FOOTHOLD SELECTION FOR QUADRUPED ROBOTS BASED ON FEASIBLE GROUND REACTION FORCE SETS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FURKAN DAVULCU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

AUGUST 2022

Approval of the thesis:

FOOTHOLD SELECTION FOR QUADRUPED ROBOTS BASED ON FEASIBLE GROUND REACTION FORCE SETS

submitted by **FURKAN DAVULCU** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Mehmet Halit S. Oğuztüzün
Head of Department, **Computer Engineering** ______________

Prof. Dr. Uluç Saranlı
Supervisor, **Computer Engineering, METU** ______________

Examining Committee Members:

Assoc. Prof. Dr. Sinan Kalkan
Computer Engineering, METU ______________

Prof. Dr. Uluç Saranlı
Computer Engineering, METU ______________

Assist. Prof. Dr. İsmail Uyanık
Electrical and Electronics Engineering, Hacettepe University ______________

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Furkan Davulcu

Signature :

ABSTRACT

FOOTHOLD SELECTION FOR QUADRUPED ROBOTS BASED ON FEASIBLE GROUND REACTION FORCE SETS

Davulcu, Furkan
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Uluç Saranlı

August 2022, 51 pages

Legged robots mainly control their speed and posture only at the stance phase, when their feet touch the ground, by producing desired forces on the body using ground reaction forces. However, the selection of footholds affects the forces that can be put on the body during the stance phase. This thesis presents a new foothold selection framework that selects footholds considering the friction cone, motor limits, and leg geometry to help the stance controllers control the speed and posture of the robot more effectively. We evaluate our framework in the scenario of coming to a stop from a diverse set of initial conditions on a planar quadrupedal robot in simulation and compare it with the widely used Raibert's heuristic for foothold selection. Our experiments show that the planar quadruped robot converges to the desired state faster and spends less energy while the motors spend less time in their torque limits using our method. Furthermore, our method makes the quadrupeds more robust against unexpected disturbances by selecting footholds that give more flexibility to stance controllers in terms of force output capability.

Keywords: legged robots, foothold selection, planar quadrupedal robot

ÖZ

DÖRT BACAKLI ROBOTLAR İÇİN UYGULANABİLİR YER TEPKİ KUVVETİ KÜMELERİNE DAYALI ADIM KONUMU SEÇİMİ

Davulcu, Furkan
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Uluç Saranlı

Ağustos 2022 , 51 sayfa

Bacaklı robotlar temel olarak sadece duruş fazında, ayakları yere değdiği zaman, yer reaksiyon kuvvetlerini kullanıp vücutta istenilen kuvvetleri üreterek hızlarını ve duruşlarını kontrol ederler. Ancak adım konumlarının seçimi, duruş fazında vücuda uygulanabilecek kuvvetlerin neler olabileceğini etkiler. Bu tez, duruş denetleyicilerinin robotun hızını ve durumunu daha etkin bir şekilde kontrol etmesine yardımcı olmak için sürtünme konisi, motor limitleri ve bacak geometrisini dikkate alarak adım konumlarını seçen yeni bir adım seçim çerçevesi sunmaktadır. Çerçevemizi, simülasyonda düzlemsel dört ayaklı bir robot üzerinde çeşitli başlangıç koşullarından durma senaryosunda değerlendiriyoruz ve onu adım konumu seçimi için yaygın olarak kullanılan Raibert'ın bulgusal yöntemi ile karşılaştırıyoruz. Çeşitli başlangıç koşullarından durma senaryosunda, yöntemimizi kullanarak dört bacaklı robotların istenen duruma daha hızlı yakınsadığını ve daha az enerji harcadığını, motorların ise tork limitlerinde daha az zaman harcadığını gösteriyoruz. Ayrıca, yöntemimiz, kuvvet çıkışı kapasitesi açısından duruş denetleyicilerine daha fazla esneklik sağlayan adım konumları seçerek, dört bacaklı robotları beklenmeyen bozulmalara karşı daha sağlam hale ge-

tirmektedir.

Anahtar Kelimeler: bacaklı robotlar, adım konumu seçimi, düzlemsel dört bacaklı robot

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Dr. Uluç Saranlı, who introduced me to the world of legged robotics. He has guided and supported me since my undergraduate years and inspired me to become a robotics researcher. I am also grateful to Assoc. Prof. Dr. Mustafa Mert Ankaralı for his discussions and guidance throughout my thesis journey.

I want to thank my dear friends from ATLAS Laboratory: Emre Tanfener, Güner Dilşad Er, Sait Sovukluk, and Mert Kaan Yılmaz. They provided a great work environment with fruitful discussions and gave me feedback and support when I needed it the most. I especially want to thank Emre for our invaluable conversations throughout my thesis.

I want to express my deepest gratitude to my parents and my sister for their endless support and all they have done for me. This thesis would not have been possible without them.

Finally, I thank my beloved partner Sonat for her unconditional love and support over these challenging years.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
SLIP	Spring-Loaded Inverted Pendulum
FFS	Feasible Force Set
PQ	Planar Quadruped
CoM	Center of Mass
DoF	Degree of Freedom

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

While animals and humans can easily reach the vast majority of the land, wheeled and tracked robots require certain infrastructures such as paved roads or continuous surfaces. Legged robots are attractive alternatives to tracked and wheeled robots for applications in rough terrain and complex, cluttered environments. One of the reasons for legged robots' superior mobility compared to their wheeled and tracked counterparts is their freedom to choose contact points with the environment [1]. This freedom enables them to traverse smoothly and successfully in challenging terrains, step over obstacles and climb stairs.

Researchers have been trying to understand and formalize the underlying mechanism of animal locomotion for decades. One of the most influential and early works in dynamic legged locomotion is Raibert's foundational work [2] with the dynamically balancing hoppers at MIT Leg Lab. These hopping robots showed impressive dynamic locomotion capabilities with simple, intuitive control laws. Later, Full and Koditschek introduced the templates and anchors framework [3] and argued that simple yet expressive models called templates could capture different locomotion behaviors. These template models are used to realize the target behaviors on the complex legged systems: anchors. One of the earliest legged robots capable of traversing rough terrain with onboard power is the biologically inspired RHex [4], which showed remarkable mobility in challenging environments with a simple mechanical design. The hierarchical template-based control framework contributed to RHex's dexterity and capabilities even further [5, 6]. One of the most well-known template models is the Spring-Loaded Inverted Pendulum (SLIP) [7] model, which essentially captures the center of mass dynamics of running animals. The SLIP model influenced many robot designs [8, 9, 10] over the years. It is also used as the target model by different legged systems [11, 12, 13] and used to perform multi-step planning [14, 15].

On the other hand, a handful of successful quadruped robot platforms were built that pushed the limits of legged locomotion in recent years. For example, BigDog [16] is a heavy, combustion engine powered quadruped which showed one of the first significant rough terrain capabilities for a quadruped robot. It could carry over 150 kg in payload and run for more than 2 hours. Next, the early quadruped HyQ [17] demonstrated capable locomotion with combining hydraulic and electric actuators. A recently improved version of this robot, HyQReal [18] showed to be completely power autonomous. Furthermore, StarlETH [19] showed how highly compliant series elastic actuation can be used to perform efficient and versatile locomotion which was later improved and became the ANYmal [20] robot. In recent years, the MIT Cheetah series robots [21, 22, 23] demonstrated high speed impressive locomotion capabilities with efficient actuator and robot designs. Finally, the latest robot from the same group, the MIT Mini Cheetah [24], pushed the boundaries even further and became the first quadruped capable of doing a backflip.

Over the years of building new and more capable legged robots, research groups explored different footstep planning and foothold selection methods. Raibert's heuristic [2, 25] based foothold selection methods are still used by a vast majority of quadruped robots with different locomotion controllers [23, 26, 27, 28, 29] as is or with small modifications. This heuristic introduces the concept of the neutral point, which results in the net acceleration of zero on the body when a foot touch-downs at that position. This neutral point is unique to each forward speed, and for Raibert's early hoppers, placing the foot on this point results in a symmetric trajectory for the body.

Later, the DARPA Learning Locomotion program [30] aimed at accelerating locomotion research to make legged systems more useful and robust in challenging scenarios by organizing a competition involving visual foothold selection based on the environment. Using LittleDog [31] quadruped, different teams explored different footstep selection methods to traverse rough terrain. For instance, Kalakrishnan et al. [32] proposed a control system that learned optimal foothold choices from expert demonstrations, while Zucker et al. [33] chose a discrete search algorithm based method to search footholds and then optimize trajectories over them. However, both methods require pre-processing of the environment before each run and cannot make online foothold selection. In recent years, vision based foothold selection methods [34, 35, 36, 37, 38] become much more popular. Nonetheless, most of the work only focuses on finding safe footholds that quadrupeds can step on, without considering the effects of footstep locations on locomotion.

Feasible wrench based methods are another approach for foothold selection and locomotion controllers for legged robots. Orsolino et al. [39] adapted the feasibility factor idea [40] from the robotic grasping community to formulate an actuation and friction aware motion planner that is statically stable for the HyQ robot. Boussema et al. [41] introduced the Feasible Impulse Set (FIS) to quantify a leg's capability (utility) based on its current position, actuation limits, and friction cone. FIS can be used to decide when to lift off or touch down a leg accordingly without predetermined contact timings.

In this thesis, we focus on the effects of foot placements on the force output capability of the body. In the state of the art quadruped robots, speed and posture are mainly controlled at the stance phase by producing desired forces on the robot, while foot placements are not used to assist the stance controllers. Our motivation is to use foothold selection to help stance controllers in order for them to control the speed and posture of the robot more effectively and robustly. In doing so, we aim to make stance controllers perform more efficiently for quadrupeds by not pushing the motors to their limits. Also, stance controllers can be more robust against disturbances because the selected footholds will give the most flexibility in force output capability against unexpected disturbances.

1.2 Contributions

In this thesis, we propose a new foothold selection method based on feasible force sets that find footholds considering each leg's friction cone, motor limits, and leg geometry. Our method uses each leg's combined force output capability on the body coordinates to select the footholds.

We evaluate our method on a planar quadruped robot in simulation and compare it with the widely used Raibert's heuristic for foothold selection.

In this thesis, we show that our method

- reduces the amount of time it takes the stance controller to stabilize the robot,
- minimizes the amount of time the motors spend at their torque limits,
- helps the amount of energy spent while stabilizing the robot,

for the scenario of coming to a stop from a broad range of initial conditions compared to Raibert's method.

1.3 The Outline of the Thesis

First, Chapter 2 introduces the planar quadruped (PQ) model robot and details the model's dynamics and the swing-leg and stance controllers we use in our work.

Then Chapter 3 introduces our foothold selection method for quadruped robots based on Feasible Force Sets. After giving an overview of our approach, we detail our framework for selecting the best foothold for the desired movement considering friction constraints, motor torque limits, and leg geometry.

In chapter 4, we evaluate our method using different performance measures. After defining each performance measure for locomotion, we show the correlation between our foot selection method and these metrics. Finally, we show that our method can produce foothold selections that enable the stance controller to balance to robot faster and more efficiently than Raibert's heuristic for foothold selection.

Finally, Chapter 5 concludes our work with a discussion and possible research directions to follow.

CHAPTER 2

PLANAR QUADRUPED ROBOT

In this chapter, we introduce the planar quadruped (PQ) robot model we use in our work in detail. We select the planar model instead of a full 3D model because we want to focus on the sagittal dynamics of the robot as many of the stability and controllability properties of legged locomotion can be studied within models constrained to the sagittal fore-aft plane. After we describe the system, we detail the dynamics of the model and finalize this chapter with flight and stance controllers for the PQ model.

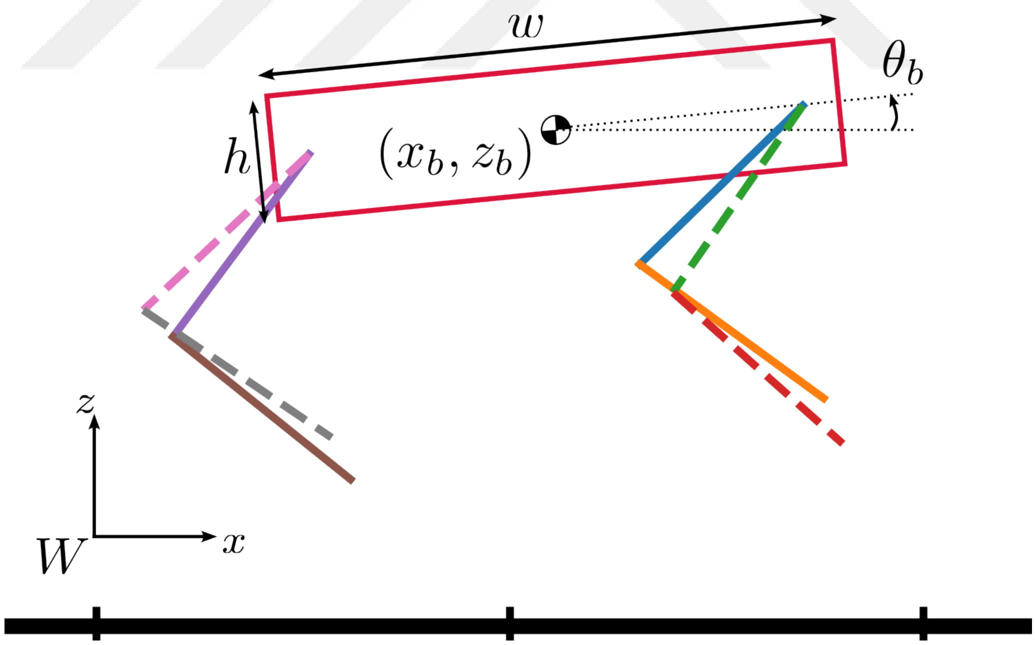


Figure 2.1: Planar Quadruped model. Solid and dashed lines correspond to the right and left legs respectively.

