and highlight the trajectory taken by the density-navigation on the quadruped. Comparatively, in Figure 4.2c, we consider an experiment with two obstacles centered at (3,0.1) and (7,-1) with the same enlargement of radius to incorporate the geometry of the robot. These parameters are utilized for constructing and synthesizing

[2]https://github.com/AndrewZheng-1011/legged_planner

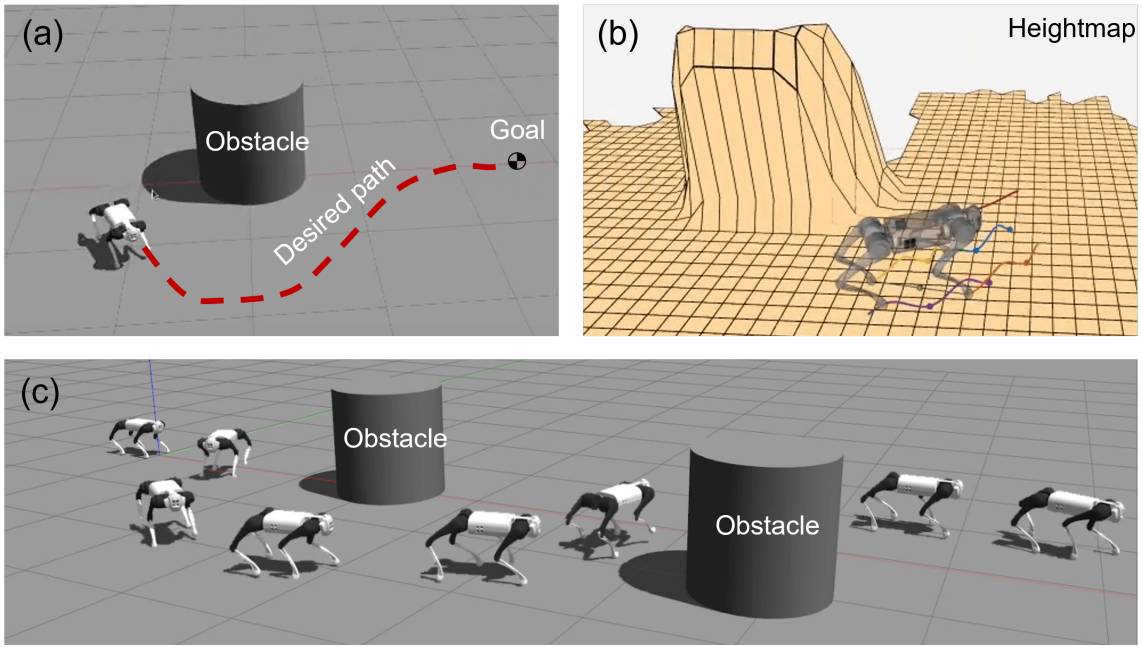

Figure 4.2: Quadruped robot executing the proposed density plan. (a) Robot executing a safe motion plan in an environment with a single obstacle and (b) the corresponding height map representation. (c) safe motion plan for an environment with two obstacles.

a density feedback controller as described in (3.11) with Table 4.1 giving implementation details about the major parameters to execute the feedback planner.

Table 4.1: Feedback Density Plan Parameter on Quadruped

Parameters			
Step size (Δt)[s]	Plan Horizon (N_{hor})	Window Size (N_{win})	Update Frequency (f_{update}) [Hz]
0.1	200	21	20

The resulting trajectory of the center of mass of the quadruped in comparison with the center of mass trajectory from the density is shown in Figure 4.3. Despite the model differences, the model predictive controller tracks the reference trajectory quite well.

Correspondingly, we are able to see the resulting optimized reaction force to track the reference trajectory from the density plan in Figure 4.4. Note, generally, when the quadruped tracks the reference trajectory in Figure 4.3, the optimized ground reaction force $\mathbf{u}$ is seen to be nonzero in the transverse direction. This is because the ground reaction force is the main component in propelling the quadruped in the transverse plane, hence nonzero forces for $\mathbf{f}_{c_i,x}$ and $\mathbf{f}_{c_i,y}$. An example is that for the quadruped to traverse a trajectory from the density plan with nonzero curvature, the lateral ground reaction forces must invert signs. Figure 4.4

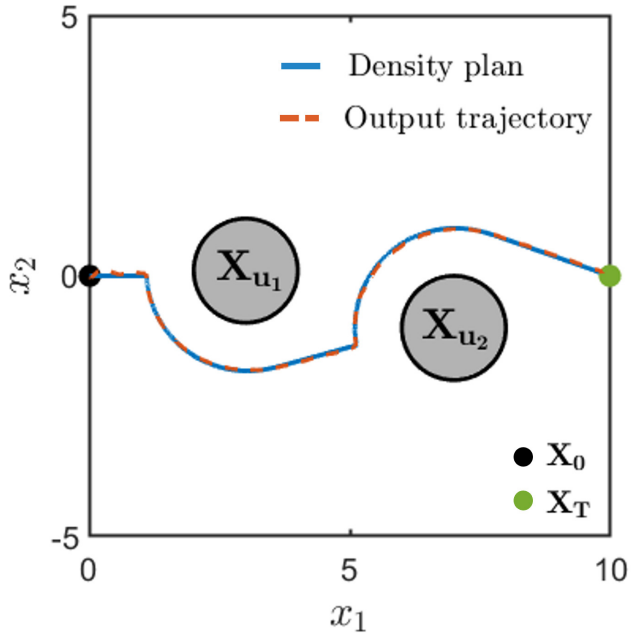

Figure 4.3: Quadruped robot state trajectory (output) with desired density-based plan.

highlighted this statement for $n \in [330, 350]$ iterations, where the quadruped goes around the obstacle.

4.6 Hardware Results

The proposed density-based planner is implemented in hardware. Although the algorithm itself is no different, the implementation in hardware will differ. Therefore, we highlight some implementation details that are necessary to integrate into hardware.

4.6.1 Contact Sensors

Contact sensors are a crucial method to determine which foot is in contact with the ground. The Unitree Go1 uses pressure bag sensors to determine whether a foot is in contact. By incorporating the contact sensors with the finite state machine, we can handle abrupt contact from the environment and generate a new set of contact sequences for the optimal control problem if the previous contact sequences are incorrect.

