Turning Behavior of Running Systems induced by Leg Placement

Aditya Sripada

CMU-RI-TR-21-54

August 19, 2021

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Hartmut Geyer, PhD, *chair*
Aaron Johnson, PhD
Ashwin Khadke

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

Compared to legged robots, animals and humans can perform much faster and larger turns, even when they run at high speeds. Such rapid turns require the body of a runner to reorient dynamically and in synchrony with its redirection during stance. While it is clear that foot placement affects both direction and orientation, the functional relationship between the three is not well understood. Understanding this relationship could lead to more advanced controllers for turning maneuvers in legged robots as well as to deeper insights into the turning behavior of animals including humans. To develop this relationship, we build on the established spring-mass model for running, replacing its point mass with a rigid body and off-center hip joints. Generalizing ideas from the theory of spring-mass running, we develop controllers for the rigid body version of this model that execute stable running at high speed (5m/s) and synchronous turning between -25° and 45° in deadbeat fashion, on terrain with large uncertainties up to 20% of the rest leg length. We also develop an analytical model for turning dynamics of legged running systems using a simplified and empirical model of forces and kinematics, that qualitatively approximates turning behaviors without the need for the full order dynamics. Our work not only provides a holistic analysis of turning behaviors of running systems with integrated rotational dynamics but also presents a terrain agnostic leg placement strategy for these systems to achieve synchronous turning at high speeds.

Acknowledgments

I would first like to thank my advisor Hartmut Geyer for guiding my Master's thesis. He not only taught me everything I know about research but also helped the engineer in me look at research from a scientist's perspective, for which I would be forever grateful. He always made himself available for discussion and offered relevant and useful advice. I would also like to thank Professor Aaron Johnson for his invaluable feedback and for creating a platform for everyone working in the area of legged locomotion to share their research and ideas in the form of the locomotion seminar. I am very grateful to Ashwin Khadke, a great friend and labmate, who was always ready to help me solve frustrating problems in an enjoyable way and motivating me every time I hit a research roadblock. My other labmates Manan Shah, Omar El Refi, Russell Xing, and Saurav Kumar had been excellent sources of discussion, information, and feedback throughout. I am also deeply grateful to my parents Haragopal and Subbalakshmi, who supported me in every way possible and encouraged me to pursue my interests. I would also like to thank all my other family and friends who helped me get through grad school during the COVID-19 pandemic.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Turning is an important aspect of legged locomotion. Animals and humans are adept at performing quick and large turns, even when they run at high speeds. Legged robots do not exhibit such agility while performing turning maneuvers at high running speeds. This is mainly due to the constraint that rapid running turns need to be synchronous i.e. the center body of the running system should reorient dynamically and in synchrony with its redirection during stance. Asynchronous turns are undesirable as the misalignment of the system's body orientation with its movement direction at the end stance will negatively impact the ability of the system to reach a desired target state in the consecutive step significantly.

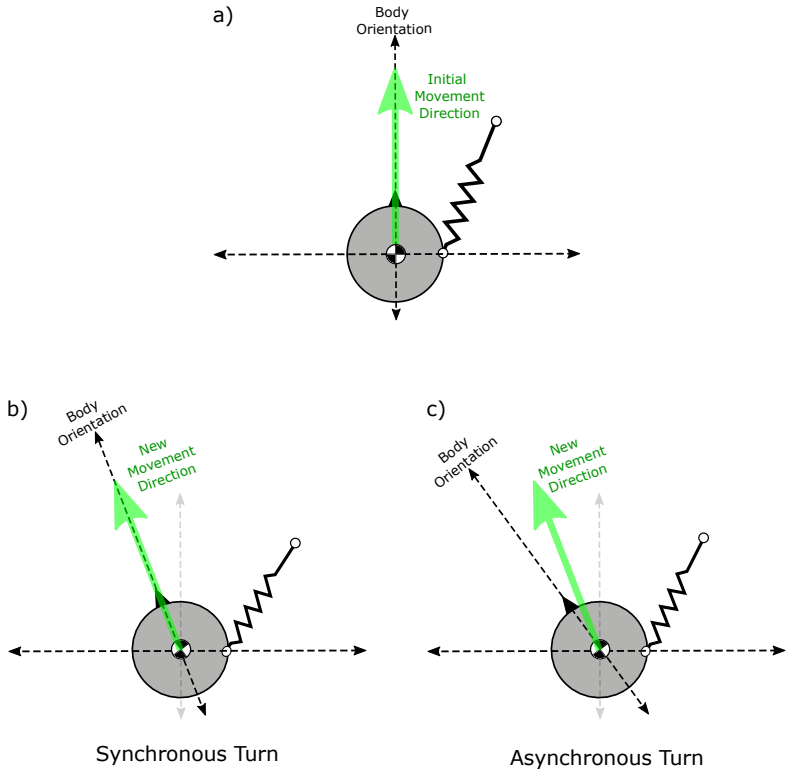


Figure 1.1: Synchronous and asynchronous turns

Fig. 1.1a shows the top view of a legged running system at the beginning of stance phase, with its center body(The black triangle on the periphery of the body represents the face of the body) aligned with its initial movement direction (direction of the Center of Mass(CoM) velocity vector). By the end of the stance phase, the system has turned and changed its course into a new movement direction. The turn is synchronous if the center body's orientation about the roll and pitch axes remains unchanged and its orientation about the yaw axis (the front of its face) is aligned with the new movement direction as depicted in Fig. 1.1b and asynchronous if the center body's face is misaligned with the new movement direction as depicted in Fig. 1.1c.

Past research has established that foot placement affects both turning direction and body orientation [1], [2], but the functional relationship between the three is not well understood. If this is well understood,

1. More advanced controllers for legged locomotion can be developed
2. Additional insights into human and animal turning dynamics can be gained

1.2 Background

Although researchers studied the dynamics of running turns in the context of biological and mechanical legged systems, there is a relative obscurity in the understanding of leg placement's influence on redirection of a bipedal running system's CoM velocity and the reorientation of its center body. This is due to the fact that a majority of the existing literature on the dynamics of running turns

1. is empirical.
2. focuses on multi-legged systems (4 or more legs)

Raibert et al [2] were one of the first to develop heuristics to achieve a net turning motion in a quadruped without changing the heading direction, effectively yawing in place. The magnitude of yawing is based on a heuristic that the torque about the yaw axis is proportional to the angle between the line connecting the hips and the line connecting the feet. More recently in [3], the authors derived heuristics to make a quadruped turn at a rate of 2 rad/s while running at a speed of 1.5 m/s by analyzing simulation data. While these heuristics work well for quadrupeds running at low speeds, whether or not they can be transferred to a bipedal system running at

higher speeds remains unknown.

In [4], the author developed heuristic control algorithms for leg placement-based running and turning for a biped by analysing human data and tested them in simulation for running speeds between 2.5 and 5 m/s. The leg placement positions were expressed as functions of the stance time and the difference in the current velocity and the desired velocity of the runner in x and y directions. Perkins & Waldron [5] formulated heuristic control laws that place the outer leg closer to the vertical, and the inner leg farther in front of the hip at touchdown at the beginning of the stance phase of a bipedal runner, running at 3 m/s. The sharpest turn that could be achieved was at 45^o. Neither of these controllers accounts for the angular momentum induced in the center body while running and assume that the mass is concentrated as a point mass. While optimally tuned heuristics work well for the systems they have designed for, they may not generalize well to other systems.

Alongside heuristics, legged locomotion researchers used simplified models for running to study legged turning dynamics and formulate leg placement based controllers. One such model is an inverted pendulum model(IPM).