2.1 System Description

In this section, we detail the physical parameters and conventions of our PQ model. The model has 4 massless legs with 2 degrees of freedom (DoF): front right (FR), front left (FL), rear right (RR) and rear left (RL). The model have 11 DoFs where 8 of the DoFs are actuated and belong to the legs. In addition, the robot's body has 3 unactuated DoFs where it can move horizontally and vertically in the world and can rotate around its center of mass (CoM). All contacts in our model are point contacts and stance feet do not slip on the ground.

The complete PQ model in our simulation environment can be seen in Figure 2.1. Each 2-DoF leg has two identical motors where the motors are placed at the hip and the knee joints. The hip joints of each leg connects the legs to the body at $d = 0.24$ m away from the center of the robot on each side. Figure 2.2 depicts the motor placements and the details of legs.

Table 2.1 shows the physical parameters of the PQ robot. While selecting parameters for our robot, we tried to be consistent with the state of the art small quadrupeds such as MIT Mini Cheetah [24].

Table 2.1: Physical parameters for the planar quadruped robot model.

Mass (m)	10 kg
Inertia (I)	0.26 kg $\cdot$ m^2
Dimensions (w, h)	0.55 m, 0.12 m
Hip to CoM distance (d)	0.24 m
Upper leg length (l_1)	0.22 m
Lower leg length (l_2)	0.22 m
Motor torque limit (τ_{max})	17.0 Nm

2.2 Model Dynamics

In this section, we describe the dynamics of our PQ robot model. Legged robots cycle between two different phases called flight and stance. When there are no foot of the

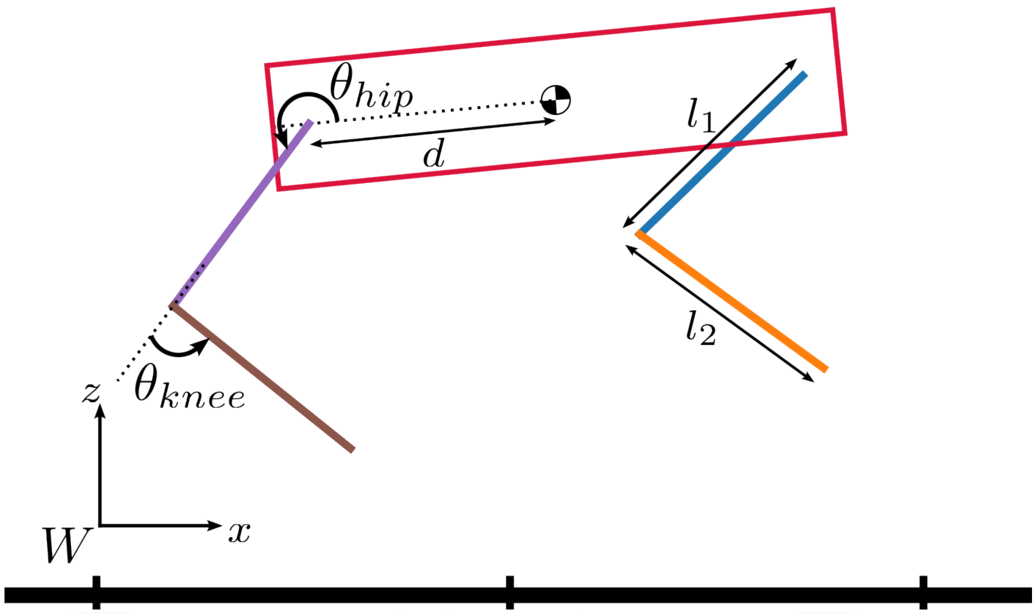


Figure 2.2: Planar Quadruped model's joint placements and joint angle conventions. Only right legs are visible in order to reduce clutter.

robot touching the ground, the system is called to be in the flight phase and the body of the robot follows a ballistic trajectory in the air. All of the legs are free to move in this phase. Stance phase (also called support phase) is when one or multiple legs support the weight of the robot's body. In this phase, stance legs are fixed to a position on the ground and ground reaction forces (GRFs) of each stance leg is included in the dynamics of the robot. We assume the GRFs are the only forces that act on the system other than gravitational force.

The dynamics of the PQ model are as follows,

$$\begin{aligned} m(\ddot{\boldsymbol{p}}_{\boldsymbol{b}} + \boldsymbol{g}) &= \sum_{i=1}^{s} \boldsymbol{F_i}, \\ I\ddot{\theta}_b &= \sum_{i=1}^{s} (\boldsymbol{p_i} - \boldsymbol{p_b}) \times \boldsymbol{F_i}, \end{aligned} \tag{2.1}$$

where s is the number of stance legs, $\boldsymbol{p_b} = (x_b, z_b)$ is the position of the body CoM in the world frame and θ_b is the angle of the body. m and I are the mass and inertia of the body respectively. $\boldsymbol{F_i}$ represents the GRF of the stance leg i and $\boldsymbol{p_i}$ represents the position of foot i.

We assume minimal inertial effects and relate hip and knee joint torques for leg i, $(\tau_{h,i}, \tau_{k,i})$, to leg forces by the following Jacobian mapping:

$$F_i = (J_i^T)^{-1} \begin{bmatrix} \tau_{h,i} \\ \tau_{k,i} \end{bmatrix}. \tag{2.2}$$

J_i is the Jacobian for leg i:

$$J_i = \begin{bmatrix} -l_1 \sin(\theta_b + \theta_{h,i}) - l_2 \sin(\theta_b + \theta_{h,i} + \theta_{k,i}) & -l_2 \sin(\theta_b + \theta_{h,i} + \theta_{k,i}) \\ l_1 \cos(\theta_b + \theta_{h,i}) + l_2 \cos(\theta_b + \theta_{h,i} + \theta_{k,i}) & l_2 \cos(\theta_b + \theta_{h,i} + \theta_{k,i}) \end{bmatrix}, \tag{2.3}$$

where $\theta_{h,i}$ and $\theta_{k,i}$ correspond to the hip and knee joint angles respectively.

Because we have massless legs, in order not to teleport the legs during flight phase, we use the following simple motor model with motor inertia I_m as our flight leg dynamics in our simulation:

$$\ddot{\theta}_{motor} = \tau_{motor}/I_m. \tag{2.4}$$

Similar to physical parameters of our PQ model, I_m is also selected to be consistent with the state of the art small quadruped robots' motor inertias.

2.3 Model Control

In this section, we detail the swing-leg and stance controllers for the PQ model. Swing-leg controller which controls the legs in the air, is active for a leg when its foot does not touch the ground, while stance controller is only active in the stance phase: when at least one of the feet is in contact with the ground.

2.3.1 Swing-leg Controller

Swing-leg controller drives the feet of the swing legs in the air to their desired positions. Given a desired position for the foot of leg i, $\boldsymbol{p_i^*}$, relative to the hip joint, we want to move the position of the foot $\boldsymbol{p_i}$ to its desired position. We calculate desired angles for hip and knee joints $\theta^*_{h,i}$ and $\theta^*_{k,i}$ respectively by using inverse kinematics. Then, the following PD control law moves the joints to their desired angles by mapping the error directly to the motor torques

$$\begin{bmatrix} \tau_{h,i} \\ \tau_{k,i} \end{bmatrix} = \begin{bmatrix} K_{p,h}(\theta^*_{h,i} - \theta_{h,i}) + K_{d,h}(\dot{\theta}^*_{h,i} - \dot{\theta}_{h,i}) \\ K_{p,k}(\theta^*_{k,i} - \theta_{k,i}) + K_{d,k}(\dot{\theta}^*_{k,i} - \dot{\theta}_{k,i}) \end{bmatrix}, \tag{2.5}$$

where $K_{p,h}$ and $K_{d,h}$ are the proportional and derivative gains for the hip controller, while $K_{p,k}$ and $K_{d,k}$ are the gains for the knee controller.

2.3.2 Stance Controller

In stance phase, we want to find the the GRFs $\boldsymbol{F}$ to drive the body to its desired pose. Our stance controller is based on the works of Bledt et. al. on MIT Cheetah 3 [23] and Focchi et. al. [42] on HyQ robot. Since our PQ model has massless legs, we ignore the effects of legs in our control model. We use a linear relationship between the GRFs from the stance legs of the robot and PQ model's translational and angular acceleration of the body. The controller model is represented as follows,

$$\underbrace{\begin{pmatrix} I_3 & \cdots & I_3 \\ [\boldsymbol{p_1} - \boldsymbol{p_b}]_\times & \cdots & [\boldsymbol{p_4} - \boldsymbol{p_b}]_\times \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \boldsymbol{F_1} \\ \vdots \\ \boldsymbol{F_4} \end{pmatrix}}_{F} = \underbrace{\begin{pmatrix} m(\ddot{\boldsymbol{p}}_{\boldsymbol{b}} + \boldsymbol{g}) \\ \boldsymbol{I}\ddot{\boldsymbol{\theta}}_{\boldsymbol{b}} \end{pmatrix}}_{b}, \tag{2.6}$$

where I_3 is the identity matrix of size 3. $\boldsymbol{p_b}$ and $\boldsymbol{\theta_b}$ corresponds to the position of the CoM of the body and the rotation of the body respectively written in full 3D coordinates. $[\boldsymbol{p_i} - \boldsymbol{p_b}]_\times$ represents the skew-symmetric matrix for the cross product $(\boldsymbol{p_i} - \boldsymbol{p_b}) \times \boldsymbol{F_i}$ where $\boldsymbol{p_i}$ is the foot position of leg i and $\boldsymbol{F_i}$ is the force acting on the foot of leg i. $\boldsymbol{g}$ is the gravity vector, I is the inertia of the body and m is the total mass of the body.

In order to calculate the desired translational and angular acceleration of the body from given desired body position ($\boldsymbol{p}^*_{\boldsymbol{b}}$), angle w.r.t. world ($\boldsymbol{\theta}^*_{\boldsymbol{b}}$), the translational speed ($\dot{\boldsymbol{p}}^*_{\boldsymbol{b}}$), and angular velocity ($\dot{\boldsymbol{\theta}}^*_{\boldsymbol{b}}$); we use the following PD control law

$$\begin{bmatrix} \ddot{\boldsymbol{p}}^*_{\boldsymbol{b}} \\ \ddot{\boldsymbol{\theta}}^*_{\boldsymbol{b}} \end{bmatrix} = \begin{bmatrix} K_{p,p}(\boldsymbol{p}^*_{\boldsymbol{b}} - \boldsymbol{p_b}) + K_{d,p}(\dot{\boldsymbol{p}}^*_{\boldsymbol{b}} - \dot{\boldsymbol{p}}_{\boldsymbol{b}}) \\ K_{p,\theta}(\boldsymbol{\theta}^*_{\boldsymbol{b}} - \boldsymbol{\theta_b}) + K_{d,\theta}(\dot{\boldsymbol{\theta}}^*_{\boldsymbol{b}} - \dot{\boldsymbol{\theta}}_{\boldsymbol{b}}) \end{bmatrix}, \tag{2.7}$$

where $K_{p,p}$ and $K_{d,p}$ are the diagonal matrices for proportional and derivative gains for the position of the CoM of the body, while $K_{p,\theta}$ and $K_{d,\theta}$ are the proportional and derivative gains for the angle of the body with respect to the world frame.

In this balance controller, we try to distribute the leg forces (GRFs) to drive the approximate CoM dynamics to the desired dynamics given by

$$\boldsymbol{b_{desired}} = \begin{bmatrix} m(\ddot{\boldsymbol{p}}_{\boldsymbol{b}}^{*} + \boldsymbol{g}) \\ \boldsymbol{I}\ddot{\boldsymbol{\theta}}_{\boldsymbol{b}}^{*} \end{bmatrix}, \tag{2.8}$$

where $\ddot{\boldsymbol{p}}_{\boldsymbol{b}}^{*}$ and $\ddot{\boldsymbol{\theta}}_{\boldsymbol{b}}^{*}$ are calculated in (2.7).