However, one demerit of solely relying on contact sensors is that the nature of the hybrid system repeatedly deteriorates the contact sensors by repeated collision with the ground. This leads to worse contact sensory readings over time, even showing nonzero values when a leg is not in contact, as shown in Figure 4.5. Note, the difference between the mentioned hardware contact forces and the simulation contact forces

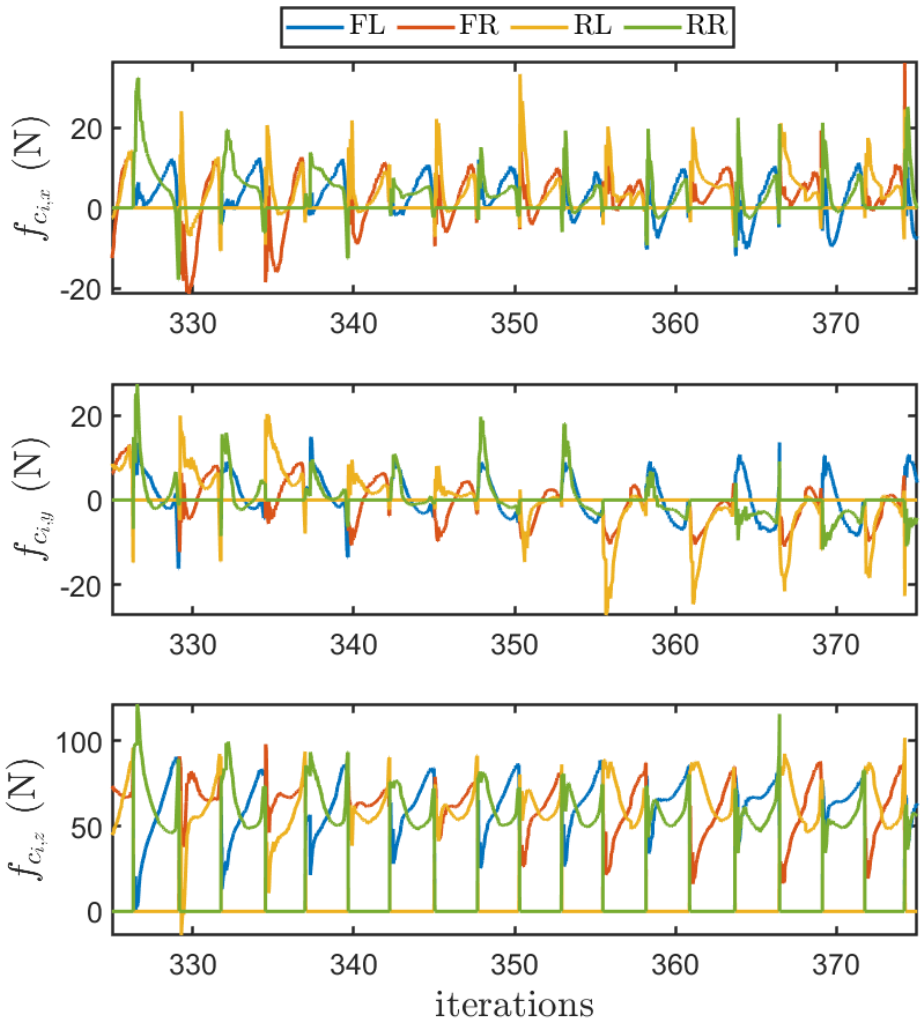

Figure 4.4: (FL: front left, FR: front right, RL: rear left, RR: rear right) Simulation GRFs produced by the feet. Note the y-direction reaction force (F_y) changes the direction track density-based plan.

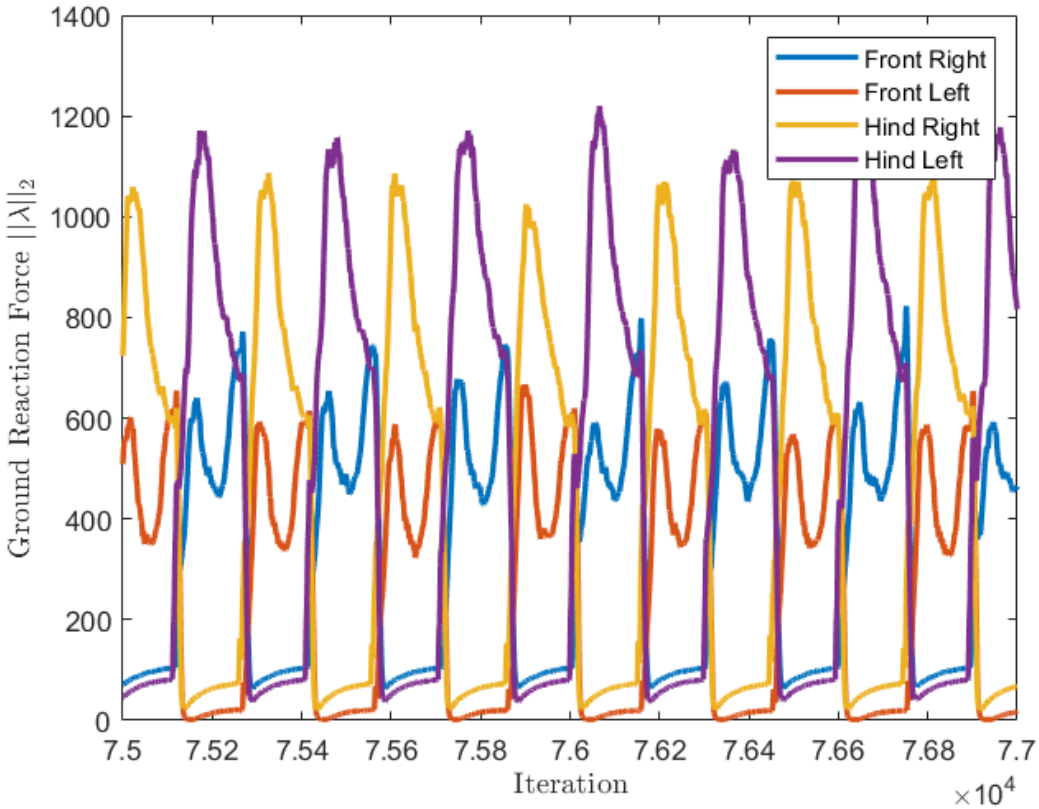

Figure 4.5: Contact readings from Unitree Go1 hardware. Display of stance and swing of Unitree Go1 with even swing phase showing nonzero contacts.

in Figure 4.4. The simulation contact forces show zero reaction force during the swing phase and nonzero during the stance. For hardware, the contact force shown shows nonzero but relatively small force sensed when a leg is in swing and a large reaction force when in stance. As we typically denote when a foot is in contact with a uniform contact threshold, a uniform contact threshold may lead to a perceived notion of a leg always being in stance, depending on the conditions of the contact sensor. Therefore, we instead denote a unique contact threshold for each leg. The effects of the change in contact threshold can be seen in Figure 4.6, where we give the quadruped a command to traverse 0.1m. Figure 4.5a shows when the contact threshold was set uniformly, showing poor results for traversing a distance of 0.1m. Likewise, Figure 4.5 shows when the contact threshold was set appropriately for each individual leg, giving improved performance.