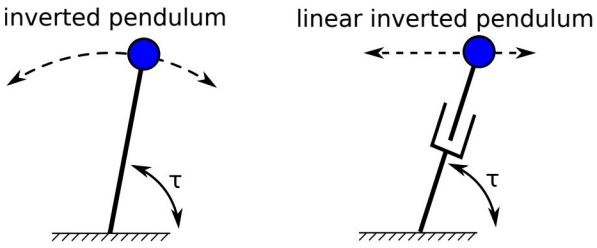
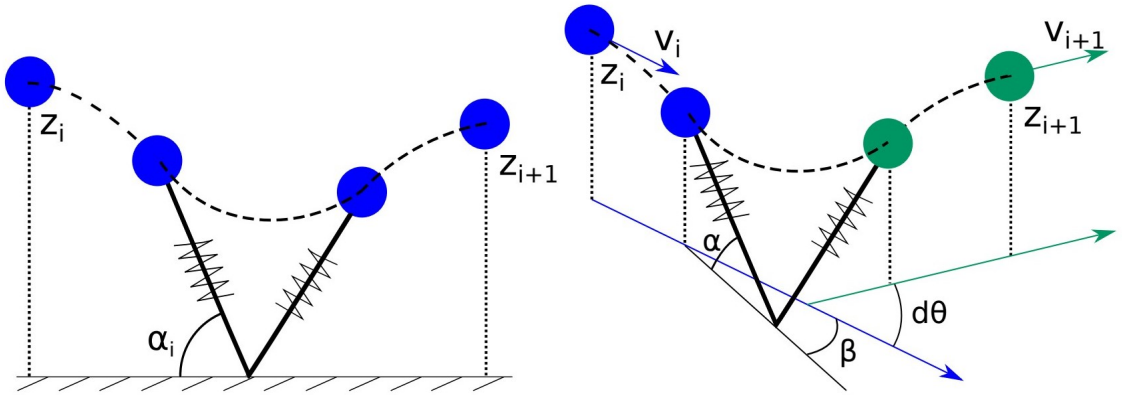


Figure 1.2: The inverted pendulum model(IPM) and the linear inverted pendulum model(LIPM)

In the IPM, the running system is modeled as a point mass connected to the top

of a rigid leg of constant length which is pivoted on tho the ground on the other end through stance(Fig.1.2). One of the first approaches that use the IPM to stabilize gait through foot placement defines foot placement as a linear function of the position and velocity of the system's CoM at touchdown [6]. Other researchers have also used the IPM to study legged systems and enhance stability through one or a sequence of leg placements [7, 8, 9].



(a) SMM in 2D (b) SMM in 3D

Figure 1.3: The 2D 3D Spring Mass Models

Another simplified gait model that was used to study running dynamics by many researchers was the 2D spring mass model(2D SMM), where the system is modeled as a point mass bounding on a mass-less spring leg in the sagittal plane(Fig. 1.3). The SMM is a better model for studying running dynamics as it mimics running ground reaction forces of humans more closely than the IPM[10]. The 3D SMM, an extension to the 2D SMM for motion in three dimensions was also developed and used in the past to study as well as develop controllers for turning[1, 11, 12, 13, 14]. A popular method among the research community was using the SMM to construct a single step

return map and analyze it to find leg placements in the form of touchdown angles to achieve locally stable gaits in 2D [12, 15, 16] as well as 3D [1, 17] in theory and simulation.

This thesis draws some of its inspiration from the work on deadbeat control of 3D SMM running and steering by Wu & Geyer [1]. The authors extend the available deadbeat control ideas for 2D running systems to a 3D running system by adding an additional control input in the form of the angle of splay, alongside the existing angle of attack based control from 2D SMM. By varying the angle of attack and splay as inputs, the authors build a time-based deadbeat controller using which the runner can be made to track a target apex height to maintain stability while changing its running direction based on the desired target requirement. They also analyze the system's robustness to change in terrain and achieve stable running and turning till up to 30% of the rest leg length.

Although these simplified models offered valuable insights into the dynamics of legged locomotion and helped researchers and scientists push the field forward, their major limitation is that they are all point mass based models and only account for the linear dynamics of the running systems, completely ignoring the rotational dynamics. So, when controllers built based on these models are transferred to real robots, they require supplementary controllers that regulate the rotational dynamics of the center bodies of the running systems, often times interfering with the original controller and deteriorating its performance. If the influence of the rotational dynamics of the center body on the runner's linear dynamics is accounted for, this problem will be solved, but none of these models have the ability to account for the rotational dynamics of

the system.

Researchers in the past have also used additional appendages to achieve redirection of the system movement as well as to reorient the center body at will during flight [18, 19, 20, 21]. While inertial appendages help perform turning maneuvers rapidly and efficiently, they are not absolutely necessary. Many four-legged animals like hares and gazelles as well as two legged-animals like humans and ostriches can perform rapid turning maneuvers without tails.

Turning dynamics of running systems was studied extensively in the context of biological systems as well. However, a majority of the research was focused on animals with four or more legs [22, 23, 24, 25] as the multi-contact nature of such systems can be exploited to generate turning behaviors more easily. One of the first works that attempt to draw relationships between foot placement, redirection and reorientation of a human runner was the work by Jindrich et al. [26]. The authors relate leg placement in the transversal plane to the ratio of body reorientation and redirection given the running velocity and stance time using an analytical model. It was derived for preset, sinusoidal patterns in the fore-aft and mediolateral leg forces, with the body orientation of the system restricted in the transverse plane, for small redirections. Whether or not these relationships generalize well to larger, more dynamic turns remains unknown.

The simplified legged running models in current literature only use point masses to approximate running dynamics and completely ignore the rotational dynamics of the center body. In order to understand the influence of leg placement on the redirection of the CoM velocity as well as the reorientation of the center body during

stance, a simplified model for legged running that also accounts for the center body rotational dynamics is necessary. So in the next chapter, we propose one such model and use that to study turning dynamics in further chapters.

1.3 Approach

We first develop the rigid-body spring mass model (RBSMM), an extension to the 3D SMM with realistic rotational dynamics. Using the RBSMM, we construct a single step apex return map, which is a discrete mapping of the system states at one apex to those at the consecutive apex of a legged running system to encapsulate the model behavior, similar to [1]. This process is explained in detail in chapter 2.

We then verify the validity of RBSMM for running turns by comparing the results from the return map generated with those computed using an analytical model for running turns, developed purely based on simplified models for forces and kinematics. Various steps involved in the development of the analytical model are described in detail in chapter 3.

Lastly, we draw functional relationships between leg placement and synchronous turning through constrained optimization on the single step apex return map. Alongside understanding how leg placement, redirection and reorientation are interwoven, we also build a controller to achieve stable and synchronous running turns on the RBSMM in simulation. The functional relationships and the controller's performance in simulation are presented in chapter 4.

Chapter 2

The Rigidbody Spring Mass Model

In this chapter, we first introduce the Rigid body spring mass model (RBSMM) and its dynamics. We then describe the process of single step apex return map generation using the RBSMM. Lastly, we compare the model behavior extracted from the return map with the one computed using the analytical expressions from [26] to validate the RBSMM.

2.1 Model Definition

To study the turning dynamics of legged running systems, we build on the 3D Spring Mass Model (3D SMM) based on the previous work by [1], [11], [12], [13], [14] and develop the Rigidbody Spring Mass Model (RBSMM) . In the RBSMM, the point mass of a 3D SMM is replaced with a rigid body of mass m and moments of inertia I_{xx}, I_{yy} and I_{zz} about the principle body axes x_b, y_b, z_b(Fig. 2.1). A mass-less spring

of rest length l_0 and stiffness k is connected to the rigid body through a hip joint, h_xm away from the center of mass in the sagittal axis.

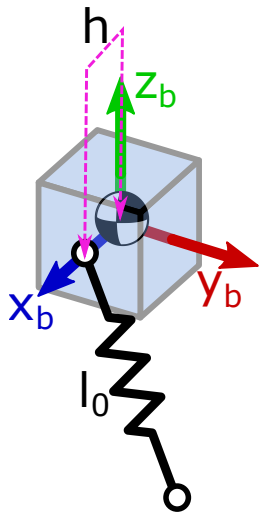


Figure 2.1: The Rigidbody Spring Mass Model(RBSMM)

A "step" of this model is a combination of an initial flight phase that starts at apex i, a stance phase and a secondary flight phase that occurs after the stance phase and ends when the body has reached apex $i+1$. An apex is described as the highest point in the flight, with the y-axis(y_b) aligned with the velocity vector $\dot{y}$. Coordinates with the subscript i were used to denoted the system states and parameters at the initial apex and those with the subscript $i+1$ represent the system states and parameters at the consecutive apex.