In the following optimization problem, we try to find the GRFs $\boldsymbol{F}$ that will drive the CoM dynamics to the calculated desired value $b_{desired}$ step by step while enforcing constraints such as motor torque limits and friction cone. The first variable of the cost function for our optimization problem

$$(A\boldsymbol{F} - \boldsymbol{b_{desired}})^T S(A\boldsymbol{F} - \boldsymbol{b_{desired}}) \tag{2.9}$$

tries to minimize the difference to the $b_{desired}$ where S matrix can be configured as giving more importance to tracking the position or the orientation of the body. While driving CoM dynamics to the desired $b_{desired}$ value, we want to minimize the total GRF we use

$$\alpha||\boldsymbol{F}||^2 \tag{2.10}$$

where α is a parameter which influences how much importance should be given to minimizing GRFs. We also want to minimize the difference between each consecutive GRFs

$$\beta||\boldsymbol{F} - \boldsymbol{F}_{\boldsymbol{prev}}^{*}||^2, \tag{2.11}$$

where β is a parameter dictating the importance of this effect. Combining (2.9), (2.10), and (2.11) with the constraints we arrive at the following optimization problem

$$\begin{aligned} \boldsymbol{F}^{*} = \min_{F} \quad & (A\boldsymbol{F} - \boldsymbol{b_{desired}})^T S(A\boldsymbol{F} - \boldsymbol{b_{desired}}) + \alpha||\boldsymbol{F}||^2 + \beta||\boldsymbol{F} - \boldsymbol{F}_{\boldsymbol{prev}}^{*}||^2, \\ \text{s.t.} \quad & C\boldsymbol{F} \leq \boldsymbol{d}, \end{aligned} \tag{2.12}$$

where the constraints $C\boldsymbol{F} \leq \boldsymbol{d}$ enforce the friction cone constraints and actuation limitations of our robot to make sure the GRFs stay inside the friction cone and feasible force limits. We also enforce that non-stance legs produce $\boldsymbol{F} = \mathbf{0}$ during flight. The resulting GRFs, $\boldsymbol{F}^*$, are converted to motor torques for each leg i by

$$\tau = J_i^T \boldsymbol{F_i^*}, \tag{2.13}$$

ignoring the leg dynamics. J_i is the Jacobian and $\boldsymbol{F_i^*}$ is the optimized GRFs for leg i.

CHAPTER 3

FEASIBLE FORCE SET BASED FOOTHOLD SELECTION

In this chapter, we introduce our Feasible Force Set (FFS) based foothold selection framework. First, we motivate and give an overview of our approach for foothold selection considering the actuator limits, leg configuration and friction constraints. Then, we formally introduce the notions of FFS and FFS series. We conclude this chapter by defining how we select footholds with a detailed analysis of different selection methods.

3.1 Foothold Selection Framework

Foot placements limit legged robots' ability to perform certain behaviors during the stance phase. Feet contact points influence the forces and the moment that each leg can produce on the body. Our primary motivation is to find footholds that will help the stance controllers to perform the desired behavior robustly and efficiently.

3.1.1 Constraints and Feasible Force Sets for Individual Legs

Three main constraints shape the set of forces we can produce on the body by each leg: friction cone restrictions, motor torque limitations, and the geometry of the leg and its configuration. Friction constraints dictate the forces exerted by a leg in order to keep the foot fixed to the ground. If these constraints are violated, the foot starts sliding on the ground, possibly leading the robot to fall down. In order to prevent this, controllers need to make sure that the forces produced by each leg respect the friction constraints of the terrain. The second constraint is the torque limitations of the motors. Motor torques directly influence the magnitude of forces we can produce by each leg. Lastly, the geometry of the leg and the joint configuration affects the forces and the moment legs can produce on the body. While it is impossible to modify the ground friction coefficient or increase the motor torque limitations without installing new hardware, we can select footholds to shape the resultant force on the body.

While there are varying names in literature for the set of forces that can be produced by a single leg with the aforementioned constraints, we use the name and formalization of *Feasible Force Set* (FFS) from Boussema et al. [41]. For completeness, we include the definition of a FFS for a leg i as

$$\mathcal{F}_i := \{\boldsymbol{F_i} \mid C\boldsymbol{F_i} \leq 0\}, \tag{3.1}$$

where C represents the constraints of the leg. Three example FFSs with different leg configurations, namely different horizontal foot positions with respect to hip, for the same leg can be seen in Figure 3.1. In other words, the foot position is given in the respecting hip coordinates.

3.1.2 Feasible Force Sets on Body Coordinates

Although an FFS is computed for a single leg; ultimately we are interested in the combined force and moment capability of stance legs on the body while respecting the constraints of each leg. We define the FFS on the body $\mathcal{F}_b$ as

$$\mathcal{F}_b := \{[\boldsymbol{F_b}, \tau_b] \mid [\boldsymbol{F_b}, \tau_b] = f_b(\boldsymbol{F_{i=1}}, \ldots),\ \boldsymbol{F_i} \in \mathcal{F}_i,\ i \in \{1, \ldots n\}\}, \tag{3.2}$$

where $\mathcal{F}_i$ represents the FFS for a leg i and n is the number of stance legs. $f_b(\boldsymbol{F_i}, \ldots)$ is a function that maps the forces that can be produced by the stance legs to their combined effect on the body.

3.1.3 Optimization Framework for Foothold Selection

In our framework, we aim to select the set of footholds $\boldsymbol{p_f}$, in air, which would minimize the cost of locomotion for a desired behavior

$$\boldsymbol{p_f} = \arg\min_{\boldsymbol{p_f}} C_k(\boldsymbol{p_f}), \tag{3.3}$$

where $C_k(\boldsymbol{p_f})$ is a function which scores the cost of locomotion for a given set of footholds $\boldsymbol{p_f}$. We introduce different foothold selection methods in Section 3.3 and evaluate them with different performance measures in Chapter 4.

3.2 Computing Feasible Force Sets

In this section, we formally define how to compute an FFS for a single leg and introduce how the capabilities and constraints of each leg maps to the body coordinates.

3.2.1 Individual Legs

In order to calculate the set of forces that is feasible to produce with a given leg i, we need to consider the actuator limits:

$$\tau_{min} \leq J_i^{\mathbf{T}} \boldsymbol{F} \leq \tau_{max}, \tag{3.4}$$

where τ_{min} and τ_{max} correspond to the minimum and the maximum torques actuators can provide. J_i corresponds to the leg i Jacobian. In order not to slip with the foot, forces should remain in the friction cone:

$$U\boldsymbol{F} \leq \mathbf{0}, \tag{3.5}$$

where $U = \left(\begin{smallmatrix} 1 & -\mu \\ -1 & -\mu \end{smallmatrix}\right)$ and μ is the friction coefficient. Combining friction constraints with actuator limits, we come up with the following inequality

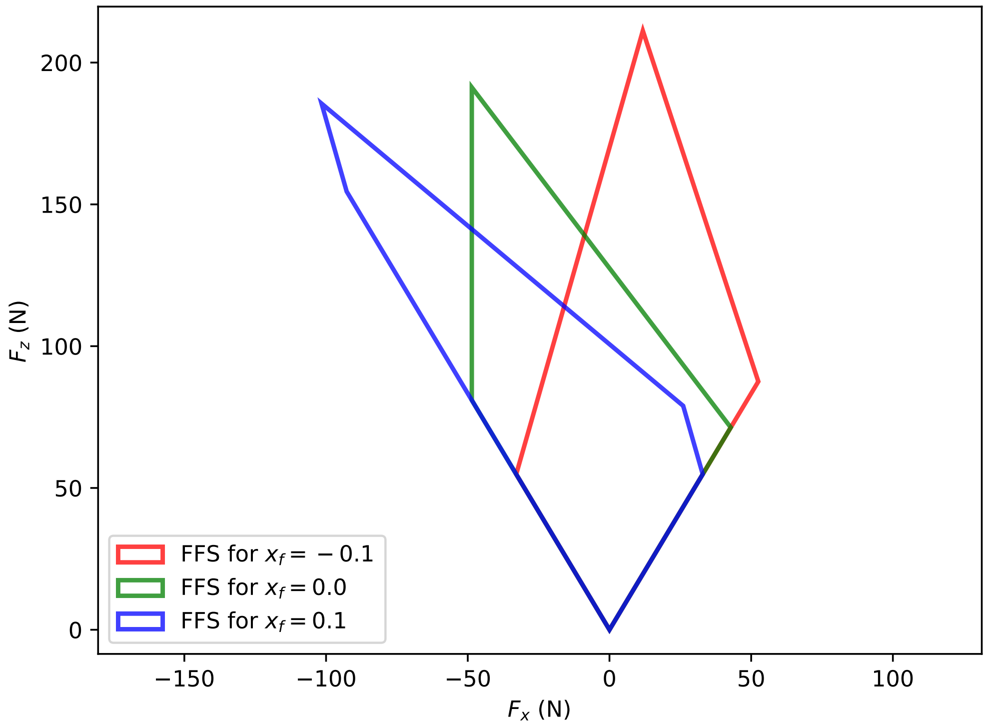


Figure 3.1: Three example Feasible Force Sets for different foot positions $\boldsymbol{p_f} = (x_f, z_f)$. The foot positions are given in the respecting leg's hip coordinates. For all foot positions, $z_f = -0.35$ m.

$$P\boldsymbol{F} \leq \boldsymbol{h}, \tag{3.6}$$

where $P = [U^T, J, -J]^T$, $\boldsymbol{F} = [F_x,\ F_z]^T$, $h = [0,\ \tau_{max}^T,\ -\tau_{min}^T]^T$. Finally, an FFS can be formally defined as:

$$\mathcal{F} = \{\boldsymbol{F} \mid P\boldsymbol{F} \leq \boldsymbol{h}\}. \tag{3.7}$$

Three example Feasible Force Sets for our PQ model for different leg configurations can be seen in Figure 3.1.

3.2.2 Body Coordinates

We are interested in finding a mapping from the constraints and limitations of each leg's force and moment output capacity to their combined effect on the body. In this work, we only use two legs, specifically front left and rear right legs to perform the desired behaviors to show the robustness of the selected footholds. Hence, the combined FFS in the body, $\mathcal{F}_b$ for two legs can be computed as

$$\mathcal{F}_b = \{[F_x, F_z, \tau] \mid [F_x, F_z, \tau] = f_b(\boldsymbol{F_f}, \boldsymbol{F_r}),\ \boldsymbol{F_f} \in \mathcal{F}_f,\ \boldsymbol{F_r} \in \mathcal{F}_r\}, \tag{3.8}$$

where $\mathcal{F}_f$ and $\mathcal{F}_r$ represent the front and rear leg's FFS respectively. The function $f_b(F_f, F_r)$ maps the forces that can be produced by legs to their combined effect on body and it is defines as

$$f_b(\boldsymbol{F_f}, \boldsymbol{F_r}) = \begin{bmatrix} F_{f,x} + F_{r,x} \\ F_{f,z} + F_{r,z} \\ \boldsymbol{v_f} \times \boldsymbol{F_f} + \boldsymbol{v_r} \times \boldsymbol{F_r} \end{bmatrix}, \tag{3.9}$$

where $F_{i,x}$ and $F_{i,z}$ correspond to the x and z components of the force $\boldsymbol{F_i}$. $\boldsymbol{v_f}$ and $\boldsymbol{v_r}$ are the vectors from front and rear foot positions to the CoM of the robot's body.

In this work, for simplicity, we only focus on the combined force capacity of the stance legs on the body and leave the analysis and effects of the net moment on the body for a desired behavior as future work.

3.3 Foothold Selection Method

In this section, we introduce our two-stage foothold selection method in detail. Better footholds help shorten the time for the stance controller to reach the desired goal state for a desired behavior and keep the motor torques away from their limits.

3.3.1 Reference Model and Controller

In order to find the set of footholds that will help the stance controllers perform a desired movement more effectively and efficiently, we introduce the reference model and controller in this section. We aim to use the reference model and controller as a template for finding a set of reference forces that the PQ robot should follow to perform a desired behavior. Our foothold selection framework tries to optimize the location of contact points considering the contact constraints mentioned earlier to maximize the ability and feasibility of the body's force output capabilities for a given set of reference forces. On a high level, for any movement or goal state we want to achieve with our PQ model, we initialize the reference model with the same initial state as our PQ model at each apex. Then, we use the reference controller to produce reference forces we want to follow with our PQ model to achieve the goal state. We then select the footholds by comparing the reference forces with the FFS series of each possible foothold by using a scoring method to determine the footholds, which would give the PQ model the best chance to output the reference forces. Selecting the best footholds this way enables the stance controller to follow reference forces more closely while being robust against disturbances by keeping the PQ robot away from the limits of its FFS at each instant.

The reference model consists of a single rigid body with the same mass and inertia as the PQ model. It has no legs, and the main difference between the reference and PQ model is that in the reference model, we can directly apply the desired forces and the torque on the body.