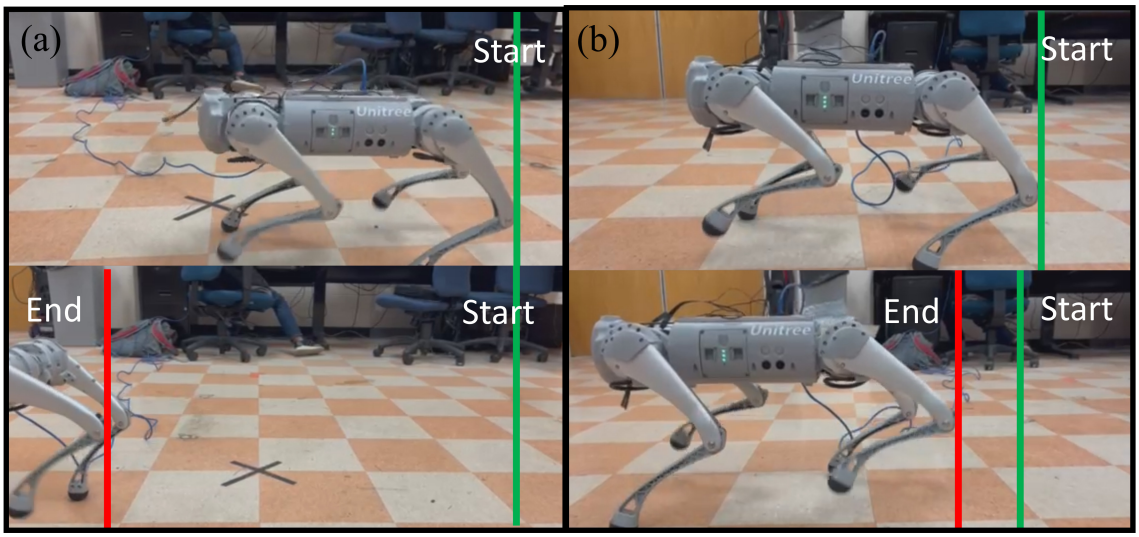

Figure 4.6: Contact threshold performance difference for a given command to travel 0.1m. Performance for a) Contact threshold uniformly set to 30 Newtons b) Appropriately set for each leg

4.6.2 State Estimation

As the robot relies on an inertial measurement unit (IMU) to measure acceleration and velocity states, there will naturally be an accumulation of drift in the position states of the robot. Since our feedback density planner relies on accurate state information – in the context of navigation, the position state is crucial – it is necessary to rely on accurate position information to ensure safety. Therefore, we rely on two major setups: a motion capture system that gives accurate localization data and a redundant mesh configuration system that is robust to the masking of LED lights for the mocap system.

First, we utilize a Phasespace motion capture system that is physically set up in the lab. The camera

capture system is a 15x10 meter configuration frame for a 24-camera setup as shown in Figure 4.7.

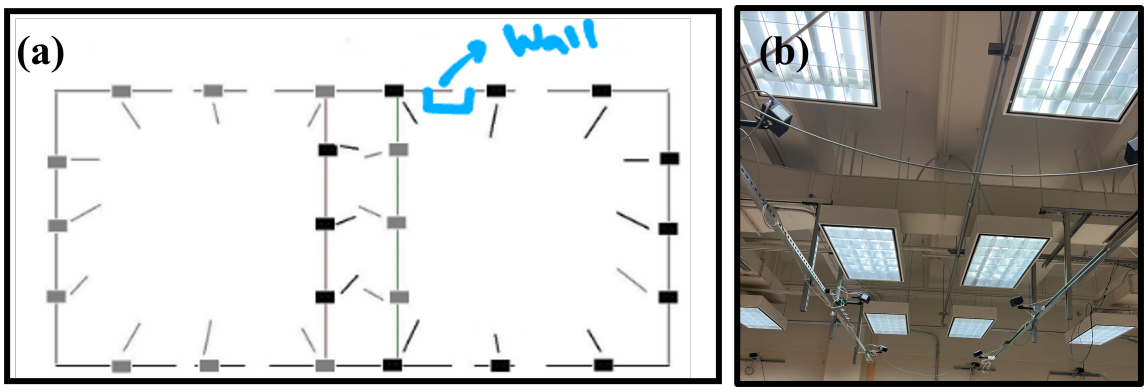

Figure 4.7: Lab Motion Capture Configuration. It is a 15x20 meter configuration with 24 camera setups. There exists overlap to get better localization between the intersection of camera setups. a) Shows the two overlapping set of camera configuration and b) shows the set of overlapping camera configurations in the lab.

The refresh rate of the motion capture system operates at 60 Hz, giving positional information with respect to the motion capture frame and information about the condition of the data (i.e. if data is not reliable due to objects covering the line of sight of the camera to the LED lights, then that particular data is stated to be unreliable). These data are sent from the microdrivers and received on the Phasespace server.

Next, we consider the redundant mesh group algorithm with a visual representation of this algorithm shown in Figure 4.8. The redundant mesh group algorithm relies on the fact that a singular LED configuration, which we denote as a mesh, captures the centroidal states of the meshes as follows:

$$\left[\mathbf{x}_{mesh,i}, \upsilon_{mesh,j,k}\right] = S(\mathbf{X}_{j,k}, \upsilon_{j,k}), \tag{4.11}$$

where $\mathbf{X}_{j,k} \in \mathbb{R}^{3\times n_L}$ are the position information from LED light j to LED light k, $\upsilon_{j,k} \in \{0,1\}$ are the corresponding condition of the LED lights, $\mathbf{x}_{mesh,i} \in \mathbb{R}^6$ is the ith centroidal or mesh state, $\upsilon_{mesh,i} \in \{0,1\}$ is the ith condition on the corresponding mesh (0 denoting unregistered and 1 denoting registered), and S is the mapping of each LED positional information to a centroidal state. We note that in this framework, if the mesh configuration consists of only a line as shown in Figure 4.8a, then the orientation states in $\mathbf{x}_{mesh,i}$ are not computed and only positional states are computed.

Then, as each mesh may not be registered, we utilize a redundant mesh configuration (or mesh group) to compute the centroidal state of the quadruped given one mesh is registered. If multiple are registered, a combined quadruped centroidal state is computed for the group of meshes. For Figure 4.8a, the

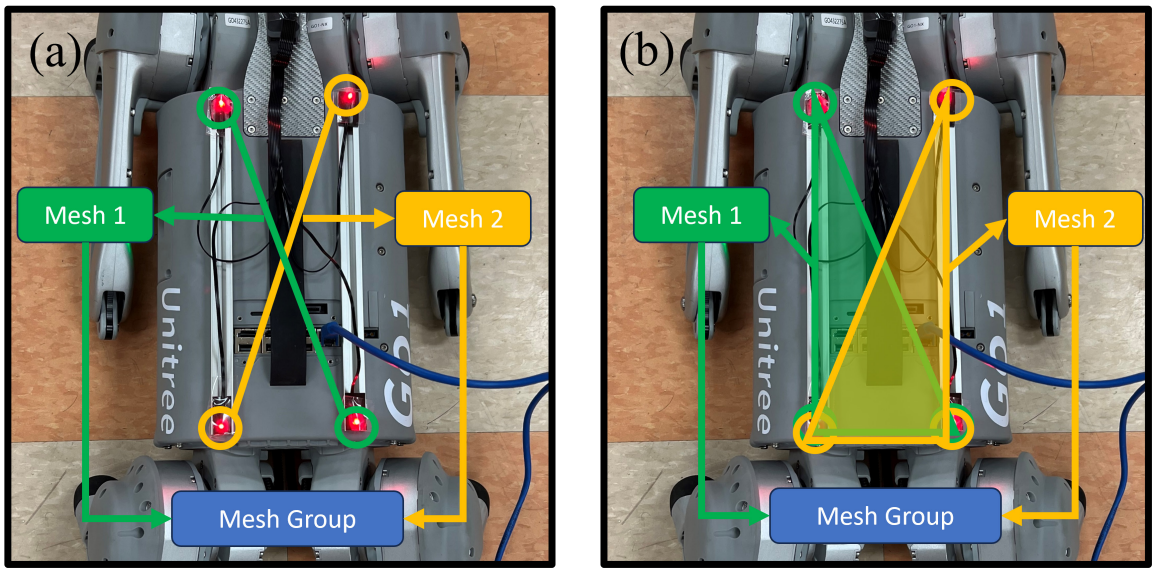

Figure 4.8: Visualization of the mesh configuration system. The general system is such that each LED light can be within a mesh, with each mesh within one mesh group. This redundant structure allows robustness to masking of a mesh. Here, a) the mesh is in the form of a line w/ the mesh group being a collection of lines, whereas, b) the mesh is in the form of a triangle w/ the mesh group being collection of triangles