2.2 Model Dynamics

2.2.1 Initial flight phase (apex i till touchdown):

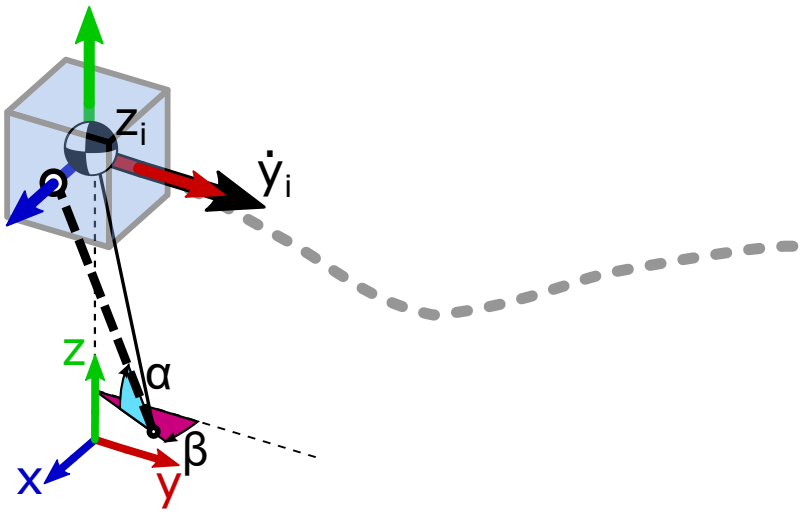


Figure 2.2: RBSMM-Initial apex

The initial flight phase is modeled as a rigid body with mass m and moments of inertia I_{xx}, I_{yy} and I_{zz} about the principle axes which follows a ballistic trajectory with a forward velocity of $\dot{y}_i$ from apex i located at $[0, 0, z_i]$. The body frame of the model(x_b, y_b, z_b) is aligned with the inertial frame(x, y, z) at the apex, whose origin is at the ground level(Fig. 2.2). The initial orientation of the rigid body is described by the quaternion q_0. Quaternions were used to describe the orientation in order to reduce the computation needed to perform expensive integrations that come with rotation matrices while solving system dynamics. Pitch(ϕ), roll(Ω) and

yaw(ψ) are also used to describe the orientation of the center body about $x-y-z$ axes respectively in some sections of this document for ease of use and intuitive understanding.

The linear dynamics of the system can be expressed as

$$m\ddot{\mathbf{r}} = \mathbf{f^{net}} \tag{2.1}$$

where r is the location of the center of mass and $\mathbf{f}_{net}$ is the net force acting on the body. As the motion of the system is ballistic in nature, the only force acting on it is gravitation. Hence,

$$\mathbf{f^{net}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{2.2}$$

After the apex event, the swing leg orientation is parametrized by the angle of attack α and the angle of splay β. The initial flight phase ends when the swing leg makes contact with the ground, characterized by center of mass of the rigid body reaching the touchdown height $z_{TD} = l_0 sin(\alpha)$.

2.2.2 Stance phase (from touchdown till takeoff):

Stance phase can be divided into 3 stages:

1 Touchdown: Touchdown is when swing leg makes contact with the ground at a point FP (foot position) and this point remains fixed through stance (slipping and friction limits are ignored)(Fig. 2.3). The coordinates of the foot position

are defined as

$$\begin{bmatrix} x_{FP} \\ y_{FP} \\ z_{FP} \end{bmatrix} = \begin{bmatrix} h + l_0 \, cos\alpha \, sin\beta \\ l_0 \, cos\alpha \, cos\beta \\ 0 \end{bmatrix} \tag{2.3}$$

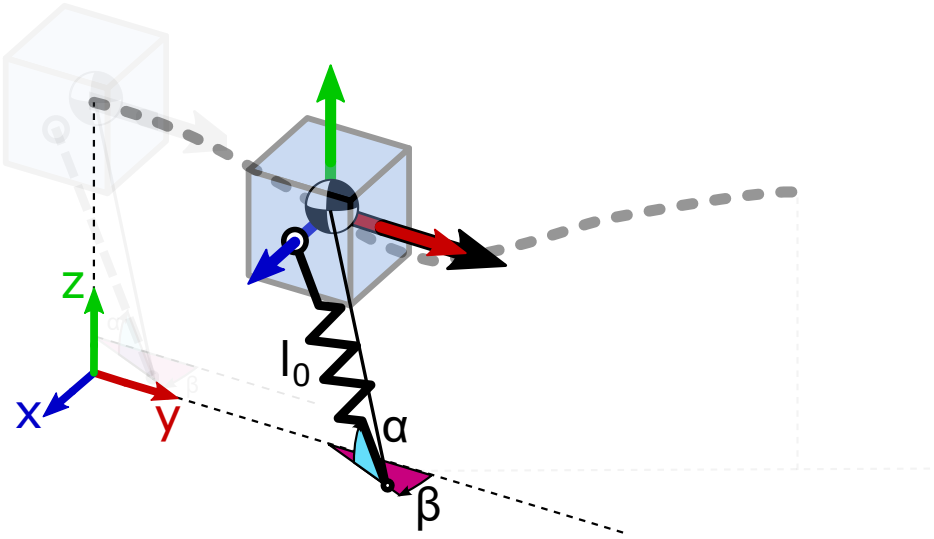


Figure 2.3: RBSMM-Touchdown

2 Mid-stance: The system reaches mid-stance stage when the spring has reached its maximum compression, i.e. the CoM is at its lower most point of stance(Fig. 2.4). After this stage, the spring extends, releasing all the accumulated energy propelling the system into its next flight phase.

3 Takeoff: The system reaches takeoff condition when the spring has reached its rest length(Fig. 2.5). At this point, the spring leg loses contact with the ground.

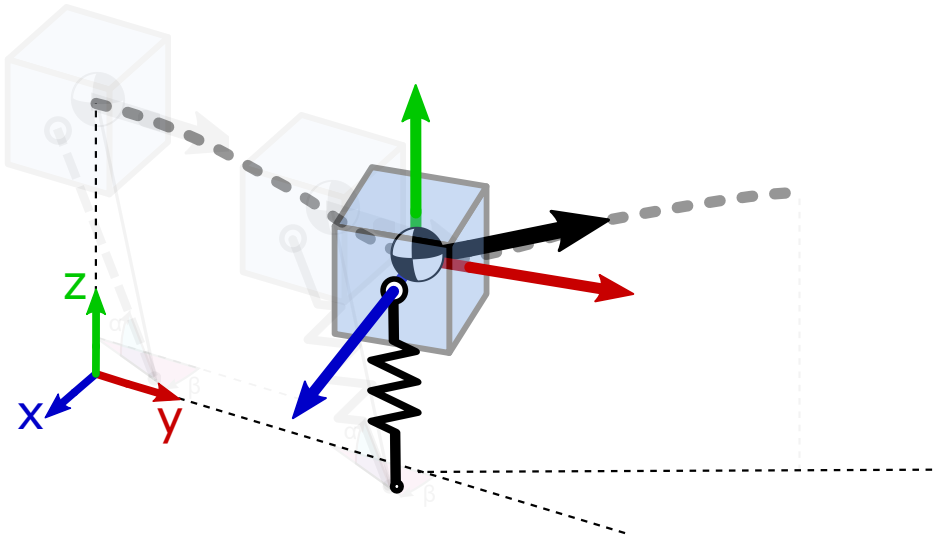


Figure 2.4: RBSMM-Midstance

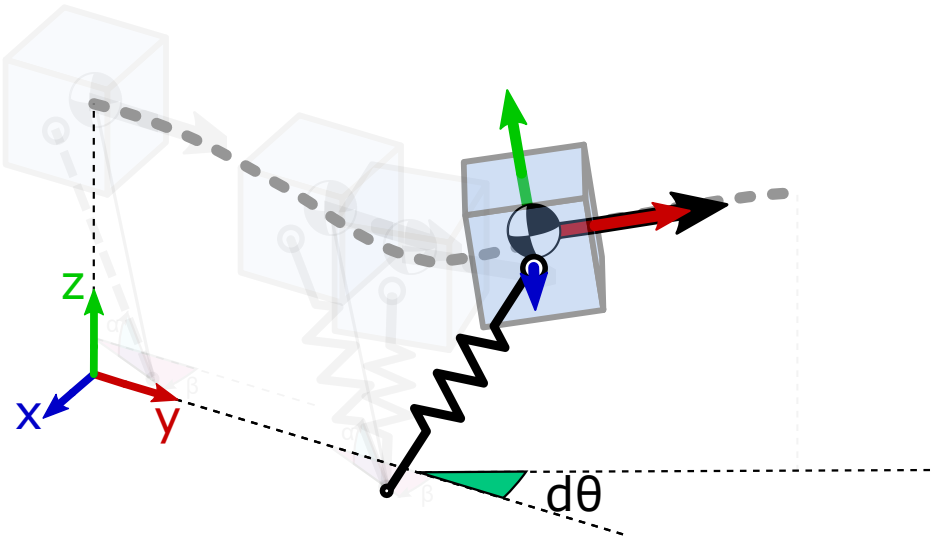


Figure 2.5: RBSMM-Takeoff

Stance Phase Dynamics

Unlike the free-falling flight phase, the rigid body rebounds on a mass-less spring through stance. A wrench F_{net} acts on the body at the hip joint, which affects its rotational and translational dynamics.