The reference model has 3 DoF where the body can move in the world horizontally, vertically and it can rotate around its CoM. The dynamics of the reference model is

$$
\begin{aligned}
m(\ddot{\boldsymbol{p}}_{\boldsymbol{r}} + \boldsymbol{g}) &= \boldsymbol{F}_{\boldsymbol{r}}, \\
I\ddot{\theta}_r &= \tau_r,
\end{aligned}
\tag{3.10}
$$

where $\boldsymbol{F}_{\boldsymbol{r}}$ and τ_r are the force and torque inputs to the reference model. $\boldsymbol{p}_{\boldsymbol{r}} = (x_r, z_r)$ and θ_r are the position and orientation of the reference body in the world frame. I and m are the inertia and mass of the reference body, $\boldsymbol{g}$ is the gravity vector. The

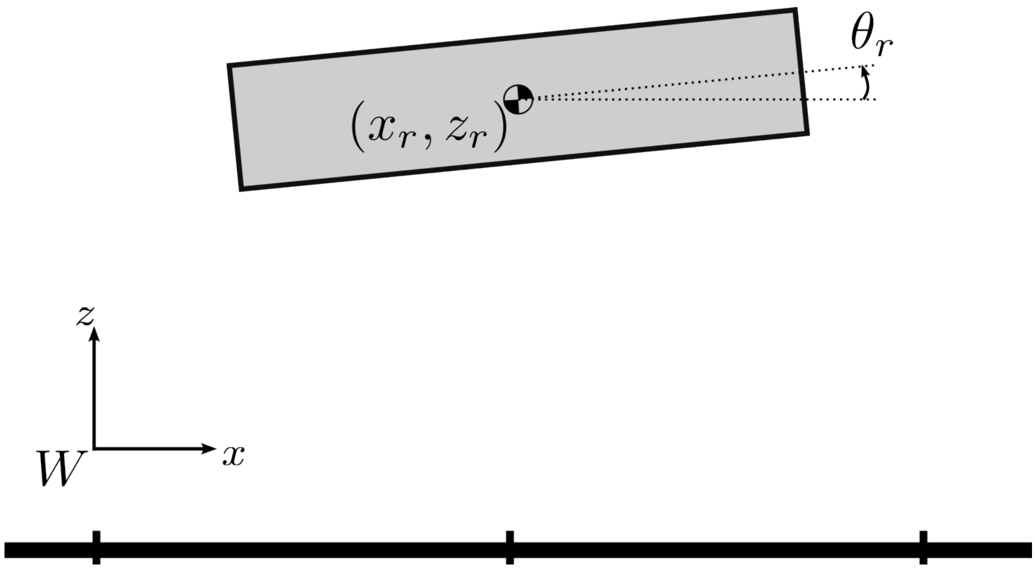


Figure 3.2: Reference model.

reference model can be seen in Figure 3.2.

In order to control the reference model, we have a reference controller loosely based on the PQ stance controller, described in Section 2.3.2. We limit the magnitude of forces and torque we can apply to the body with the reference controller to create realistic forces that the PQ stance controller can follow. In addition, we limit the forces that can be produced on the body by considering the friction cone constraints because the output forces from the reference controller are used to select footholds for the PQ model. Because we aim to find the best possible footholds that would enable the stance controller to follow the reference controller closely, if some of the output forces from the reference controller are out of the friction cone, it is impossible to realize these reference forces with the stance controller.

The reference controller model is

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{A_r} \underbrace{\begin{pmatrix} F_{r,x} \\ F_{r,z} \\ \tau_r \end{pmatrix}}_{\boldsymbol{F_r}} = \underbrace{\begin{pmatrix} m(\ddot{\boldsymbol{p}}_{\boldsymbol{r}} + \boldsymbol{g}) \\ I\ddot{\theta}_r \end{pmatrix}}_{\boldsymbol{b_r}}, \tag{3.11}$$

where p_r and θ_r corresponds to the reference body's position of the CoM and orientation respectively. $\boldsymbol{F_r} = [F_{r,x}, F_{r,z}]$ is the forces and the τ_r is the torque we can we can produce on the reference body.

Similar to the stance controller; given desired reference body position ($\boldsymbol{p_r^*}$), orientation (θ_r^*), the translational speed ($\dot{\boldsymbol{p}}_{\boldsymbol{r}}^{\boldsymbol{*}}$), and angular velocity ($\dot{\theta}_r^*$) we use the following PD control law to calculate the desired reference translational and angular acceleration of the body, $\boldsymbol{b_{r,desired}}$:

$$\begin{bmatrix} \ddot{\boldsymbol{p}}_{\boldsymbol{r}}^{\boldsymbol{*}} \\ \ddot{\theta}_r^* \end{bmatrix} = \begin{bmatrix} K_{p,p}(\boldsymbol{p_r^*} - \boldsymbol{p_r}) + K_{d,p}(\dot{\boldsymbol{p}}_{\boldsymbol{r}}^{\boldsymbol{*}} - \dot{\boldsymbol{p}}_{\boldsymbol{r}}) \\ K_{p,\theta}(\theta_r^* - \theta_r) + K_{d,\theta}(\dot{\theta}_r^* - \dot{\theta}_r) \end{bmatrix}, \tag{3.12}$$

where $K_{p,p}$, $K_{d,p}$, $K_{p,\theta}$, and $K_{d,\theta}$ are the diagonal matrices for proportional and derivative gains for the position and orientation of the reference body.

In the following optimization problem, we try to find the forces and the torque to apply to the reference body to drive the reference CoM dynamics to the desired reference dynamics calculated at (3.12).

$$\begin{aligned} \boldsymbol{F_{out}^*} = \min_{\boldsymbol{F_{out}}} \quad & (A_r\boldsymbol{F_{out}} - \boldsymbol{b_{r,desired}})^T S_r (A_r\boldsymbol{F_{out}} - \boldsymbol{b_{r,desired}}) + \alpha_r ||F_{out}||^2, \\ \text{s.t.} \quad & ||\boldsymbol{F_{out}}|| \le \boldsymbol{t_F}, \quad F_{r,z} \ge 0, \quad U\boldsymbol{F_r} \le \boldsymbol{0}, \end{aligned} \tag{3.13}$$

where $\boldsymbol{F_{out}} = [F_r, \tau_r]$. The first optimization variable tries to minimize the difference to the $\boldsymbol{b_{r,desired}}$. S_r matrix changes the importance of tracking the position or the orientation of the reference body. The second optimization variable minimizes the forces and the torque we produce on the reference body. α_r controls the importance of minimizing the output forces and the torque. The first constraint limits the magnitude of the forces and the torque while the second constraint makes sure that the forces do not push the robot to the ground. The last constraint enforces the friction cone constraints to make sure that the output forces stay inside the friction cone.

3.3.2 Feasibility Metric for Instantaneous Force Vectors

In this subsection, we introduce and compare different methods to evaluate the distance between a single reference force and a single FFS, where the reference force is a vector and an FFS is a polygon. In Figure 3.3, we show two example FFS polygons with a reference force. We are interested in finding a method that would represent an FFS's feasibility to produce a reference force. In this thesis, we focus on the following methods between an FFS $\mathcal{F}$ and a reference force $\boldsymbol{F_r}$ to select the best FFS for a reference force:

- the angle difference between the vector from origin to the polygon of $\mathcal{F}$'s center and $\boldsymbol{F_r}$, $M_{angle}\left(\mathcal{F}, F_r\right)$,
- the minimum distance of the $\boldsymbol{F_r}$ to an edge of the polygon of $\mathcal{F}$, $M_{edge}\left(\mathcal{F}, \boldsymbol{F_r}\right)$,
- the distance to the center of the polygon of $\mathcal{F}$ from $\boldsymbol{F_r}$, $M_{center}\left(\mathcal{F}, \boldsymbol{F_r}\right)$,
- if the $\boldsymbol{F_r}$ is inside of the polygon of $\mathcal{F}$, $M_{in}\left(\mathcal{F}, \boldsymbol{F_r}\right)$.

The method $M_{angle}(\mathcal{F}, \boldsymbol{F_r})$ uses the angle difference between a vector from the origin to the center of a FFS polygon $\mathcal{F}$ and a reference force $\boldsymbol{F_r}$

$$M_{angle}\left(\mathcal{F}, \boldsymbol{F_r}\right) = d_\theta(\boldsymbol{F_r}, \boldsymbol{c_{\mathcal{F}}}), \tag{3.14}$$

where $\boldsymbol{c_{\mathcal{F}}}$ is the vector to the center of the FFS from origin and $d_\theta(\boldsymbol{v_1}, \boldsymbol{v_2})$ computes the angle difference vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$. Example visualizations for the $M_{angle}\left(\mathcal{F}, \boldsymbol{F_r}\right)$ can be seen in Figure 3.4a.

The method $M_{edge}\left(\mathcal{F}, \boldsymbol{F_r}\right)$ uses the minimum distance of a reference force $\boldsymbol{F_r}$ to the edges of an FFS polygon $\mathcal{F}$

$$M_{edge}\left(\mathcal{F}, \boldsymbol{F_r}\right) = \min_{d_{vls}}\ d_{vls}(\boldsymbol{F_r}, e_i)\ \mid\ e_i \in E_{\mathcal{F}}, \tag{3.15}$$

where $E_{\mathcal{F}}$ is the set of all edges of the FFS and $d_{vls}(v, e)$ is the distance between a vector v and a line segment e. Example visualizations for this method can be seen in Figure 3.4b.

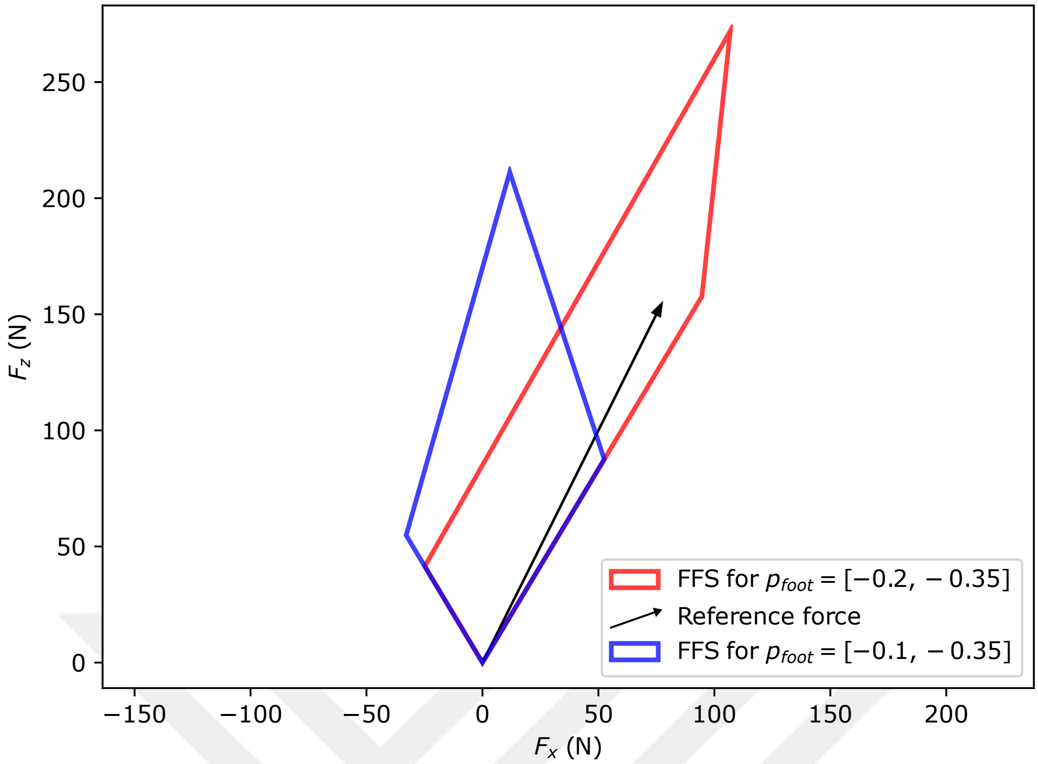


Figure 3.3: An example reference force $\boldsymbol{F_r} = [75, 150]$ with two Feasible Force Sets representing two different relative foot positions to hip.

The method $M_{center}\left(\mathcal{F}, \boldsymbol{F_r}\right)$ uses the distance between the center of a FFS polygon $\mathcal{F}$ and a reference force $\boldsymbol{F_r}$

$$M_{center}\left(\mathcal{F}, \boldsymbol{F_r}\right) = d(\boldsymbol{F_r}, \boldsymbol{c_{\mathcal{F}}}), \tag{3.16}$$

where $\boldsymbol{c_{\mathcal{F}}}$ is the vector to the center of the FFS from origin and $d(\boldsymbol{v_1}, \boldsymbol{v_2})$ is the distance between vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$. Example visualizations for this method can be seen in Figure 3.4c.