computation of the mesh group state from overlapping line meshes would be

$$\mathbf{x}_{mesh} = \sum_{i=1}^{n} \mathbf{x}_{mesh,i}/n$$

whereas Figure 4.8b would require more than simply computing a uniformly weighted average. Then, given a robust mesh state, which is a state in the inertial aligned motion capture frame, a transformation matrix is utilized to be mapped to the odometry frame of the robot. Thus, using this method of computing the quadrupedal state, the redundant mesh group algorithm allows us to handle the partial obscuring of LED lights due to unforeseen objects.

4.6.3 Results

We show results on hardware where the feedback density planner and corresponding NMPC were computed on an AMD Ryzen 5 4600H computer offboard. The following optimized torque command is then sent to the low-level control module in the Unitree Go1 through a UDP connection, where the IMU states and motor states are sent back to the offboard computer for computation of the following torque.

The first results do not use a motion capture system for localization and, therefore, have large esti-

Figure 4.9: Feedback Density Planner on Go1 Hardware without a motion capture system. Extra enlargement of the obstacle set was done to account for state estimation drifts. The following figure shows the Unitree Go1 at a) start, b) beginning to avoid $\mathbf{X}_u$, c) after avoiding $\mathbf{X}_u$, and d) goal

mation drifts. We enlarge the obstacle set to account for this uncertainty, where the original X_u is of radius 0.25m and the enlarged is 1m (these radii account for robot radius and obstacle radius). This is shown to achieve safe navigation as shown in Figure 4.9. A video of the implementation is available on YouTube[3].

Correspondingly, we show another experimental setup with the motion capture system. Given an $\mathbf{X}_0 = (0,0), \mathbf{X}_T = (4,0)$ with $\mathbf{X}_u$ origin to be (1.5, 0) and radius to be 0.5m, we note the improved localization with the motion capture system. Figure 4.10 shows that the estimation of the position states is highly accurate, so much so that obstacle avoidance with convergence is guaranteed without any unnecessary enlargement of the obstacle radius.

4.7 Conclusions

In this work, we develop a safe motion planning architecture for quadruped locomotion. We use density functions to design a safe reference trajectory for the robot. The trajectories are obtained as a positive gradient of the proposed density function, giving a closed-loop feedback form. The proposed algorithm is

[3]https://www.youtube.com/watch?v=gJH6RTcHrfg

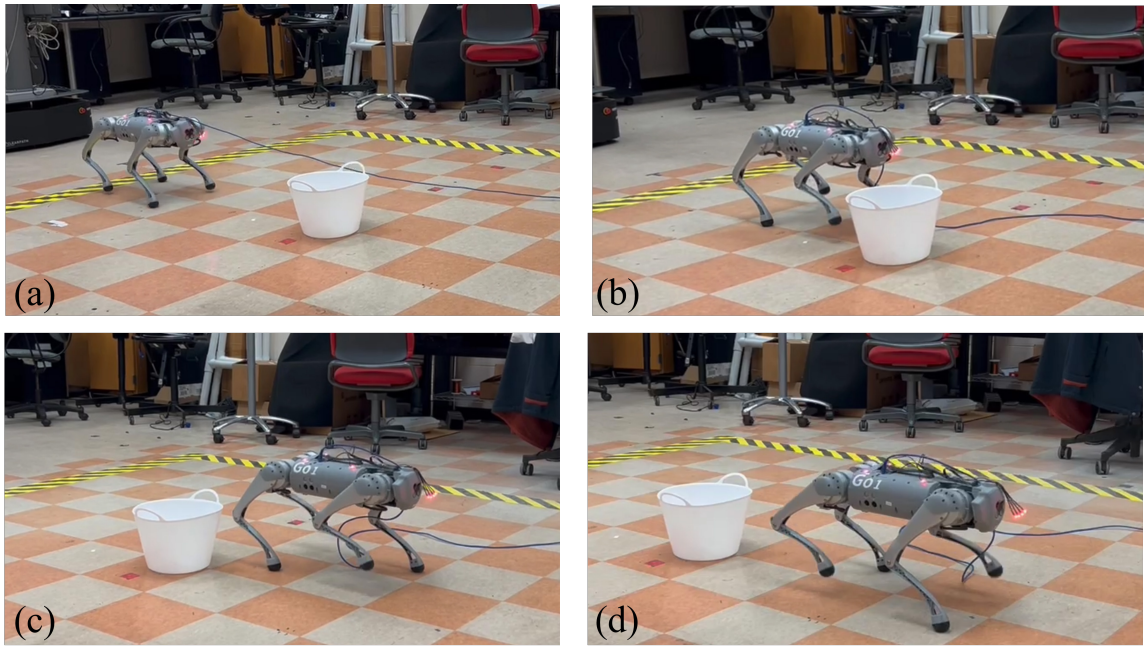

Figure 4.10: Feedback Density Planner on Go1 Hardware with a motion capture system. No extra enlargment of the obstacle set was needed. The following figure shows the Unitree Go1 at a) start, b) beginning to avoid $\mathbf{X}_u$, c) after avoiding $\mathbf{X}_u$, and d) goal

integrated with a nonlinear MPC and low-level controller from the OCS2 framework. Simulation results show that the robot is able to track safe reference trajectories provided by the density-based motion planning framework.

One of the current limitations is that the density formulation is limited to a binary representation of the environment, safe and unsafe. Therefore, there is no distinction for the degree of traversability, limiting the extension of this work to unstructured environments. Additionally, the current formulation assumes a holonomic system, hence limiting the class of systems for this feedback controller. Lastly, the quality of trajectories can be improved; therefore, a proper filter to smoothen state and control trajectory without introducing phase delays will significantly increase the planner's performance. Future works look to integrate a trajectory optimizer to smoothen the trajectory and extend this framework to non-holonomic systems and off-road navigation.

Chapter 5

Conclusions and Outlook

5.1 Summary

The contribution of this work is twofold: the formulation of an analytically constructed density feedback planner for the almost everywhere navigation problem and the use of the density feedback planner applied on quadruped due to their diverse capabilities in locomotion to integrate the density-based plan into the legged locomotion problem.

For the first contribution, the novelty leverages the interpretation of navigation through density from linear operator theory to analytically construct a density function. Other works have approached the navigation problem through the mathematical optimization lens, utilizing the convex form of the almost everywhere convergence criterion to solve for a density function and feedback controller. However, regarding navigation under safety constraints, the physical interpretation of safety through density is lost through the optimization. More related works have tried leveraging the physical interpretation of density with linear operator theory to convexly optimize for the density function. However, the challenge to this method is the curse of dimensionality, where the infinitesimal linear operator is approximated, and hence, the computation of a density function is intractable. Contrary to utilizing optimization-based method, this work utilizes the occupation-based interpretation to directly construct an analytical density function for the safe navigation problem. By utilizing occupation interpretation from linear operator theory, the density constructed is able to incorporate safety constraints implicitly. Moreover, this work proves that the analytically constructed density function and the corresponding density-based controller satisfy the almost everywhere convergence criterion.