The translational dynamics of the rigid body in the stance phase are described by

$$m\ddot{\boldsymbol{r}} = \mathbf{f}^{leg} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{2.4}$$

where

$$\mathbf{f}^{leg} = k(l_0 - l)\frac{\boldsymbol{l}}{l} \tag{2.5}$$

with $l = \|\boldsymbol{l}\|$ being the current leg length.

From $\mathbf{f}^{leg}$, the double integration that relates linear accelerations to velocities and positions of the CoM of the rigid body thus be represented by

$$\dot{\boldsymbol{r}} = \frac{1}{m}\int \left(k(l_0 - l)\frac{\boldsymbol{l}}{l} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \right) dt + \dot{\boldsymbol{r}}_{\mathbf{0}} \tag{2.6}$$

and

$$\boldsymbol{r} = \int \dot{\boldsymbol{r}}\ dt + \boldsymbol{r}_{\mathbf{0}} \tag{2.7}$$

Once $\mathbf{f}_{leg}$ is computed, the net torque acting on the rigid body can be computed

as

$$\boldsymbol{\tau}^{net} = h \times \mathbf{f}^{leg} \tag{2.8}$$

To reduce the complexity of the system, rotational dynamics are easier to track in the body frame. The rate of change of angular velocity of the rigid body ($\dot{\omega_b}$)is expressed as

$$\mathbf{I_b}\dot{\boldsymbol{\omega}}_\mathbf{b} = \tau_b^{net} - \omega_b \times I_b\omega_b \tag{2.9}$$

Here, I_b is the constant inertia matrix of the rigid body and τ_b^{net} is the net torque acting on the body. τ_b^{net} can be computed from τ^{net} using quaternion rotation.

$$\boldsymbol{\tau}_b^{net} = \boldsymbol{q}^* \circ \boldsymbol{\tau}^{net} \circ \boldsymbol{q} \tag{2.10}$$

where $\boldsymbol{q}$ is the quaternion that represents the orientation of the rigid body.

Using the body frame torques computed using Eq. 2.10, the double integration that relates angular accelerations to orientations of the rigid body thus be represented by

$$\boldsymbol{\omega_b} = \int \boldsymbol{I_b}^{-1}\left(\boldsymbol{\tau}_b^{net} - \boldsymbol{\omega}_b \times \boldsymbol{I}_b\boldsymbol{\omega}_b\right) dt + \boldsymbol{\omega}_0^b \tag{2.11}$$

$$\boldsymbol{q} = \int \frac{1}{2}\left(\boldsymbol{q} \circ \boldsymbol{\omega}_b\right) + \boldsymbol{q}_0 \tag{2.12}$$

Thus $\dot{r}$, r, ω_b and q can be computed by integrating over initial states with the dynamics governing every consecutive state by using Eq. 2.6, 2.7, 2.11 and 2.12

respectively.

2.2.3 Secondary flight phase (from takeoff till apex $i+1$):

After the stance phase, the system propels into a flight phase where it takes an upward ballistic trajectory till it reaches another apex, apex $i+1$. The step ends when the system reaches the apex $i+1$.

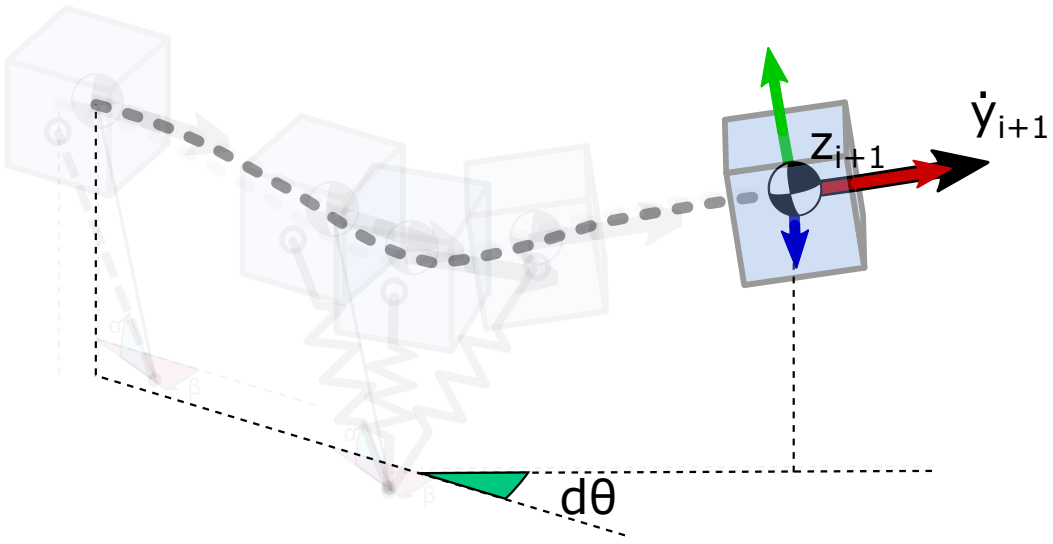


Figure 2.6: RBSMM-Apex 1

2.3 Return Map Generation

2.3.1 Behavior Function

The behavior of the RBSMM is captured by an apex return map. An apex return map is a discrete mapping of the system states between apex i and the consecutive apex $i+1$.

The system states

The system starts with an initial CoM position $\boldsymbol{r}_i = (0, 0, z_i)$ where z_i is the apex height and with a forward velocity of $\dot{\boldsymbol{r}}_{\boldsymbol{i}} = (0, \dot{y}_i, 0)$. The body frame is assumed to be aligned with the inertial frame initially, represented by the quaternion $q_i = (1, 0, 0, 0)$ and has zero angular velocity, i.e. $\boldsymbol{\omega}_{bi} = (0, 0, 0)$. So, the state vector of the system at apex i can be represented as

$$\boldsymbol{s}_i = \left[\boldsymbol{r_i}\ \dot{\boldsymbol{r}}_{\boldsymbol{i}}\ \boldsymbol{q_i}\ \boldsymbol{\omega_{bi}}\right] \tag{2.13}$$

The control inputs

The control problem deals with methods to identify the control inputs that can be varied to drive the model to a desired state from a given set of initial conditions. The parameter vector of the RBSMM can be characterized as

$$\boldsymbol{p}_i = \left[E_s\ m\ I_b\ g\ k\ l_0\ \alpha\ \beta\right] \tag{2.14}$$

Similar to [1], we choose α and β, the two angles that represent the swing leg orientation as the control inputs.

$$\boldsymbol{u} = \begin{bmatrix} \alpha & \beta \end{bmatrix} \tag{2.15}$$

2.3.2 Discrete Representation of the RBSMM behavior

Using the system states s_i and the control inputs u, a single step apex return map R was constructed by scanning over all the possible initial conditions and control inputs. The return map thus created can be represented as

$$\boldsymbol{s}_{i+1} = R(\boldsymbol{s}_i, u) \tag{2.16}$$

As control inputs, α was chosen within the range of $[45^o, 90^o]$ discretized at 0.5^o intervals and beta in the range of $[-90^o, 90^o]$ discretized at 1^o intervals. The initial apex height z_i was considered in the range from $[0.71, 2.27]$m at an interval of $0.005m$, the minimum value being the lowest apex height that can be achieved, given the control inputs and the maximum value being the height at which all the system energy gets converted to potential energy. All the other parameters in the parameter vector $\boldsymbol{p_i}$ other than α were kept constant while the return map was being computed. As the actions of the system are completely passive in nature and other opposing forces like friction and air resistance have been ignored, the total system energy remains constant throughout and hence the system is conservative in nature. Hence, Eq. 2.16

can be simplified to

$$(z,\ \theta,\ q,\ \omega_b)_{i+1} = R(z_i,\ \alpha,\ \beta) \tag{2.17}$$

where $d\theta$ is the angle at which the system redirects into a new movement direction every consecutive step which is computed as

$$d\theta = tan^{-1}\left(\frac{-\dot{x_i}}{\dot{y_i}}\right) \tag{2.18}$$

with $\theta_{i+1} = \theta_i + d\theta$

Using the aforementioned ranges, a return map of resolution 91x181x313 in $\alpha - \beta - z_i$ was generated.