While the aforementioned methods give useful information between an FFS and a reference controller, we found that the single most important factor between a reference force and an FFS is if the reference force is inside of the FFS. The method $M_{in}\left(\mathcal{F}, \boldsymbol{F_r}\right)$ returns if the reference force $\boldsymbol{F_r}$ is inside of an FFS polygon $\mathcal{F}$

$$M_{in}\left(\mathcal{F}, \boldsymbol{F_r}\right) = d_{in}(\mathcal{F}, \boldsymbol{F_r}), \tag{3.17}$$

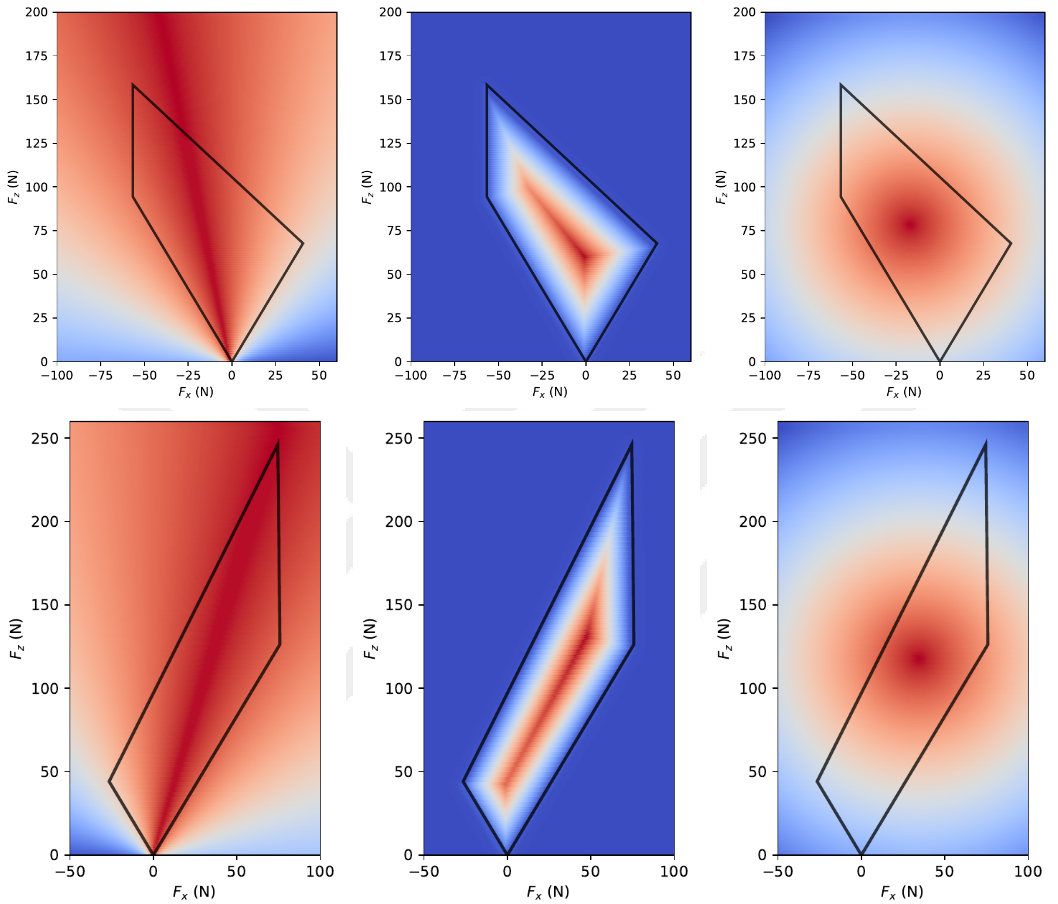
where $d_{in}(\mathcal{P}, \boldsymbol{v})$ returns 1 if the vector $\boldsymbol{v}$ is inside of the polygon $\mathcal{P}$, 0 otherwise. Two different FFSs with a reference force where one FFS can produce the reference force while the other FFS cannot can be seen in Figure 3.3.

3.3.3 Performance Metric for Entire Trajectories

An FFS by definition only captures a single leg configuration where the relative foot position to hip is fixed. However, while legged robots' feet are fixed to the ground for the duration of the stance phase, relative positions of hips to feet change over time with the motion of the body. In order to capture this sweeping motion of the leg, we calculate a series of Feasible Force Sets. An FFS series represent the set of forces that a leg can produce while following a trajectory.

We start by selecting the initial relative foot positions to hip in x direction which represents if the body will go forward or backward relative to the feet. This selection is mainly based on the kinematic limits of the robot. For example, with our PQ model we find the starting values between -0.24 and 0.16 meters feasible in x direction with a reasonable touch-down height of 0.35 meters. In our work, all the final positions for each starting foot position in x direction is 0 meters, where the hip is exactly at the top of the foot. The starting height is selected as the target height for foot to touch the ground and the final height is selected as the nominal stance height for the robot. After selecting the initial and final foot positions, we take N linearly spaced samples between them and calculate an FFS for each sample. In our work, initial and final relative heights for the foot positions are also fixed. Thus, we only consider different initial foot positions in the x direction.

Formally, between initial relative foot position $\boldsymbol{p_i} = [x_i, z_i]$ and the final relative foot position $\boldsymbol{p_f} = [x_f, z_f]$, we linearly sample N foot positions and create a set $P_N\ (\boldsymbol{p_i}, \boldsymbol{p_f})$



(a) Visualization for angle difference to center method M_{angle} for two different FFSs.

(b) Visualization for distance to center method M_{center} for two different FFSs.

(c) Visualization for minimum distance to edges method M_{edge} for two different FFSs.

Figure 3.4: Visualization of different methods with different FFSs. Blue to red gradient correspond to increasingly desired F_r vectors for each method.

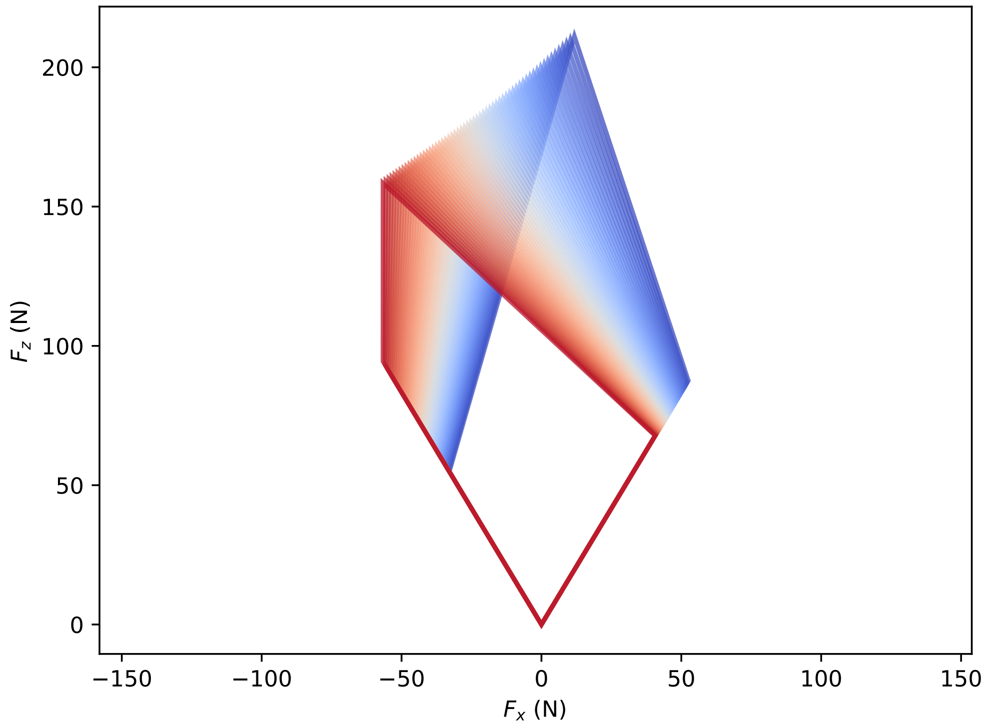


Figure 3.5: An example FFS series for initial relative foot position with respect to hip $\boldsymbol{p_i} = [-0.1, -0.35]$, final relative foot position $\boldsymbol{p_f} = [0.0, -0.30]$, and sample count $N = 50$. The gradient from blue color to red color represents the samples from $\boldsymbol{p_i}$ to $\boldsymbol{p_f}$ respectively.

$$
\begin{aligned}
& \boldsymbol{p_n} \in P_N\ (\boldsymbol{p_i}, \boldsymbol{p_f}), \\
\text{s.t.}\quad & \boldsymbol{p_n} = [x_n, y_n], \\
& x_n = (1-n) \cdot x_i + n \cdot x_f, \\
& y_n = (1-n) \cdot y_i + n \cdot y_f \ \mid\ n \in \{{}^0/_N, {}^1/_N, \ldots, {}^N/_N\}.
\end{aligned} \tag{3.18}
$$

Because only the initial position of the foot x_i in x direction changes between different FFS series, x_i is sufficient to identify an FFS series. The FFS series $\mathcal{FS}$ for initial and final foot positions p_i, p_f is defined as

$$
\mathcal{FS}_{x_i} = \{\mathcal{F}_{p_n} \mid p_n \in P_N\ (p_i, p_f)\}, \tag{3.19}
$$

where $\mathcal{F}_{p_n}$ is the FFS for the foot position p_n.

An example FFS series can be seen in Figure 3.5. The FFS series for different footholds can be computed offline and stored for efficient and fast online computations.

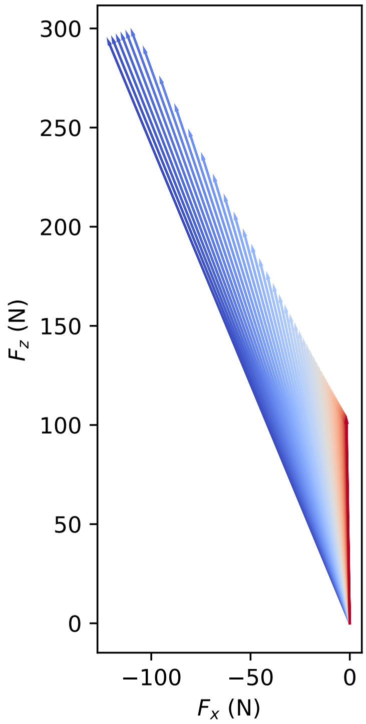


Figure 3.6: An example series of reference forces stopping a reference model with the initial speed of 0.9 m/s and the initial height of 0.5 m. The gradient from blue color to red color represents the first to last reference force respectively.

On each apex, we decide on the next set of footholds the PQ model should step on and start by initializing the reference model with the same state as the PQ model and simulate forward until the desired goal state is achieved with the reference controller. At this stage, reference controller calculates the reference forces for the desired movement of the body. An example series of reference forces can be seen in Figure 3.6. From these reference forces, we sample N of them where N is the number of FFSs calculated for each FFS series representing different foothold selections and denote these sampled reference forces with $\mathcal{FR}$.

The reference forces are then compared with the previously calculated FFS series of each foothold option to select the next footholds. While different legs can be directed to different footholds, in this work we move both of the legs that are going to touch-down to the same foothold relative to their hip joints.

We use a two stage selection method for deciding on the next footholds $\boldsymbol{p_f}$ for a desired behavior. However, because we move both of the feet to the same position and keep them at the same height with respect to their hip locations, we only select a single horizontal position, x_f, as a foothold. On the first stage, we compare different foothold selections using the method $M_{in}\left(\mathcal{F}, \boldsymbol{F_r}\right)$

$$x_f = \underset{x_f \,\in\, \mathcal{X}_{fh}}{\arg\min} \sum_{n=1}^{N} M_{in}\left(\mathcal{F}_{b,n}, \mathcal{FR}_n\right) \tag{3.20}$$

where $\mathcal{F}_{b,n}$ correspond to the combined FFS on the body of both touch-down legs' nth FFS in the FFS series for foothold x_f, $\mathcal{FS}_{x_f}$. $\mathcal{FR}_n$ is the nth reference force in the reference forces $\mathcal{FR}$. $\mathcal{X}_{fh}$ is the footholds for which an FFS series are previously computed offline.

If the number of reference forces inside of FFSs is same for two or more different footholds, we move onto the second stage of foothold selection. In our experiments, we find the best performing method after M_{in} is the angle difference method $M_{angle}\left(\mathcal{F}, \boldsymbol{F_r}\right)$ and it is used for selecting the next foothold among the footholds scored same on the first stage

$$x_f = \underset{x_f \,\in\, \mathcal{X}_{fh,first}}{\arg\min} \frac{1}{N} \sum_{n=1}^{N} M_{angle}\left(\mathcal{F}_{b,n}, \mathcal{FR}_n\right) \tag{3.21}$$

where $\mathcal{X}_{fh,first}$ are the footholds that passed the first stage. We believe, one of the main reasons why M_{angle} performs better than other methods is that it captures the inclination of an FFS polygon's force output capability. We take the average of the angle differences between all reference forces and FFSs. The foothold with the minimum average angle difference is selected for the next relative touch-down location for the touch-down legs.

CHAPTER 4

SIMULATION RESULTS

In this chapter, we present our performance measures and experiments with the FFS based foothold selection framework. We evaluate our method on different performance measures for the scenario of landing and stabilizing the body from an initial condition. This scenario can capture different behaviors, such as coming to a stop from a leaping motion or high-speed running.