The density-based feedback controller is then integrated into a density-based navigation architecture

applied to the quadruped. Due to the hierarchical nature of the legged locomotion problem, it is noted that simplified models are used to achieve legged locomotion. Furthermore, it has been shown that these simplified models are sufficient enough to give reliable trajectories for the locomotion problem. Therefore, inspired by other works that use simple integrator dynamics as the reference trajectory, we integrate a feedback planner applied on a physical quadruped and show simulation and hardware results.

5.2 Future Research Directions

We see large potential in leveraging the intuitive occupation interpretation through density to construct an analytical feedback controller for almost everywhere convergence under safety constraints.

However, current works have only proven this in the static environment case. Future works should focus on utilizing the controller under more realistic applications where the environment contains static and dynamic safety constraints. As the computation is quite simple compared to other related works [34], a promising lead is the recomputation of density function under changing environments. Progressing these works will aid in the theoretical understanding and showcase the practical limits of density for safe navigation in more realistic environmental conditions. Furthermore, as the current work only utilizes integrator dynamics, extending the work to ensure safety for non-trivial dynamics (e.g. nonholonomic systems) would be beneficial.

Moreover, we note that the integration of simplified dynamics for the motion planning of the legged locomotion problem, do not fully utilize the dynamics of the quadruped. Therefore, the current work does not leverage the unstructured terrain capabilities of quadruped robots. By extending the analytical controller to a model more appropriate for quadrupeds, we can leverage more of the legged locomotion capabilities for navigation under safety constraints. Future works would include extensions of the density-based navigation to a 3-dimensional constrained environment and utilizing occupation information to navigate in unstructured terrains.

Appendices

Appendix A Proof of Almost Everywhere Navigation using Analytically Constructed Density Functions

The proof of Theorem 1 relies on the following Lemma.

Lemma 1 *Consider the navigation density function as given in equation (3.4), then under Assumption 1, we have*

$$\nabla \cdot (\mathbf{k}(\mathbf{x})\rho(\mathbf{x})) \geq 0, \quad a.e. \ \mathbf{x} \in \mathbf{X}, \tag{1}$$

$$\nabla \cdot (\mathbf{k}(\mathbf{x})\rho(\mathbf{x})) \geq \xi > 0 \ \text{ for } \ \mathbf{x} \in \mathbf{X}_0, \tag{2}$$

where $\mathbf{k}(\mathbf{x}) = \nabla\rho(\mathbf{x})$ *is the feedback control input (3.11).*

Proof: We have

$$\nabla \cdot (\mathbf{k}(\mathbf{x})\rho(\mathbf{x})) = \rho(\mathbf{x})\nabla \cdot \mathbf{k}(\mathbf{x}) + \frac{\partial \rho}{\partial \mathbf{x}} \frac{\partial \rho}{\partial \mathbf{x}}^{\top}. \tag{3}$$

The proof will follow if we can show that $\nabla \cdot \mathbf{k}(\mathbf{x}) \geq 0$. Since $\rho(\mathbf{x}) > 0$ and $\frac{\partial \rho}{\partial \mathbf{x}} \frac{\partial \rho}{\partial \mathbf{x}}^{\top} \geq 0$, we have

$$\nabla \cdot \mathbf{k}(\mathbf{x}) = \sum_{j=1}^{n} \frac{\partial^2 \rho}{\partial x_j^2}. \tag{4}$$

Letting $\Psi(\mathbf{x}) = \prod_{k=1}^{L} \Psi_k(\mathbf{x})$, we obtain

$$\begin{aligned}
\frac{\partial^2 \rho}{\partial x_j^2} &= \frac{\partial}{\partial x_j}\left(-\frac{\alpha}{V^{\alpha+1}}\frac{\partial V}{\partial x_j}\Psi(\mathbf{x}) + \frac{1}{V^{\alpha}}\frac{\partial \Psi}{\partial x_j}\right) \\
&= \frac{\alpha}{V^{\alpha}}\left(\frac{(\alpha+1)}{V^2}\left|\frac{\partial V}{\partial x_j}\right|^2 - \frac{1}{V}\frac{\partial^2 V}{\partial x_j^2}\right)\Psi(\mathbf{x}) \\
&+ \frac{\alpha}{V^{\alpha}}\left(-\frac{2}{V}\frac{\partial V}{\partial x_j}\frac{\partial \Psi}{\partial x_j} + \frac{1}{\alpha}\frac{\partial \Psi^2}{\partial x_j^2}\right).
\end{aligned} \tag{5}$$

It is important to note that the last two terms in the above expression are non-zero only in the transition region $\mathbf{X}_{s_k}$. Outside this transition region $\frac{\partial \Psi}{\partial x_j} = 0$ and $\frac{\partial^2 \Psi}{\partial x_j^2} = 0$. To show that the above quantity is positive outside the transition region in $\mathbf{X}_1$, we use the bounds from Assumption 1. We have

$$\frac{(\alpha+1)}{V}\left|\frac{\partial V}{\partial x_j}\right|^2 - \frac{\partial^2 V}{\partial x_j^2} \geq (\alpha+1)\bar{d}_V^{-1}\underline{d}_{V_x}^2 - \bar{d}_{V_x^2}$$

Thus by choosing α sufficiently large the above quantity can be made positive. We next show that equation (5) is non-negative in the transition region. For this, we make use of the following facts. First, $\Psi_k(\mathbf{x}) \geq \theta > 0$ for $k = 1, \ldots, L$ and hence $\Psi(\mathbf{x})$ is bounded away from zero. Second, from the construction of $\Psi(\mathbf{x})$ and $V(\mathbf{x})$ functions there exists uniform bounds on $\frac{\partial \Psi_k}{\partial x_j}$, $\frac{\partial^2 \Psi_k}{\partial x_j^2}$, $\frac{\partial V}{\partial x_j}$, and $\frac{\partial^2 V}{\partial x_j^2}$. Third, using Assumption 1, we know that the distance between the unsafe set and the target set is bounded away from zero by a positive constant ζ and hence $\left|\frac{\partial V}{\partial x_j}\right|^2$ is bounded away from zero. Hence, the following bounds can be obtained for the $\frac{\partial^2 \rho}{\partial x_j^2}$ term

$$\frac{(\alpha+1)}{V^2}\left|\frac{\partial V}{\partial x_j}\right|^2 \Psi(\mathbf{x}) \geq \left((\alpha+1)\bar{c}_V^{-2}\underline{c}_{V_x}^2\right)\theta, \quad \frac{1}{\alpha}\frac{\partial^2 \Psi}{\partial x_j^2} \geq -\frac{\bar{c}_{\Psi_x^2}}{\alpha},$$
$$-\frac{1}{V}\frac{\partial^2 V}{\partial x_j^2}\Psi(\mathbf{x}) \geq -\underline{c}_V^{-1}\bar{c}_{V_{x^2}}\theta, \quad -\frac{2}{V}\frac{\partial V}{\partial x_j}\frac{\partial \Psi}{\partial x_j} \geq -2\underline{c}_V^{-1}\bar{c}_{V_x}\bar{c}_{\Psi_x}.$$

Therefore, we have following lower bound for $\frac{\partial^2 \rho}{\partial x_j^2}$

$$\begin{aligned}\frac{\partial^2 \rho}{\partial x_j^2} \geq & \frac{\alpha}{V^\alpha}\left(\left((\alpha+1)\bar{c}_V^{-2}\underline{c}_{V_x}^2 - \underline{c}_V^{-1}\bar{c}_{V_{x^2}}\right)\theta\right) \\ & + \frac{\alpha}{V^\alpha}\left(-2\underline{c}_V^{-1}\bar{c}_{V_x}\bar{c}_{\Psi_x} - \frac{\bar{c}_{\Psi_x^2}}{\alpha}\right).\end{aligned}$$

Hence, by choosing α sufficiently large, of order $\frac{1}{\theta}$, we can make the term inside the bracket positive.