2.3.3 Computational Setup

The apex return map R was computed using MATLAB2019b and Simulink on a MATLAB Cluster with 32 workers where each worker is one core of an Intel i-7 3rd generation processor. The cluster has a memory of 32GB, however only a fraction of it was utilised for the necessary computation.

2.4 Control of the trunk orientation during stance

2.4.1 Why do we need to control the trunk orientation?

One of the primary objectives of this work is to identify control policies that can drive the RBSMM to turn synchronously while maintaining gait stability. From the return map R computed, it is evident that control combinations that drive the RBSMM to reach a desired goal state passively are quite scarce. In Fig. 2.7, there are no leg placement combinations where roll, pitch are both zero, though it is a requirement for synchronous turns.

Turning synchronously while maintaining gait stability requires controlling the CoM height and velocity vector direction and the center body's orientation and angular velocity. However, to control so many parameters with only two control inputs (α and β) makes the system highly underactuated and reduces the number of desired outcomes greatly. Hence, active torques are to be applied to control the trunk during stance in order to drive the system into a desired state (Fig. 2.8).

2.4.2 Control of the centre body

A simple PID controller was used to regulate the roll and pitch of the system to zero. However, another problem springs from this- The rigid body cannot yaw anymore due to the force vector passing through its CoM with the aggressive control of its roll and pitch. This might seem non-intuitive at first, but with infinite torques continuously

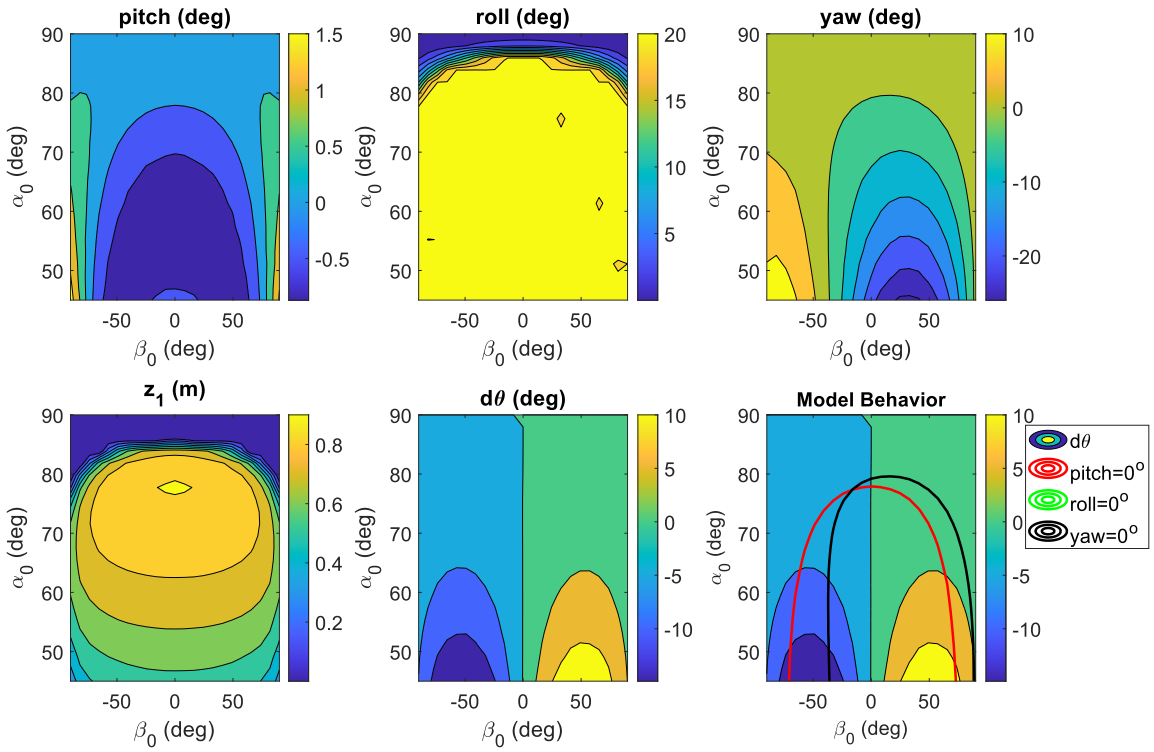


Figure 2.7: Passive Dynamic Behavior of the RBSMM

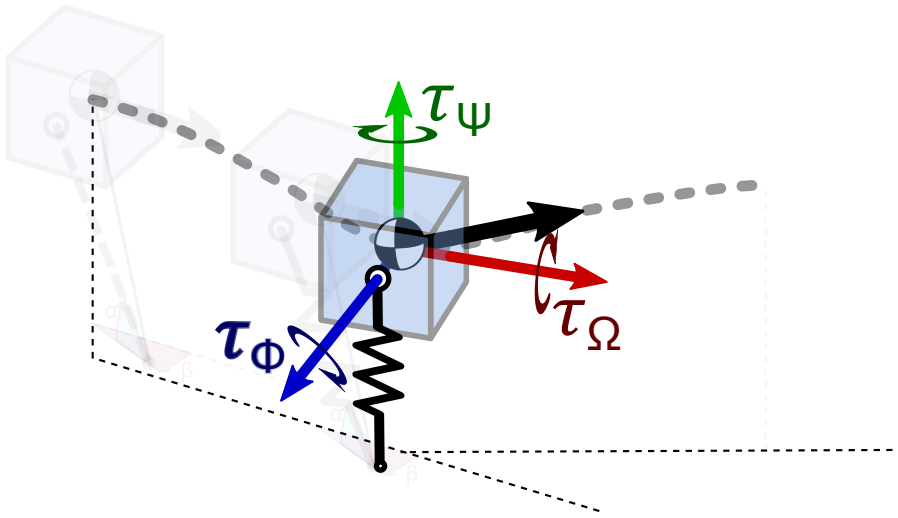


Figure 2.8: Active control to regulate the orientation of the center body

driving the roll and pitch to zero (effectively aligning the transverse plane of the body frame parallel to the transverse plane of the world frame) would cause this. This problem prevents the system from achieving synchronous turning. Hence, additional torque is also required about the z-axis to reorient the centre body to align the body y-axis (y_b) with the new movement direction of the system. So, the target of the PID controller controlling the body orientation is to maintain 0 roll and pitch and yaw same as $d\theta$, the change in the velocity vector direction, i.e. the turning angle. The PID controller used can be represented as

$$\boldsymbol{u_{active}(t)} = \begin{bmatrix} \tau_\phi \\ \tau_\Omega \\ \tau_\psi \end{bmatrix} = \boldsymbol{k_p}.\boldsymbol{e} + \boldsymbol{k_i}\int_0^t \boldsymbol{e}\ dt + \boldsymbol{k_d}\frac{d}{dt}\boldsymbol{e}$$

(2.19)

where k_p, k_i and k_d are proportional, integral and derivative gains respectively and $\boldsymbol{e}$ is the error vector e is defined as

$$\boldsymbol{e} = \begin{pmatrix} 0 - \phi_{err} \\ 0 - \Omega_{err} \\ d\theta - \psi_{err} \end{pmatrix}$$

The torques that were applied to control the centre body's orientation cause reaction forces that affect the leg force profiles, which in turn affect the system dynamics. After incorporating these reaction forces into the model, a single step apex return map was computed again with the same discretization as the one described in the subsection 2.3.2. The resulting return map was analysed to formulate a control policy in chapter-4.