We conducted the simulations and implemented our controllers, methods, visualizations, and experiments in Python using Drake [43]. An example visualization of our simulation on recovery from a jump can be seen in Figure 4.1.

4.1 Performance Measures

In this section, we suggest three different performance measures to evaluate our method. The first measure is the time it takes for the stance controller to stabilize the robot from an initial condition. The second measure is the total time the motors spent on their torque limits. The final measure is the amount of energy spent by the motors in order to stabilize the robot completely.

4.1.1 Time to Stabilize the Body

The first measure we use to evaluate our framework is the total time it takes for the stance controller to stabilize the body from an initial condition starting from the first touchdown moment, the earliest time any one of the legs touchdown the ground. This measure is defined as

$$
\begin{aligned}
t_s = & \min_{t_i} \ t_i \\
\text{s.t. } & E_{\dot{x}}(t_i) \leq 0.1, \\
& E_{\dot{z}}(t_i) \leq 0.1, \\
& E_{\dot{\theta}}(t_i) \leq 0.1,
\end{aligned} \tag{4.1}
$$

where $E_{\dot{x}}(t_i)$, $E_{\dot{z}}(t_i)$, and $E_{\dot{\theta}}(t_i)$ are the errors of horizontal, vertical and angular velocity of the PQ model in an experiment at time t_i. This measure computes how fast the stance controller can stabilize the robot based on a given foothold selection. Because the stance controller is same for each experiment, we aim to capture the effects of the force capacity for a given foothold with this performance measure.

4.1.2 Time Spent by Motors at Their Torque Limits

One of the advantages of our framework is it tries to select footholds that would keep the motors away from their torque limits. Thus, the second measure we use to evaluate our framework is the the total time all of the motors spent at their torque limits starting from the touchdown moment to the moment the body is stabilized. This measure is defined as

$$
t_m = \sum_{i=1}^{M} \mathcal{T}(i), \tag{4.2}
$$

where M is the total number of motors at the stance legs and $\mathcal{T}(i)$ is the time motor i spent on its torque limits. Using this measure, we aim to understand if the foothold selection can provide the desired forces without pushing motors to their limits. If the motors are not on their limits, it is more plausible for PQ robot to handle unexpected disturbances.

4.1.3 Total Energy Spent While Stabilizing the Body

The final evaluation measure for our framework is the total energy spent by motors while stabilizing the body. Similar to previous performance measure, this measure only considers the energy spent from the first touchdown moment to the moment body is stabilized and it is defined as

$$E_m = \sum_{i=1}^{M} \mathcal{E}(i), \tag{4.3}$$

where M is the total number of motors at the stance legs and $\mathcal{E}(i)$ calculates the amount of energy the motor i spends by taking the integral of mechanical power of each motor, $P_i = \tau_i \cdot \dot{\theta}_i$ where τ_i and $\dot{\theta}_i$ are the torque and angular velocity of motors.

4.2 Evaluation of our Method

In this section, we evaluate our foothold selection method with the performance measures we discussed in Section 4.1. First, we show the correlation between our foothold selection method which we discussed in 3.3.3 and the aforementioned performance measures. Then, we compare our method with one of the most widely used methods for foothold selection for quadrupeds: Raibert's heuristic.

For the experiments, we created a set of footholds for feet relative to respecting hips in x direction, $\mathcal{X}_{fh}$

$$\begin{aligned} \mathcal{X}_{fh} := \{\, x_f \;\mid\; x_f \in & \{\, -0.24, -0.20, -0.16, -0.12, -0.08, \\ & -0.04, 0.0, 0.04, 0.08, 0.12, 0.16 \,\}\}. \end{aligned} \tag{4.4}$$

and created FFS series for each one of them offline. We selected these footholds in range $(-0.24, 0.16)$ m based on the kinematic limits of each leg. While experimenting with different sampling distances, we selected the sampling distance as 0.04 m because with lower values we saw minimal increase in performance and also using a larger sampling distance reduces the computation time. To show the robustness of the selected footholds, we focus on stabilizing the robot from different initial conditions with only two legs, specifically, front left and rear right leg. For evaluating our framework, we use initial conditions with different horizontal velocities and starting heights. Different orientations of the body introduces different hip heights, which

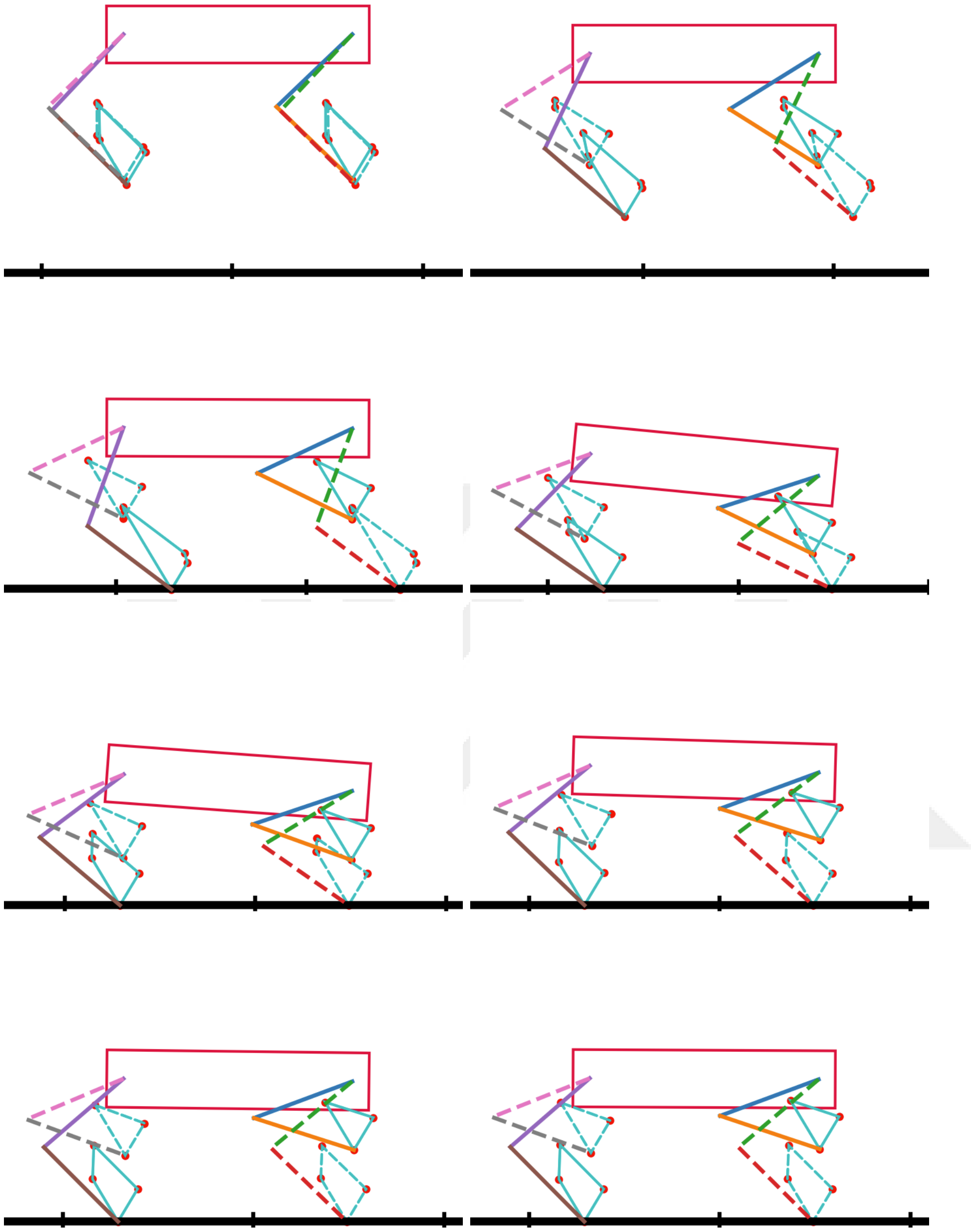


Figure 4.1: Example recovery from a jump using only two legs. Right legs are shown with solid lines, whereas left legs are shown as dashed lines. Solid and dashed cyan polygons show the FFS of that leg at that time; drawn only for visualization without scale. Time differences between consecutive frames are 0.09 seconds. Initial height $z = 0.5$ m, initial forward velocity $\dot{x} = 1.4$ m/s.

results in touch-down legs not touching-down at the same time. In this work, we do not focus on the angular velocity or the orientation of the body for simplicity.

4.2.1 Correlation Between the Selected Footholds and Performance Measures

In this section, we show that our footholds selected by our foothold selection method results in better performance in the performance measures described in Section 4.1. We evaluate our foothold selection method for three distinctive scenarios: stopping from an initial condition with positive velocity, negative velocity, and with no horizontal velocity.

Table 4.1: PQ robot recovering from the initial state $z = 0.50$ m, $\dot{x} = 0.9$ m/s and coming to a stop. The bold numbers represent the selected footholds and best performances at that performance measure.

x_f	0.04	0.08	0.12	0.16
M_{in}	**50**	**50**	43	42
M_{angle}	**6.28**	6.32	-	-
t_s	**0.30**	0.31	0.44	Fall
t_m	**0.19**	0.27	0.44	Fall
E_m	**34.07**	37.24	44.95	Fall

Table 4.2: PQ robot recovering from the initial state $z = 0.50$ m, $\dot{x} = -0.5$ m/s and coming to a stop. The bold numbers represent the selected footholds and best performances at that performance measure.

x_f	-0.24	-0.20	-0.16	-0.12	-0.08	-0.04
M_{in}	41	45	**50**	**50**	41	39
M_{angle}	-	-	**5.07**	7.89	-	-
t_s	Fall	Fall	**0.09**	0.11	0.26	0.41
t_m	Fall	Fall	**0.12**	0.15	0.22	0.41
E_m	Fall	Fall	**22.94**	23.72	27.19	37.95

Results from an experiment with the initial state of going forward with $\dot{x} = 0.9$ from height $z = 0.50$ can be seen in Table 4.1. For the footholds $x_f = \{0.04, 0.08\}$, number of reference forces that are inside of the FFS, M_{in}, are the same. Thus, we move onto the second stage for selecting a foothold, M_{angle}, which checks the angle difference between the reference force and a vector to the center of an FFS from origin. Both $x_f = 0.04$ and $x_f = 0.08$ have very similar angle differences and this is also visible in the performance measures. While they both have very similar time to stabilize, t_s, values; the best selection of our framework outperforms other selections on all measures. On the other hand, for $x_f = 0.16$, we have the lowest M_{in} score and that foothold selection results in the robot falling to the ground.

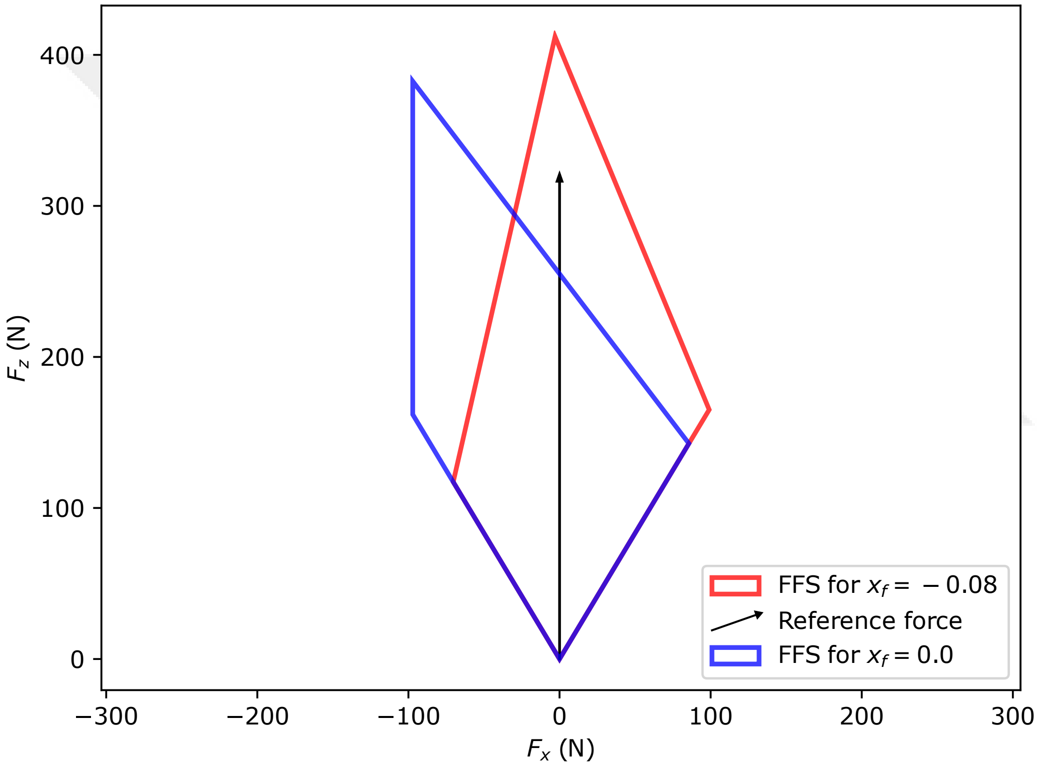


Figure 4.2: FFSs for two different foothold selections $x_f = -0.08$ and $x_f = 0.00$ in the body coordinates for two stance legs shown in red and blue polygons. The first reference force for the initial state $z = 0.53$ and $\dot{x} = 0.00$ m/s shown in black arrow. The body can produce the desired reference force when the front and rear stance legs touchdown at $x_f = -0.08$.