To show that equation (2) is satisfied, we again make use of Assumption 1 and the fact that $\Psi(\mathbf{x}) = 1$, $\frac{\partial \Psi}{\partial \mathbf{x}_j} = 0$, and $\frac{\partial^2 \Psi}{\partial x_j^2} = 0$ for $\mathbf{x} \in \mathbf{X}_0$ and $j = 1, \ldots, n$. Further, $\left|\frac{\partial V}{\partial x_j}\right|^2$ is bounded away from zero. Hence for, $\mathbf{x} \in \mathbf{X}_0$ and for some $\xi > 0$, we obtain

$$\begin{aligned}\nabla \cdot (\mathbf{k}(\mathbf{x})\rho(\mathbf{x})) = & \frac{\partial \rho}{\partial \mathbf{x}}\frac{\partial \rho}{\partial \mathbf{x}}^\top + \frac{\alpha(\alpha+1)}{V^{\alpha+2}}\left|\frac{\partial V}{\partial x_j}\right|^2 \\ & - \frac{1}{V^{\alpha+1}}\frac{\partial^2 V}{\partial x_j^2} \geq \xi > 0.\end{aligned}$$

Proof of Theorem 1: Using the results of Lemma 1, we know that the density ρ satisfies

$$\nabla \cdot (\mathbf{k}(\mathbf{x})\rho(\mathbf{x})) = g(\mathbf{x}) \tag{6}$$

for some $g(\mathbf{x}) \geq 0$ such that $g(\mathbf{x}) \geq \xi > 0$ for $\mathbf{x} \in \mathbf{X}_0$.

Since $\rho(\mathbf{x})$ satisfies the linear partial differential equation (6), it follows using the method of characteristics that the solution $\rho(x)$ can be written in terms of the solution $\mathbf{s}_t(\mathbf{x})$, of the system $\dot{\mathbf{x}} = \mathbf{k}(\mathbf{x})$ as

follows [39]

$$\rho(\mathbf{x}) = \frac{\Psi(\mathbf{x})}{V^{\alpha}(\mathbf{x})} = \int_0^{\infty} g(\mathbf{s}_{-t}(\mathbf{x})) \left| \frac{\partial \mathbf{s}_{-t}(\mathbf{x})}{\partial \mathbf{x}} \right| dt, \tag{7}$$

where $|\cdot|$ is the determinant. The proof follows by substituting the integral formula for $\rho(\mathbf{x})$ from (7) in (6) and using the fact that

$$\lim_{t\to\infty} g(\mathbf{s}_{-t}(\mathbf{x})) \left| \frac{\partial \mathbf{s}_{-t}(\mathbf{x})}{\partial \mathbf{x}} \right| = 0. \tag{8}$$

The limit in (8) goes to zero as $\rho(\mathbf{x})$ is bounded for all $\mathbf{x} \in \mathbf{X}_1$ and using Barbalat's Lemma. The integrant in (7) defines a semi-group of linear Perron-Frobenius (P-F) operator, $\mathbb{P}_t$, acting on function $g(\mathbf{x})$ and can be written compactly as

$$[\mathbb{P}_t g](\mathbf{x}) = g(\mathbf{s}_{-t}(\mathbf{x})) \left| \frac{\partial \mathbf{s}_{-t}(\mathbf{x})}{\partial \mathbf{x}} \right|. \tag{9}$$

Using (9), (7) can be written as

$$\rho(\mathbf{x}) = \int_0^{\infty} [\mathbb{P}_t g](\mathbf{x}) dt. \tag{10}$$

Furthermore, (8) can be written as

$$\lim_{t\to\infty} [\mathbb{P}_t g](\mathbf{x}) = 0 \implies \lim_{t\to\infty} [\mathbb{P}_t \mathbb{1}_{\mathbf{X}_0}](\mathbf{x}) = 0,$$

where $\mathbb{1}_{\mathbf{X}_0}$ is the indicator function for set $\mathbf{X}_0$. This implication follows because $g(\mathbf{x}) \geq \xi > 0$ for all $\mathbf{x} \in \mathbf{X}_0$ and from dominated convergence theorem. For any set $A \subseteq \mathbf{X}_1$, we have

$$\begin{aligned} \int_A [\mathbb{P}_t \mathbb{1}_{\mathbf{X}_0}](\mathbf{x}) d\mathbf{x} &= \int_{\mathbf{X}_1} [\mathbb{P}_t \mathbb{1}_{\mathbf{X}_0}](\mathbf{x}) \mathbb{1}_A(\mathbf{x}) d\mathbf{x} \\ &= \int_{\mathbf{X}_1} \mathbb{1}_{\mathbf{X}_0}(\mathbf{x}) \mathbb{1}_A(\mathbf{s}_t(\mathbf{x})) dx. \end{aligned} \tag{11}$$

The above follows by using the definition of $\mathbb{P}_t$ in (9) and change of variables in the integration, i.e., $\mathbf{y} = \mathbf{s}_{-t}(\mathbf{x})$

and $d\mathbf{y} = |\frac{\partial \mathbf{s}_{-t}(\mathbf{x})}{\partial \mathbf{x}}| d\mathbf{x}$ and after relabeling. Note that the right-hand side of (11) is nothing but

$$\int_A [\mathbb{P}_t \mathbb{1}_{\mathbf{X}_0}](\mathbf{x}) d\mathbf{x} = m\{\mathbf{x} \in \mathbf{X}_0 : \mathbf{s}_t(\mathbf{x}) \in A\}.$$

From Lebesgue dominated convergence theorem

$$0 = \int_A \lim_{t\to\infty} [\mathbb{P}_t \mathbb{1}_{\mathbf{X}_0}](\mathbf{x}) d\mathbf{x}$$

$$= \int_{\mathbf{X}_1} \mathbb{1}_{\mathbf{X}_0}(\mathbf{x}) \lim_{t\to\infty} \mathbb{1}_A(\mathbf{s}_t(\mathbf{x})) d\mathbf{x} = m\{\mathbf{x} \in \mathbf{X}_0 : \mathbf{s}_t(\mathbf{x}) \in A\}.$$