2.5 Results & Discussion

Due to the highly underactuated nature of the RBSMM, it is impossible to achieve redirection in the system CoM velocity and align the orientation of the center body with the new movement direction simultaneously. However, similar to the case of the 3D SMM steering [1], redirection of the CoM velocity can be achieved in a feed-forward manner, solely through leg placement. To align the center body orientation with the

new movement direction, active torques have to be applied to it in a feedback fashion as discussed in the section 2.4.2. This makes all possible leg placements result in synchronous turns. From Fig 2.9a, it can be noticed that all combinations of α and β cause synchronous turns (ratio of the yaw angle ψ and the redirection angle $d\theta$ becomes 1). However, leg placement has a significant influence on the total power required to synchronize the yaw angle ψ and redirection angle $d\theta$. Fig 2.9b depicts the total power required to regulate the rolling and pitching of the body to zero and drive the yaw of the center body to align with the redirection. Evidently, certain leg placements require lesser energy to achieve synchrony than the others.

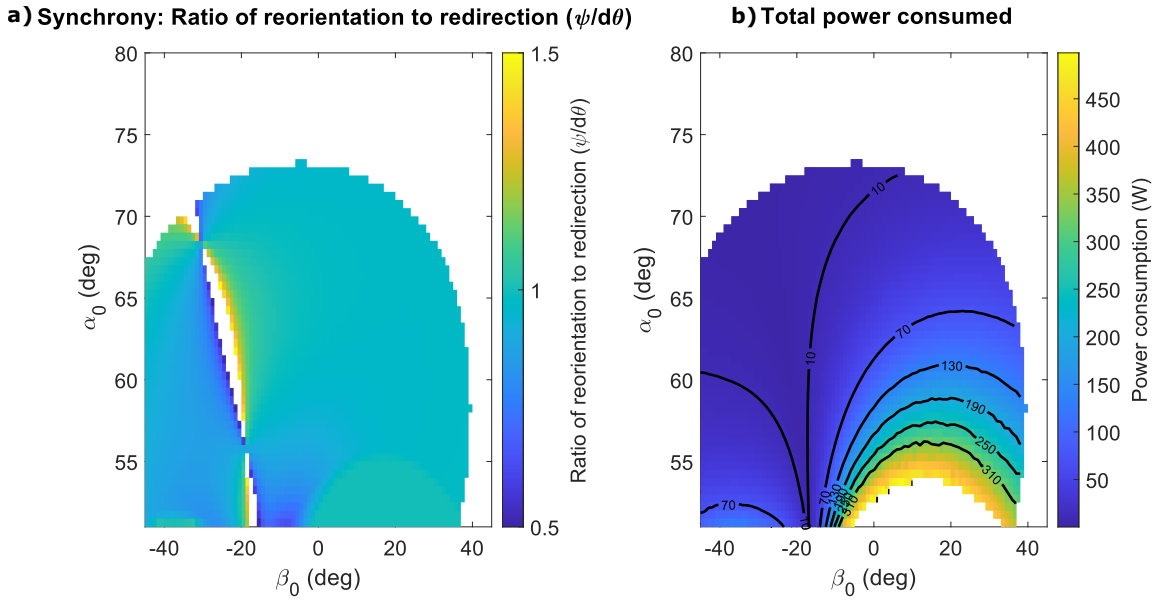


Figure 2.9: Power consumed to achieve synchronicity

From the return map generated using the RBSMM with active torques, we computed the reorientation(yaw motion, ψ) of the center body and the redirection of the CoM velocity($d\theta$). We also computed the values of reorientation and redirection

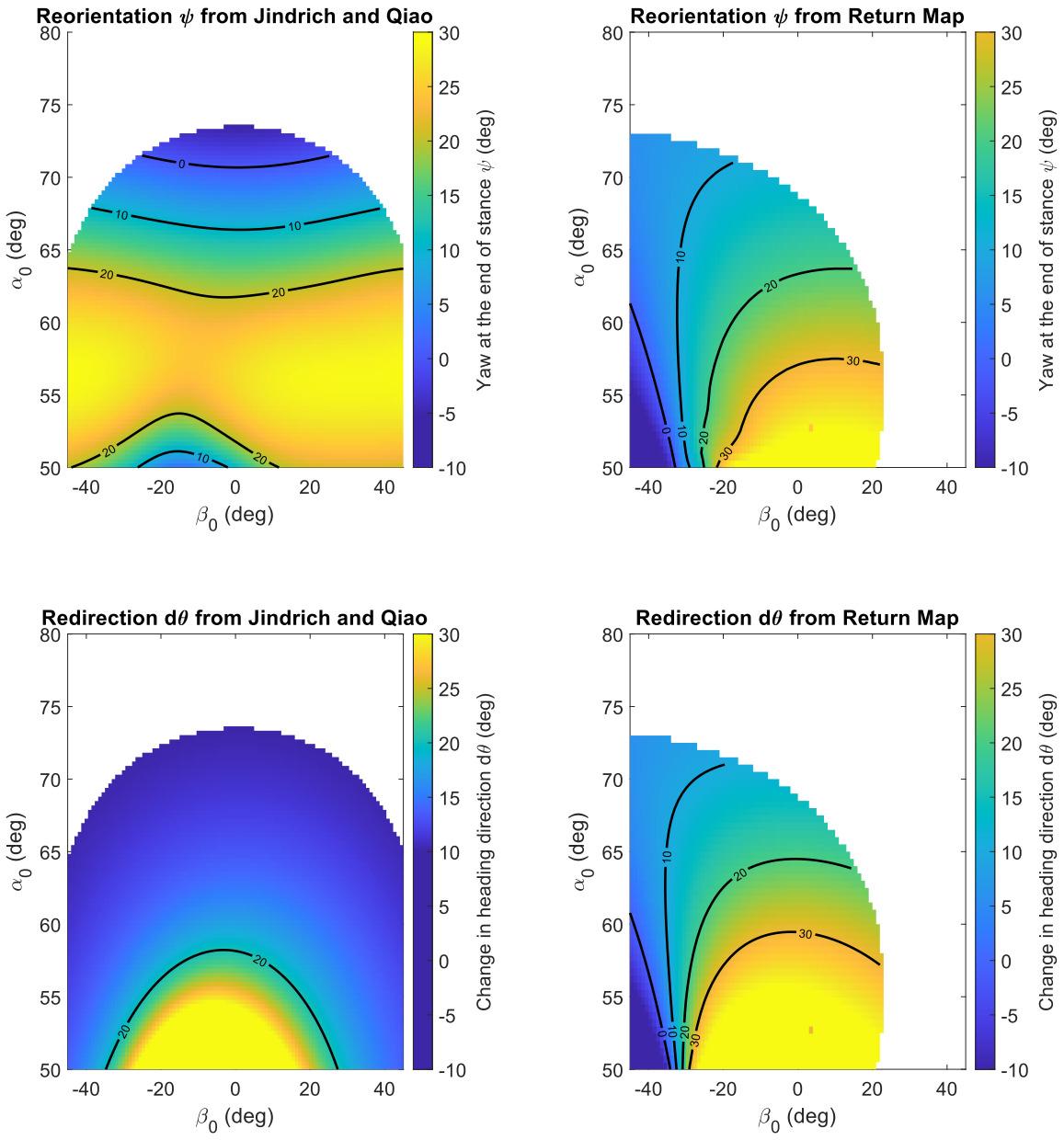


Figure 2.10: Comparison of the reorientation to redirection behaviors predicted by Jindrich et al.[26] and observed in the return map

predicted by the expressions from [26]. Upon close inspection, it is clear that the reorientation ψ and the redirection $d\theta$ behaviors observed from the return map R are quite different from those predicted by Jindrich et al. [26]. To understand the source of this discrepancy, we develop an analytical model with similar assumptions as of the former but with a few modifications. Our analytical model is explained in detail in the next chapter.