Results of the second experiment with the initial state of going backward with $\dot{x} = -0.5$ from height $z = 0.50$ can be seen in Table 4.2. Foot positions $x_f = -0.16$ and

$x_f = -0.12$ both score the same in M_{in} score, however, M_{angle} score for $x_f = -0.16$ is better than $x_f = -0.12$. The correlation is also clear to see from the performance measures. The best scoring foothold selection of $p_x = -0.16$ performs best on all three categories compared to other foothold selections.

Table 4.3: PQ robot recovering from the initial state $z = 0.53$ m, $\dot{x} = 0.0$ m/s and coming to a stop. The bold numbers represent the selected footholds and best performances at that performance measure.

x_f	-0.12	-0.08	-0.04	0.00	0.04
M_{in}	44	**50**	42	38	36
M_{angle}	-	-	-	-	-
t_s	Fall	**0.28**	0.31	0.46	Fall
t_m	Fall	**0.19**	0.27	0.46	Fall
E_m	Fall	**31.42**	33.68	45.90	Fall

The interesting experiment that shows the importance of the foothold selection in action is dropping the robot from the initial height of 0.53 m with no horizontal velocity, $\dot{x} = 0$ m/s. Instinctively, because the robot starts with no horizontal velocity, we expect the robot to put its feet directly below its hips. However, because of the leg geometry of the PQ model, the vertical force capability of the foot position $x_f = 0.0\,m$ is lower than some of the foot positions that are behind the hips. The results of the experiment can be seen in Table 4.3. Only the $x_f = 0.8$ has all of the reference forces inside of each FFS of the FFS series for that foothold and it outperforms other foothold selections on every measure. The difference of FFSs between the foot positions 0.0 and -0.08 in the body coordinates with the first reference force for the initial condition $z = 0.53$ m and $\dot{x} = 0$ m/s can be seen in Figure 4.2. The difference of velocity errors for both foothold selections can be seen in Figure 4.3

4.2.2 Comparison with Raibert's Heuristic

In this section, we compare our method with the Raibert's heuristic [2, 25] for foothold selection. Majority of the state of the art quadruped robots still use Raibert's heuristic

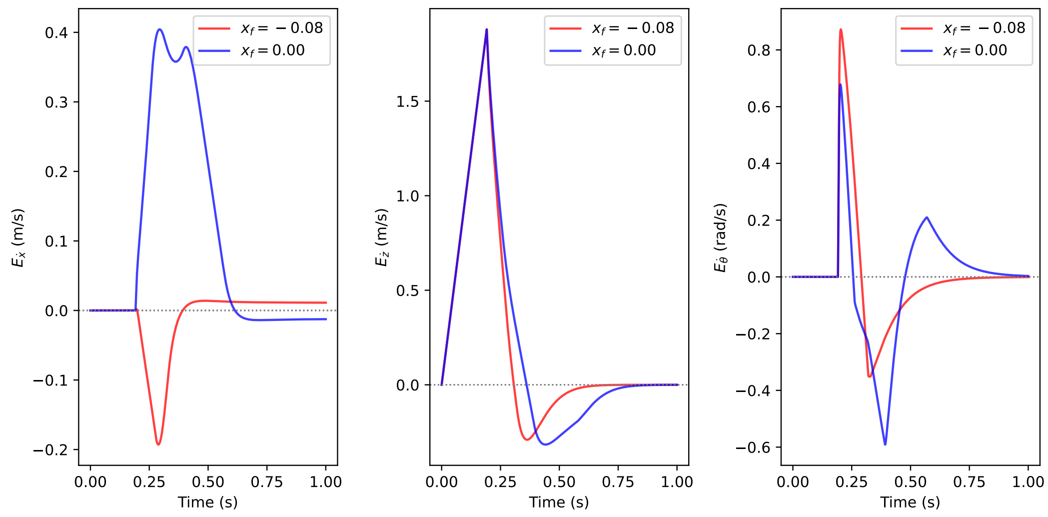


Figure 4.3: Velocity errors for two different foothold selections. Initial state is $z = 0.53$ m and $\dot{x} = 0.0$ m/s for both runs.

with small modifications for choosing their footsteps. Using Raibert's heuristic, the foothold is determined by the following equation:

$$x_f = p_{\text{hip},x} + p_{\text{symmetry}} + p_{\text{velocity}}, \tag{4.5}$$

where $p_{\text{hip},x}$ is the respecting hip position in the x direction. p_{symmetry} is found as

$$p_{\text{symmetry}} = \frac{\dot{x}T_s}{2}, \tag{4.6}$$

where T_s is the expected stance time and $\dot{x}$ is the horizontal velocity of the body. p_{velocity} is

$$p_{\text{velocity}} = k_{\dot{x}}(\dot{x} - \dot{x}_d), \tag{4.7}$$

where $\dot{x}_d$ is the desired velocity. $k_{\dot{x}}$ is the feedback gain for introducing asymmetry from the symmetry point to accelerate or decelerate the body. We optimized the T_s and $k_{\dot{x}}$ values to the best of our ability for our PQ robot and choose stance time $T_s = 0.2$ and $k_{\dot{x}} = 0.03$. To prevent trying to reach footholds that are out of the reach for our PQ robot, we clip the target footholds to be in between $x_f = [-0.24, 0.16]$ m.

We compare our method and Raibert's heuristic for foothold selection in a broad set of initial conditions, precisely, horizontal velocities in range $\dot{x} \in [-2, 2]$ m/s and initial heights $z \in \{0.40, 0.50\}$ m. These initial conditions are selected because they are on the limits of what the PQ stance controller can stop, given that only two legs are used. As we observe from our experiments in the simulation environment, the PQ robot falls regardless of the foothold selection if the initial height or horizontal speed is increased further. Therefore, the following experiments exhaustively evaluate our method against Raibert's heuristic for all possible initial conditions that our PQ stance controller can handle.

First, we compare two methods on the initial conditions of horizontal velocities of $\dot{x} \geq 0$ and the height of 0.4 m. The results of the experiment can be seen in Table 4.4. Overall, for this height and different velocities across the range, both methods perform very similarly but Raibert's method have a slight edge on several occasions. This initial conditions seem to be the best case scenario for our well-tuned Raibert's heuristic implementation.

Then, we compare the two methods on the initial conditions of horizontal velocities $\dot{x} < 0$ and the height of 0.4 m. The results of the experiment can be seen in Table 4.5. For our method, we see a clear trend on choosing footholds that are much more behind the hip's location for stopping a backward initial motion. The main reason behind this behavior is our method pushes to find FFS series that are able to produce the desired reference forces without getting close to the boundaries of an FFS. This is also visible in the performance measure t_m, the motors spent much less time, sometimes never, on their torque limits compared to Raibert's heuristic. While the positions selected by Raibert's heuristic is mirrored for forward and backward motions, our method chooses different footholds for them because it includes the leg geometry in its computations. This leads to better performance overall on all negative horizontal initial velocities.

For the starting initial height of $z = 0.5$ m, we did the same experiments with the horizontal velocities $\dot{x} \in [-2, 2]$ m/s. Because the robot starts from a higher position, the PQ robot needs to be able to produce more force in the z direction in order to

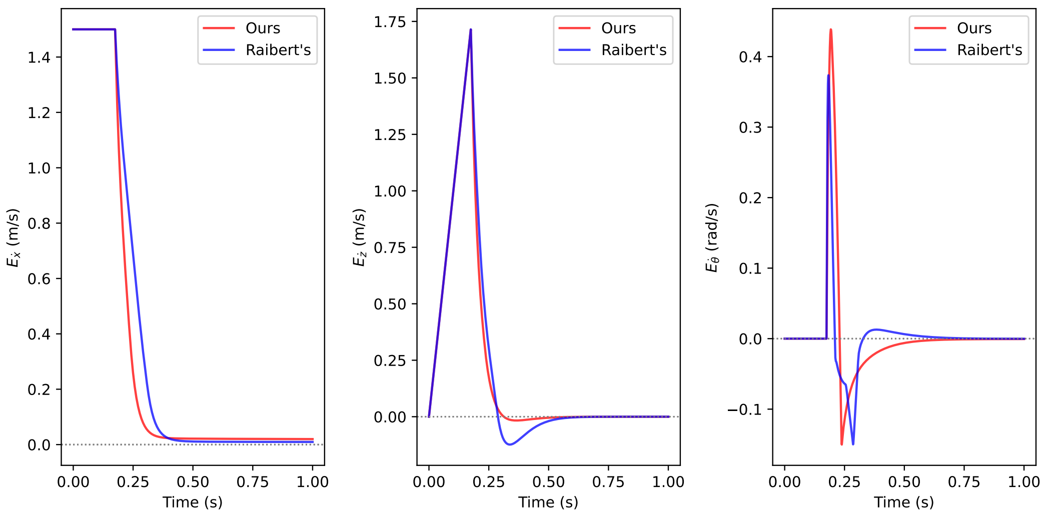


Figure 4.4: Velocity errors for our method and Raibert's heuristic for foothold selection. Initial state is $z = 0.5$ m and $\dot{x} = -1.5$ m/s for both runs.

come to a stop. Our method decides on the footholds including this new requirement and compared with the experiments done on initial height $z = 0.40$ m, it shifts the footholds a little behind to cope with this new height.

The results of the experiments for the initial height of $z = 0.5$ m and the horizontal velocities $\dot{x} \geq 0$ m/s can be seen in Table 4.6. Beginning from the experiment of dropping the PQ robot with zero horizontal velocity, our method outperforms Raibert's heuristic on every performance measure. Especially in lower speeds, our method can select footholds that result in lower energy consumption and the motors of stance legs spend less time on their limits. While the results of Raibert's heuristic are on par with our method on height $z = 0.40$ m, the difference of performance between our method and Raibert's heuristic in these experiments on the higher initial height of 0.50 m is mainly because of the higher touchdown velocity which is taken into account in our method's computations to find a suitable foothold.

The results of the experiments for the initial height of $z = 0.5$ m and the horizontal velocities $\dot{x} < 0$ m/s can be seen in Table 4.7. Similar to starting with initial forward velocities with initial height 0.5 m, our method performs better than Raibert's. Difference between our method and Raibert's heuristic on the velocity errors for the same initial condition of $z = 0.5$ m and $\dot{x} = -1.5$ m/s can be seen in Figure 4.4.

Table 4.4: Comparison of our method and Raibert's heuristic for foothold selection in non-negative horizontal velocities $\dot{x} \geq 0$ m/s. The initial height is 0.4 m for every experiment. O and R stand for "Our Method" and "Raibert's Heuristic" respectively. For each initial condition, the selected foothold is also listed as x_f. The bold numbers represent the best performances at that performance measure.

$\dot{x}$	0.00		0.25		0.50		0.75		1.00		1.25		1.50		1.75		2.00	
Method	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R
x_f	-0.08	0.00	0.04	0.03	0.04	0.07	0.08	0.10	0.12	0.13	0.12	0.16	0.12	0.16	0.16	0.16	0.16	0.16
t_s	**0.12**	**0.12**	**0.12**	**0.12**	**0.12**	**0.12**	**0.12**	**0.12**	**0.13**	**0.13**	**0.14**	0.15	**0.19**	**0.19**	**0.25**	**0.25**	**0.31**	**0.31**
t_m	**0.00**	0.03	**0.02**	**0.02**	0.03	**0.02**	**0.03**	**0.03**	**0.03**	**0.03**	0.05	**0.04**	0.08	**0.05**	**0.06**	**0.06**	**0.11**	**0.11**
E_m	**9.71**	9.75	10.15	**10.05**	**10.85**	11.04	**12.44**	12.61	**14.56**	14.82	**16.94**	17.32	20.56	**20.48**	**24.58**	**24.58**	**29.45**	**29.45**

Table 4.5: Comparison of our method and Raibert's heuristic for foothold selection in negative horizontal velocities $\dot{x} < 0$ m/s. The initial height is 0.4 m for every experiment. O and R stand for "Our Method" and "Raibert's Heuristic" respectively. For each initial condition, the selected foothold is also listed as x_f. The bold numbers represent the best performances at that performance measure.