Since the above is true for any measurable and positive Lebesgue measure set $A \subseteq \mathbf{X}_1 := \mathbf{X} \setminus \mathscr{B}_\delta$ for arbitrary small δ, we obtain

$$m\{\mathbf{x} \in \mathbf{X}_0 : \lim_{t\to\infty} \mathbf{s}_t(\mathbf{x}) \neq 0\} = 0. \tag{12}$$

We next show that the unsafe set $\mathbf{X}_{u_k}$ will be avoided by trajectories $\mathbf{s}_t(\mathbf{x})$ starting from almost all w.r.t. Lebesgue measure initial condition $\mathbf{x} \in \mathbf{X}_0$. We have for $\mathbf{x} \in \mathbf{X}_{u_k}$

$$\rho(\mathbf{x}) = \frac{\Psi_k(\mathbf{x})}{V^\alpha} = \frac{\theta}{V^\alpha}. \tag{13}$$

Following Assumption 1 (equation (3.9)), we have

$$\rho(\mathbf{x}) = \frac{\theta}{V^\alpha} \leq \frac{\theta}{V^k_{min}}. \tag{14}$$

Using the above bound on $\rho(\mathbf{x})$, we obtain

$$\mathbf{G} := \int_{\mathbf{X}_{u_k}} \int_0^\infty [\mathbb{P}_t \mathbb{1}_{\mathbf{X}_0}](\mathbf{x}) dt d\mathbf{x} = \int_{\mathbf{X}_{u_k}} \rho(\mathbf{x}) d\mathbf{x} \leq \frac{\theta}{V^k_{min}} m(\mathbf{X}_{u_k}),$$

where $m(\cdot)$ is the Lebesgue measure. Utilizing that $d\mathbf{y} = |\frac{\partial \mathbf{s}_{-t}(\mathbf{x})}{\partial \mathbf{x}}| d\mathbf{x}$, which is described through the definition of $\mathbb{P}_t$ and performing a change of variable $\mathbf{y} = \mathbf{s}_{-t}(\mathbf{x})$, we can use the bounds on $\rho(\mathbf{x})$ in (14) for $\mathbf{x} \in \mathbf{X}_{u_k}$ to obtain

$$\mathbf{G} = \int_{\mathbf{X}_1} \mathbb{1}_{\mathbf{X}_0}(\mathbf{y}) \int_0^\infty \mathbb{1}_{\mathbf{X}_{u_k}}(\mathbf{s}_t(\mathbf{y})) dt d\mathbf{y} \leq \frac{\theta}{V^k_{min}} m(\mathbf{X}_{u_k}).$$

The time integral on the left-hand side is the time spent by system trajectories starting from the initial set $\mathbf{X}_0$ in the unsafe set $\mathbf{X}_{u_k}$. Let this time be denoted by $T(\mathbf{y})$. Hence, we obtain

$$\int_{\mathbf{X}_1} T(\mathbf{y}) \mathbb{1}_{\mathbf{X}_0}(\mathbf{y}) d\mathbf{y} \leq \frac{\theta}{V_{min}^k} m(\mathbf{X}_{u_k}).$$

Following Assumption 1 (equation (3.10)), we have

$$\theta \leq \varepsilon \frac{V_{min}^k}{m(\mathbf{X}_{u_k})} \implies \int_{\mathbf{X}_0} T(\mathbf{y}) d\mathbf{y} \leq \varepsilon,$$

for any given $\varepsilon > 0$.

Choose some $\eta < 1$, then using Chebyshev's inequality and the fact that $\mathbf{X}_0 \subset \mathbf{X}_1$, we have

$$m\{\mathbf{x} \in \mathbf{X}_0 : T(\mathbf{y}) \geq \varepsilon^{\eta}\} \leq \varepsilon^{-\eta} \int_{\mathbf{X}_0} T(\mathbf{y}) d\mathbf{y} \leq \varepsilon^{-\eta+1}.$$

Since the above is true for arbitrary small $\varepsilon > 0$, we have

$$m\{\mathbf{x} \in \mathbf{X}_0 : T(\mathbf{y}) = \int_0^{\infty} \mathbb{1}_{\mathbf{X}_{u_k}}(\mathbf{s}_t(\mathbf{x})) dt > 0\} = 0. \tag{15}$$

Now we make use of the continuity property of the flow $\mathbf{s}_t(\mathbf{x})$ w.r.t. time to show that $\mathbb{1}_{\mathbf{X}_{u_k}}(\mathbf{s}_t(\mathbf{x})) = 0$ for all $t \geq 0$. Assume not, then there exists γ and $\bar{t}$ such that $\mathbb{1}_{\mathbf{X}_{u_k}}(\mathbf{s}_{\bar{t}}(\mathbf{x})) \geq \gamma > 0$. Then from the continuity of solution $\mathbf{s}_t(\mathbf{x})$ w.r.t. time, we know that there exists $\Delta > 0$ such that $\mathbb{1}_{\mathbf{X}_{u_k}}(\mathbf{s}_t(\mathbf{x})) > 0$ for $t \in [\bar{t}, \bar{t} + \Delta]$. This violates (15).

Bibliography

[1] Ali Agha, Kyohei Otsu, Benjamin Morrell, David D Fan, Rohan Thakker, Angel Santamaria-Navarro, Sung-Kyun Kim, Amanda Bouman, Xianmei Lei, Jeffrey Edlund, et al. Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge. *arXiv preprint arXiv:2103.11470*, 2021.

[2] Nancy M Amato and Yan Wu. A randomized roadmap method for path and manipulation planning. In *Proceedings of IEEE international conference on robotics and automation*, volume 1, pages 113–120. IEEE, 1996.

[3] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE, 2019.

[4] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[5] Gerardo Bledt, Matthew J Powell, Benjamin Katz, Jared Di Carlo, Patrick M Wensing, and Sangbae Kim. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2245–2252. IEEE, 2018.

[6] Jared Di Carlo, Patrick M Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1–9. IEEE, 2018.

[7] Moritz Diehl, Hans Georg Bock, and Johannes P Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on control and optimization*, 43(5):1714–1736, 2005.

[8] Dimos V Dimarogonas and Kostas J Kyriakopoulos. An application of rantzer's dual lyapunov theorem to decentralized formation stabilization. In *2007 European Control Conference (ECC)*, pages 882–888. IEEE, 2007.

[9] David D Fan, Kyohei Otsu, Yuki Kubo, Anushri Dixit, Joel Burdick, and Ali-Akbar Agha-Mohammadi. Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation. *arXiv preprint arXiv:2103.02828*, 2021.

[10] Gilbert Feng, Hongbo Zhang, Zhongyu Li, Xue Bin Peng, Bhuvan Basireddy, Linzhu Yue, Zhitao Song, Lizhi Yang, Yunhui Liu, Koushil Sreenath, et al. Genloco: Generalized locomotion controllers for quadrupedal robots. *arXiv preprint arXiv:2209.05309*, 2022.

[11] Ioannis F Filippidis and Kostas J Kyriakopoulos. Navigation functions for everywhere partially sufficiently curved worlds. In *2012 IEEE International Conference on Robotics and Automation*, pages 2115–2120. IEEE, 2012.

[12] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. Batch informed trees (bit): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 3067–3074. IEEE, 2015.

[13] Ruben Grandia, Andrew J Taylor, Aaron D Ames, and Marco Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8352–8358. IEEE, 2021.