Chapter 3

The Analytical Model

To further investigate the disagreement between the turning behavior exhibited by the RBSMM proposed in chapter 2 and the analytical model of planar turning developed by Jindrich et.al[26], we created an analytical turning model with similar simplifying assumptions to those made in [26]. The aim of this analytical model for turning on compliant legs is to:

1. Identify the source of discrepancy, if any, between the turning behaviors observed in R and the analytical model from [26].
2. Develop a simple model that does not require the full order dynamics to approximate turning behaviors of legged running systems
3. Validate the results from the turning behaviors observed from the RBSMM return map

3.1 Problem Formulation

We aim to establish a relationship between the redirection of the CoM velocity vector and the reorientation of the center body similar to [26], and with the leg placement position(FP) as well. So we propose a simplified model where a body of mass m and moment of inertia of I_z about the longitudinal axis, traveling with a forward speed v along the initial movement direction at the beginning of the stance phase as shown in Fig. 3.1a and seeks to change its movement direction by an angle $d\theta$ by placing its leg at a point FP on the ground in a stance time period of t. The ideal scenario would be the one where the center body reorients exactly as much as $d\theta$, making the yaw angle $\psi = d\theta$ and the turn synchronous by the end of the stance phase(Fig. 3.1b).

The leg is assumed to be a mass-less spring of stiffness k and rest length l_0 connected to the body at the hip joint H, hm away from the CoM on the frontal axis(Fig. 3.1b). We make a few more simplifying assumptions as well to reduce the complexity of the model, such as:

1. The motion is strictly restricted to the transverse plane, meaning the center body cannot roll or pitch
2. The center of mass of the system is travelling at a constant speed v throughout the stance phase, in a constant time period t
3. The motion of the leg is linear relative to the CoM throughout the stance phase
4. The fore-aft force F_x is half-sinusoidal spring leg force and the mediolateral force F_y is close to full-sinusoidal spring leg force (Fig. 3.2)
5. The motion is symmetric about mid-stance

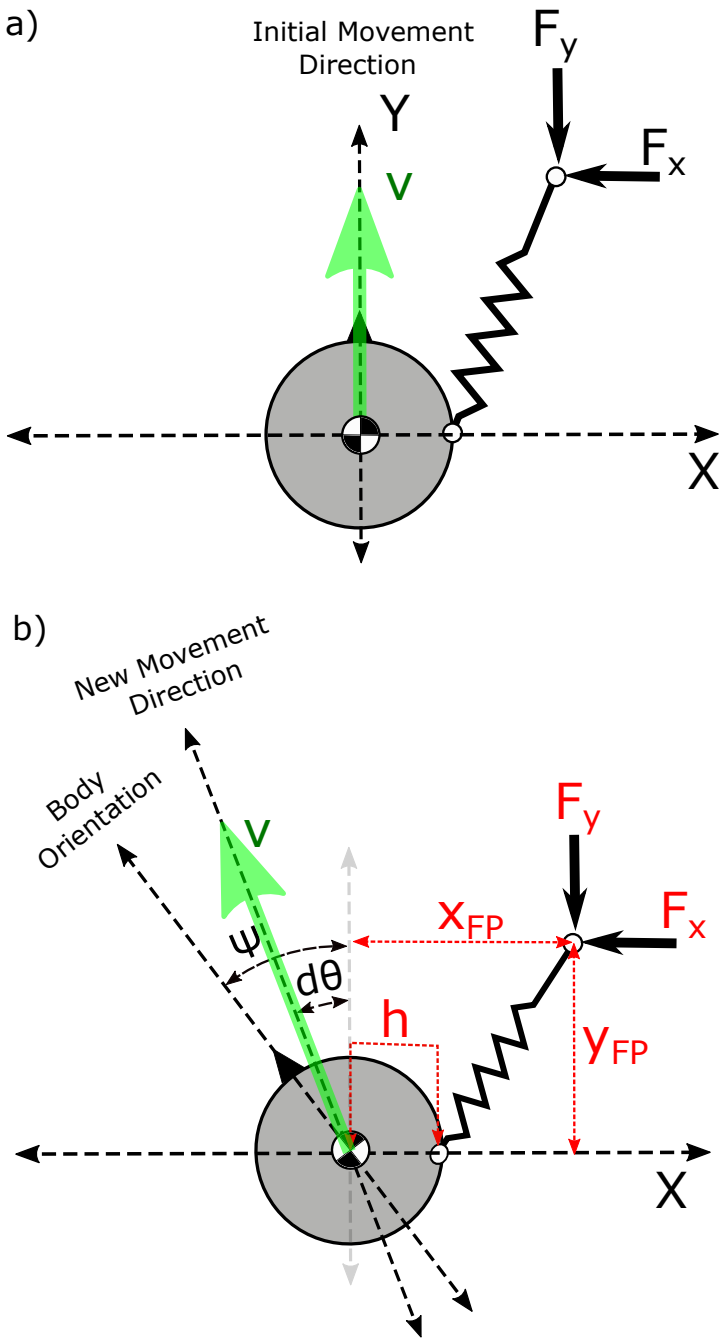


Figure 3.1: Analytical model

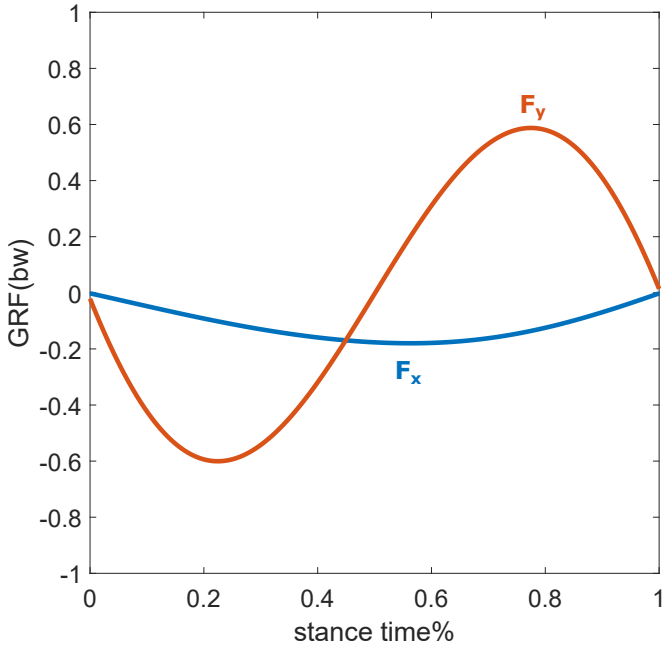


Figure 3.2: Spring leg forces

This model was used to derive expressions for the reorientation of the center body and the redirection of the CoM velocity in the transverse plane as a function of leg placement position and then derive an expression for the ratio of the angle of reorientation to the angle of redirection similar to [26].

3.2 Expression for the reorientation of the center body

The yaw dynamics of the system are given by

$$\tau = I_z \ddot{\psi} \tag{3.1}$$

resulting in the differential equation

$$\ddot{\psi} = \frac{1}{I_z} \left[F_y(t).x_{FP} - F_x(t).y_{FP} \right] \tag{3.2}$$

where x_{FP} and y_{FP} are the x and y components of the foot position. Here, we approximate $F_x(t)$ and $F_y(t)$ by time averages $\overline{F_x}$ and $\overline{F_y}$ (Fig. 3.1b) for the sake of simplicity.

We first compute the solution to the differential equation 3.2 assuming general averaging forces $\overline{F_x}$ and $\overline{F_y}$. In the second part of this section, the average leg force in the mediolateral and fore-aft directions are calculated.