$\dot{x}$	-2.00		-1.75		-1.50		-1.25		-1.00		-0.75		-0.50		-0.25	
Method	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R
x_f	-0.24	-0.24	-0.24	-0.23	-0.24	-0.20	-0.24	-0.16	-0.24	-0.13	-0.24	-0.10	-0.20	-0.07	-0.16	-0.03
t_s	**0.34**	**0.34**	**0.29**	**0.29**	**0.23**	0.24	0.18	**0.16**	0.15	**0.14**	0.14	**0.13**	**0.12**	**0.12**	0.12	**0.11**
t_m	**0.05**	**0.05**	**0.00**	0.08	**0.00**	0.16	**0.00**	0.15	**0.02**	0.14	**0.06**	0.13	**0.06**	0.11	**0.00**	0.08
E_m	**29.90**	**29.90**	**24.97**	25.41	**20.94**	22.41	**17.39**	19.24	**14.69**	16.86	**12.83**	15.17	**11.16**	13.00	**10.18**	11.20

Table 4.6: Comparison of our method and Raibert's heuristic for foothold selection in non-negative horizontal velocities $\dot{x} \geq 0$ m/s. The initial height is 0.5 m for every experiment. O and R stand for "Our Method" and "Raibert's Heuristic" respectively. For each initial condition, the selected foothold is also listed as x_f. The bold numbers represent the best performances at that performance measure.

$\dot{x}$	0.00		0.25		0.50		0.75		1.00		1.25		1.50		1.75		2.00	
Method	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R
x_f	-0.08	0.00	0.00	0.03	0.00	0.07	0.04	0.10	0.08	0.13	0.12	0.16	0.12	0.16	0.12	0.16	0.12	0.16
t_s	**0.23**	0.33	**0.28**	0.35	**0.26**	0.38	**0.28**	0.41	**0.30**	0.42	**0.36**	0.40	**0.39**	0.41	**0.41**	0.44	**0.46**	**0.46**
t_m	**0.15**	0.33	**0.24**	0.35	**0.19**	0.39	**0.22**	0.42	**0.26**	0.43	**0.27**	0.38	**0.23**	0.33	**0.22**	0.25	0.26	**0.25**
E_m	**24.02**	31.88	**28.65**	34.40	**29.13**	37.85	**32.95**	41.54	**37.96**	44.81	**43.86**	46.64	**47.64**	49.13	**50.56**	53.84	**54.49**	57.56

Table 4.7: Comparison of our method and Raibert's heuristic for foothold selection in negative horizontal velocities $\dot{x} < 0$ m/s. The initial height is 0.5 m for every experiment. O and R stand for "Our Method" and "Raibert's Heuristic" respectively. For each initial condition, the selected foothold is also listed as x_f. The bold numbers represent the best performances at that performance measure.

$\dot{x}$	-2.00		-1.75		-1.50		-1.25		-1.00		-0.75		-0.50		-0.25	
Method	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R
x_f	-0.24	-0.24	-0.24	-0.23	-0.24	-0.20	-0.24	-0.16	-0.24	-0.13	-0.20	-0.10	-0.16	-0.07	-0.12	-0.03
t_s	**0.17**	**0.17**	**0.14**	0.15	**0.11**	0.20	**0.10**	0.25	**0.10**	0.27	**0.09**	0.29	**0.09**	0.30	**0.22**	0.32
t_m	**0.18**	**0.18**	**0.10**	0.17	**0.05**	0.21	**0.06**	0.24	**0.07**	0.25	**0.10**	0.27	**0.12**	0.29	**0.15**	0.31
E_m	**40.08**	**40.08**	**35.60**	35.79	**31.43**	33.33	**28.00**	31.58	**26.15**	30.27	**24.12**	29.57	**22.95**	29.58	**24.02**	30.33

CHAPTER 5

CONCLUSION

This thesis introduced a new foothold selection method for quadruped robots based on feasible force sets. Foot locations shape the forces and the moment that can be produced on the body by each leg during stance, and our method selects footholds considering the constraints of friction cone, motor limits, and leg geometry. In doing so, our method finds footholds that can assist the stance controllers in performing the desired motion more effectively and make them more robust against unexpected disturbances. To find better footholds, we proposed the notion of the FFS series, which captures the force capabilities of a leg during its sweeping motion in the stance phase for a foothold selection. Our method finds a crude approximation of the desired forces that need to be applied to the PQ body for the desired behavior using a reference model and a controller. Lastly, we proposed a two-stage selection method to find better footholds by comparing each possible foot location's FFS series with the desired reference forces.

We implemented a simulation environment with the PQ robot for testing and evaluating our method. In order to evaluate our framework, we proposed three different performance measures:

- time it takes to stabilize the robot's body to a goal state,
- total amount of time each motor spends on their torque limits,
- the amount of energy spent to stabilize the body.

We evaluated our method on the scenario of coming to a stop from an initial condition using only two legs. First, we measured the performance of our foothold selection method by comparing the selected footholds by our method with all possible footholds and showed that our method performs better on a vast majority of initial conditions for all three performance measures. Second, we compared our method with the widely used Raibert's heuristic for foothold selection. The initial results we obtain on our PQ model with the proposed performance measures show that our method can be a promising alternative for Raibert's heuristic for foothold selection.

5.1 Limitations and Future Work

This thesis introduces a novel way to select footholds based on feasible force sets to help the stance phase controllers perform more robust and efficient. However, this work also contains several simplifying assumptions with their corresponding limitations. This section suggests different research directions to help overcome these limitations and new future work for our method.

In our two-stage selection method, we only focused on the forces produced on the body by each stance leg to guide the foothold selection process. However, as we briefly discussed in Chapter 3, each leg also produces a moment and the net moment capability of the body is a crucial part of locomotion. We believe further work and analysis on how to include this net moment capability of each leg on the body for foothold selection is a natural follow-up for this work. Including the moment capabilities of the body in our foothold selection method can negate unwanted angular rotation of the body more effectively.

In this work, we focused on the sagittal dynamics of the quadrupeds and used a planar model of a quadrupedal robot. We believe continuing this work on full 3D model legged systems is a natural step going forward. While the higher dimensional system may not even be feasible with our approach, we can use our framework for foothold selection in the sagittal and lateral planes separately and combine their outputs. Going forward, we can continue evaluating our framework on more realistic 3D simulations and eventually on a real quadruped platform.

In our evaluation, we only focused on the scenario of coming to a stop from an initial condition. However, it is possible to easily extend our method to work for different behaviors by using different templates (reference models and controllers) for the desired behaviors. This way, we can test and evaluate our framework for scenarios involving running with different gaits, such as trotting, pronking, and bounding.

In this thesis, we selected a certain number of footholds to consider and computed the FFS series offline for them. Also, in all of these FFS series, the foot's starting and final relative heights are the same. An interesting direction can be thoroughly analyzing what different initial and final points for the FFS series can be used and step sizes we can take to discretize the foot locations. While we take 0.04 m as the step size in the horizontal direction, a thorough analysis of computational feasibility and its effects on performance can result in more effective ways to sample footholds for start and end foot positions. Another exciting direction can be framing our problem as an optimization one. This way, it may be possible to use continuous footstep locations and forgo the offline phase altogether.

Lastly, another assumption in our work is that we do not consider if the PQ model can actually follow the reference forces. The FFS series is a crude approximation of the robot's force output capability in the stance phase. Because we do not sample and compare the reference forces and FFS series based on time, we do not consider the durations of forces each leg can produce. This is especially important when the robot is moving very fast, more than 2 m/s. While the FFS series with our two-stage selection method can state that it is possible to output the reference forces, in reality, it is possible that the robot cannot produce enough force in a short amount of time to stop the robot because it may move too fast during stance. An improved foothold selection method, including the total force output capability for a given time, can improve the method's performance, especially at high speeds. However, only looking at the impulse capabilities can also be misleading because it is important to produce the reference forces in a particular order.

REFERENCES

[1] M. LaBarbera, "Why the wheels won't go," *The American Naturalist*, vol. 121, no. 3, pp. 395–408, 1983.

[2] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[3] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of experimental biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[4] U. Saranli, M. Buehler, and D. E. Koditschek, "Rhex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.

[5] U. Saranli and D. E. Koditschek, "Template based control of hexapedal running," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 1, pp. 1374–1379, IEEE, 2003.

[6] M. M. Ankaralı and U. Saranlı, "Control of underactuated planar pronking through an embedded spring-mass hopper template," *Autonomous Robots*, vol. 30, no. 2, pp. 217–231, 2011.

[7] H. Geyer, U. Saranli, A. Goswami, and P. Vadakkepat, "Gait based on the spring-loaded inverted pendulum," in *Humanoid Robotics: A Reference*, pp. 1–25, 2017.

[8] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[9] J. A. Grimes and J. W. Hurst, "The design of atrias 1.0 a unique monopod, hopping robot," in *Adaptive Mobile Robotics*, pp. 548–554, World Scientific, 2012.

[10] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate, and J. Hurst, "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, 2016.

[11] W. C. Martin, A. Wu, and H. Geyer, "Experimental evaluation of deadbeat running on the atrias biped," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1085–1092, 2017.

[12] İ. Uyanık, Ö. Morgül, and U. Saranli, "Experimental validation of a feed-forward predictor for the spring-loaded inverted pendulum template," *IEEE Transactions on robotics*, vol. 31, no. 1, pp. 208–216, 2015.

[13] B. Dadashzadeh, H. R. Vejdani, and J. Hurst, "From template to anchor: A novel control strategy for spring-mass running of bipedal robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2566–2571, IEEE, 2014.

[14] M. Rutschmann, B. Satzinger, M. Byl, and K. Byl, "Nonlinear model predictive control for rough-terrain robot hopping," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1859–1864, IEEE, 2012.

[15] Ö. Arslan and U. Saranli, "Reactive planning and control of planar spring–mass running on rough terrain," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 567–579, 2011.

[16] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10822–10825, 2008.

[17] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq–a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

[18] C. Semini, V. Barasuol, M. Focchi, C. Boelens, M. Emara, S. Casella, O. Villarreal, R. Orsolino, G. Fink, S. Fahmi, *et al.*, "Brief introduction to the quadruped robot hyqreal," *Istituto di Robotica e Macchine Intelligenti (I-RIM)*, 2019.

[19] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart, "Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion," in *Adaptive Mobile Robotics*, pp. 483–490, World Scientific, 2012.

[20] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 38–44, IEEE, 2016.

[21] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, "Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3307–3312, IEEE, 2013.

[22] H.-W. Park, P. M. Wensing, and S. Kim, "High-speed bounding with the mit cheetah 2: Control design and experiments," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 167–192, 2017.

[23] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. 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)*, pp. 2245–2252, IEEE, 2018.

[24] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6295–6301, IEEE, 2019.

[25] J. K. Hodgins and M. H. Raibert, "Adjusting step length for rough terrain locomotion," *Dynamically Stable Legged Locomotion*, p. 27, 1991.

[26] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[27] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conference on Robot Learning*, pp. 773–783, PMLR, 2022.

[28] Z. Xie, X. Da, B. Babich, A. Garg, and M. van de Panne, “Glide: Generalizable quadrupedal locomotion in diverse environments with a centroidal model,” *arXiv preprint arXiv:2104.09771*, 2021.

[29] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. Kim, and P. Agrawal, “Learning to jump from pixels,” *arXiv preprint arXiv:2110.15344*, 2021.

[30] J. Pippine, D. Hackett, and A. Watson, “An overview of the defense advanced research projects agency’s learning locomotion program,” *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 141–144, 2011.

[31] M. P. Murphy, A. Saunders, C. Moreira, A. A. Rizzi, and M. Raibert, “The littledog robot,” *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 145–149, 2011.

[32] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, “Fast, robust quadruped locomotion over challenging terrain,” in *2010 IEEE International Conference on Robotics and Automation*, pp. 2665–2670, IEEE, 2010.

[33] M. Zucker, J. A. Bagnell, C. G. Atkeson, and J. Kuffner, “An optimization approach to rough terrain locomotion,” in *2010 IEEE International Conference on Robotics and Automation*, pp. 3589–3595, IEEE, 2010.

[34] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, “Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2464–2470, IEEE, 2020.

[35] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini, “Mpc-based controller with terrain insight for dynamic legged locomotion,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2436–2442, IEEE, 2020.

[36] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter, “Perceptive locomotion in rough terrain–online foothold optimization,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5370–5376, 2020.

[37] R. Grandia, A. J. Taylor, A. D. Ames, and M. 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)*, pp. 8352–8358, IEEE, 2021.

[38] O. Melon, R. Orsolino, D. Surovik, M. Geisert, I. Havoutis, and M. Fallon, "Receding-horizon perceptive trajectory optimization for dynamic legged locomotion with learned initialization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9805–9811, IEEE, 2021.

[39] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. G. Caldwell, and C. Semini, "Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3363–3370, 2018.

[40] R. Krug, A. J. Lilienthal, D. Kragic, and Y. Bekiroglu, "Analytic grasp success prediction with tactile feedback," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 165–171, IEEE, 2016.

[41] C. Boussema, M. J. Powell, G. Bledt, A. J. Ijspeert, P. M. Wensing, and S. Kim, "Online gait transitions and disturbance recovery for legged robots via the feasible impulse set," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1611–1618, 2019.

[42] M. Focchi, A. Del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Autonomous Robots*, vol. 41, no. 1, pp. 259–272, 2017.

[43] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019.