[14] Alain Haït, Thierry Simeon, and Michel Taïx. Algorithms for rough terrain trajectory planning. *Advanced Robotics*, 16(8):673–699, 2002.

[15] Qiang Huang, Kazuhito Yokoi, Shuuji Kajita, Kenji Kaneko, Hirohiko Arai, Noriho Koyachi, and Kazuo Tanie. Planning walking patterns for a biped robot. *IEEE Transactions on robotics and automation*, 17(3):280–289, 2001.

[16] Nicolas Hudson, Fletcher Talbot, Mark Cox, Jason Williams, Thomas Hines, Alex Pitt, Brett Wood, Dennis Frousheger, Katrina Lo Surdo, Thomas Molnar, et al. Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61's approach to the darpa subterranean challenge. *arXiv preprint arXiv:2104.09053*, 2021.

[17] Lucas Janson, Edward Schmerling, Ashley Clark, and Marco Pavone. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *The International journal of robotics research*, 34(7):883–921, 2015.

[18] Fabian Jenelten, Ruben Grandia, Farbod Farshidian, and Marco Hutter. Tamols: Terrain-aware motion optimization for legged systems. *IEEE Transactions on Robotics*, 38(6):3395–3413, 2022.

[19] Yandong Ji, Zhongyu Li, Yinan Sun, Xue Bin Peng, Sergey Levine, Glen Berseth, and Koushil Sreenath. Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot. *arXiv preprint arXiv:2208.01160*, 2022.

[20] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

[21] Oussama Khatib. The potential field approach and operational space formulation in robot control. In *Adaptive and learning systems*, pages 367–377. Springer, 1986.

[22] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer, 1986.

[23] Donghyun Kim, Daniel Carballo, Jared Di Carlo, Benjamin Katz, Gerardo Bledt, Bryan Lim, and Sangbae Kim. Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2464–2470. IEEE, 2020.

[24] D. E. Koditschek. The Control of Natural Motion in Mechanical Systems. *Journal of Dynamic Systems, Measurement, and Control*, 113(4):547–551, 12 1991.

[25] Bruce Krogh. A generalized potential field approach to obstacle avoidance control. In *Proc. SME Conf. on Robotics Research: The Next Five Years and Beyond, Bethlehem, PA, 1984*, pages 11–22, 1984.

[26] Andrzej Lasota and Michael C Mackey. *Chaos, fractals, and noise: stochastic aspects of dynamics*, volume 97. Springer Science & Business Media, 1998.

[27] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[28] Steven M LaValle and James J Kuffner. Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan. *Algorithmic and Computational Robotics*, pages 303–307, 2001.

[29] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[30] Savvas G Loizou. Closed form navigation functions based on harmonic potentials. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6361–6366. IEEE, 2011.

[31] Savvas G Loizou and Ali Jadbabaie. Density functions for navigation-function-based systems. *IEEE Transactions on Automatic Control*, 53(2):612–617, 2008.

[32] Tamas G Molnar, Ryan K Cosner, Andrew W Singletary, Wyatt Ubellacker, and Aaron D Ames. Model-free safety-critical control for robotic systems. *IEEE robotics and automation letters*, 7(2):944–951, 2021.

[33] Tamas G. Molnar, Ryan K. Cosner, Andrew W. Singletary, Wyatt Ubellacker, and Aaron D. Ames. Model-free safety-critical control for robotic systems. *IEEE Robotics and Automation Letters*, 7(2):944–951, 2022.

[34] Joseph Moyalan, Yongxin Chen, and Umesh Vaidya. Convex approach to data-driven off-road navigation via linear transfer operators. *IEEE Robotics and Automation Letters*, pages 1–8, 2023.

[35] Sriram Narayanan, Andrew Zheng, and Umesh Vaidya. Safe motion planning for quadruped robots using density functions. In *2023 Indian Control Conference (ICC), accepted*. IEEE, 2023.

[36] David E Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous robots*, 35:161–176, 2013.

[37] Panagiotis Papadakis. Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence*, 26(4):1373–1385, 2013.

[38] Steven C Peters and Karl Iagnemma. An analysis of rollover stability measurement for high-speed mobile robots. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3711–3716. IEEE, 2006.

[39] R. Rajaram, U. Vaidya, M. Fardad, and B. Ganapathysubramanian. Almost everywhere stability: Linear transfer operator approach. *Journal of Mathematical analysis and applications*, 368:144–156, 2010.

[40] Anders Rantzer. A dual to Lyapunov's stability theorem. *Systems & Control Letters*, 42(3):161–168, 2001.

[41] Anders Rantzer and Stephen Prajna. On analysis and synthesis of safe control laws. In *42nd Allerton Conference on Communication, Control, and Computing*, 2004.

[42] Elon Rimon. *Exact robot navigation using artificial potential functions*. Yale University, 1990.

[43] Tomáš Rouček, Martin Pecka, Petr Čížek, Tomáš Petříček, Jan Bayer, Vojtěch Šalanskỳ, Daniel Heřt, Matěj Petrlík, Tomáš Báča, Vojěch Spurnỳ, et al. Darpa subterranean challenge: Multi-robotic exploration of underground environments. In *Modelling and Simulation for Autonomous Systems: 6th International Conference, MESAS 2019, Palermo, Italy, October 29–31, 2019, Revised Selected Papers 6*, pages 274–290. Springer, 2020.

[44] 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.

[45] Jean-Pierre Sleiman, Farbod Farshidian, Maria Vittoria Minniti, and Marco Hutter. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6(3):4688–4695, 2021.

[46] Marco Tranzatto, Takahiro Miki, Mihir Dharmadhikari, Lukas Bernreiter, Mihir Kulkarni, Frank Mascarich, Olov Andersson, Shehryar Khattak, Marco Hutter, Roland Siegwart, et al. Cerberus in the darpa subterranean challenge. *Science Robotics*, 7(66):eabp9742, 2022.

[47] Loring W Tu. Manifolds. In *An Introduction to Manifolds*, pages 47–83. Springer, 2011.

[48] Umesh Vaidya. Optimal motion planning using navigation measure. *International Journal of Control*, 91(5):989–998, 2018.

[49] Umesh Vaidya and Prashant G Mehta. Lyapunov measure for almost everywhere stability. *IEEE Transactions on Automatic Control*, 53(1):307–323, 2008.

[50] Miomir Vukobratović and Branislav Borovac. Zero-moment point—thirty five years of its life. *International journal of humanoid robotics*, 1(01):157–173, 2004.

[51] Patrick M Wensing, Michael Posa, Yue Hu, Adrien Escande, Nicolas Mansard, and Andrea Del Prete. Optimization-based control for dynamic legged robots. *arXiv preprint arXiv:2211.11644*, 2022.

[52] Alexander W Winkler. *Optimization-based motion planning for legged robots*. PhD thesis, ETH Zurich, 2018.

[53] Hongzhe Yu, Joseph Moyalan, Umesh Vaidya, and Yongxin Chen. Data-driven optimal control of nonlinear dynamics under safety constraints. *IEEE Control Systems Letters*, 6:2240–2245, 2022.

[54] Andrew Zheng, Sriram S.K.S Narayanan, and Umesh Vaidya. Safe navigation using density functions. *IEEE Robotics and Automation Letters*, pages 1–8, 2023.