The model proposed by Jindrich et al. in [26] integrates the fore-aft leg force F_y to zero as it has an odd symmetry about mid-stance, and only uses the mediolateral leg force F_x to derive algebraic expressions that relate redirection and reorientation. When a force that has an odd symmetry is applied on a body, it first accelerates and then decelerates it, bringing it to rest eventually. Even though the final velocity of the body is zero, the force would have displaced the body. Similarly, the fore-aft force F_y acting on the center body would displace it by a certain yaw angle. We incorporate

the effect of the fore-aft force F_y on the model dynamics and the displacement caused by it with an aim to approximate turning behaviors more accurately. In order to do that, the solution to 3.2 through stance must be obtained in two parts:

1. From touchdown(TD) till mid-stance(MS) and
2. From mid-stance(MS) till end-stance(ES)

The initial conditions for the second part (ψ_2^i and $\dot{\psi}_2^i$) will be the final values for the quantities ψ and $\dot{\psi}$ of the first part (ψ_1^f and $\dot{\psi}_1^f$ respectively).

Integrating 3.2 from touchdown($t = 0$) till mid-stance($t = t_{MS} = y_{FP}/v$), with initial conditions $\psi_0^i = 0$ and $\dot{\psi}_0^i = 0$, we get

$$\dot{\psi}_1^f = \frac{y_{FP}}{I_z v} \left[\overline{F_y}.x_{FP} - \overline{F_x}.y_{FP}\right] \tag{3.3}$$

and

$$\psi_1^f = \frac{y_{FP}^2}{2I_z v^2} \left[\overline{F_y}.x_{FP} - \overline{F_x}.y_{FP}\right] \tag{3.4}$$

where $\dot{\psi}_1^f$ and ψ_1^f are the yaw velocity and the yaw angle of the center body at mid-stance.

After mid-stance, the fore-aft force direction inverts. Hence $\overline{F_y}$ in 3.2 should be replaced by $-\overline{F_y}$. Hence the equation 3.2 becomes

$$\ddot{\psi} = \frac{1}{I_z} \left[-\overline{F_y}.x_{FP} - \overline{F_x}.y_{FP}\right] \tag{3.5}$$

Integrating 3.5 from mid-stance($t = t_{MS} = y_{FP}/v$) till end-stance($t = t_{ES} =$

$2y_{FP}/v$), with initial conditions $\psi_2^i = \psi_1^f$ and $\dot{\psi}_2^i = \dot{\psi}_1^f$, we get

$$\dot{\psi}_2)f = -\frac{y_{FP}}{I_z v}(2\overline{F_x}) \tag{3.6}$$

and

$$\psi_2^f = \frac{y_{FP}^2}{I_z v^2}\left(-2\overline{F_x}y_{FP} + \overline{F_y}x_{FP}\right) \tag{3.7}$$

Hence, the yaw angle at the end of stance can be expressed as

$$\psi = \frac{y_{FP}^2}{I_z v^2}\left(-2\overline{F_x}y_{FP} + \overline{F_y}x_{FP}\right) \tag{3.8}$$

where $x_{FP} = h + l_0\ cos\alpha\ sin\beta$ and $y_{FP} = l_0\ cos\alpha\ cos\beta$. After deriving the expression for reorientation of the center body, an expression for redirection can be derived as shown in the next section.

3.3 Redirection of the CoM velocity

The redirection of the CoM velocity is assumed to depend only on the linear dynamics in the mediolateral direction. It is approximated as

$$d\theta \approx \frac{\Delta v_x}{v} \tag{3.9}$$

where Δv_x is the change in velocity accumulated in the mediolateral direction. It can be readily computed from the change in linear momentum,

$$m\Delta v_x = \int_0^{t_{end}} F_x(t)dt \tag{3.10}$$

The change in mediolateral velocity follows a half sine wave shape, so the mediolateral force can be approximated as

$$F_x(t) = F_x^{max} sin\left(\frac{\pi t}{t_{end}}\right) \tag{3.11}$$

From equations 3.10 and 3.11, we can compute the change in velocity accumulated in the mediolateral direction as

$$\Delta v_x = \frac{1}{m}\int_0^{t_{end}} F_x^{max} sin\left(\frac{\pi t}{t_{end}}\right) \tag{3.12}$$

where $t_{end} = \frac{2y_{FP}}{v}$ Upon integration equation 3.12, we get

$$\Delta v_x = \frac{4F_x^{max} y_{FP}}{mv\pi} \tag{3.13}$$

We know that $\frac{2}{\pi}F_x^{max} = \overline{F_x}$. So from 3.13, 3.9 becomes

$$d\theta = \frac{\Delta v_x}{v} = \frac{-2y_{FP}\overline{F_x}}{mv^2} \tag{3.14}$$

Hence, the redirection of the CoM velocity vector at the end of stance can be expressed as

$$d\theta = \frac{-2y_{FP}\overline{F_x}}{mv^2} \tag{3.15}$$

3.4 Ratio of the reorientation angle to the redirection angle

To compare our results with those published in [26], we also calculate the ratio of the reorientation angle(ψ) to the redirection angle($d\theta$). The ratio,

$$\frac{\psi}{d\theta} = \frac{\frac{y_{FP}^2}{I_z v^2}\left(-2\overline{F_x}y_{FP} + \overline{F_y}x_{FP}\right)}{\frac{-2y_{FP}\overline{F_x}}{mv^2}} \tag{3.16}$$

which simplifies to

$$\frac{\psi}{d\theta} = \frac{my_{FP}}{2I_z\overline{F}_x}\left[2\overline{F}_x y_{FP} - \overline{F}_y x_{FP}\right] \tag{3.17}$$

In order to compute ψ or $d\theta$, we need $\overline{F}_x$ and $\overline{F}_y$. We compute these expressions as shown in section 3.5.

3.5 Average Forces

The average forces $\overline{F_x}$ and $\overline{F_y}$ are estimated from the leg force

$$\mathbf{f}^{leg} = k(l_0 - l)\frac{\boldsymbol{l}}{l} \tag{3.18}$$

Here, the leg vector $\boldsymbol{l} = \boldsymbol{r_h} - \boldsymbol{r_{FP}}$, goes from the footpoint FP to the hip h. Thus, the average fore-aft force $\overline{F_y}$ will be negative from touchdown till mid-stance and positive from mid-stance to end-stance. The average mediolateral force will be negative for outward steps($\beta_0 > 0$) and positive for inward steps($\beta_0 < 0$).

mediolateral force

The mediolateral force rises until mid-stance and then drops, following a sinusoidal shape. The maximum is achieved at mid-stance, at which the CoM and hip are parallel to the footpoint ($y_{FP} = 0$). Maximum mediolateral force can be calculated by the expression

$$F_x^{max} = k(l_0 - l_{min})\frac{l_{x,min}}{l_{min}} \tag{3.19}$$

where

$l_{min} = \sqrt{z_{TD}^2 + x_{FP}^2} = \sqrt{l_0^2 - y_{FP}^2} = \sqrt{l_0^2 - l_0^2 \cos^2\alpha_0 \; cos^2\beta_0}$ and

$l_{x,min} = x_{FP} = h + l_0 \; cos\alpha_0 \; sin\beta_0$.

After substituting the values of l_{min} and $l_{x,min}$ in Eq. 3.19, we get

$$F_x^{max} = -k(h + l_0 \; cos\alpha_0 \; sin\beta_0)\left(\frac{1}{\sqrt{1 - cos^2\alpha_0 \; cos^2\beta_0}} - 1\right) \tag{3.20}$$

Assuming $F_x(t)$ has is a half sinusoid, the average force resolves to

$$\overline{F_x} = \frac{2}{\pi}F_x^{max} \tag{3.21}$$

Hence,

$$\overline{F}_x = -\frac{2}{\pi}\, k\, l_0\, \cos\alpha_0 \sin\beta_0 \left(\frac{1}{\sqrt{1-\cos^2\alpha_0 \cos^2\beta_0}} - 1 \right) \tag{3.22}$$

Fore-aft force

Unlike the mediolateral force, the appropriate average $\overline{F_y}$ of the fore-aft force is more difficult to determine. It is an odd function with respect to mid-stance and can be roughly approximated as a complete sine wave. However, the maximum F_y^{max} is reached before mid-stance and can not be determined by simple algebra. We thus compute the maximum by root finding of the actual fore-aft force. The actual fore-aft force is given by the expression

$$F_y(t) = k(l_0 - l(t))\frac{l_y(t)}{l(t)} \tag{3.23}$$

where $l_y = y_{FP}(t)$ and $y_{FP}(t) = y_{FP} - vt$. then $F_y(t)$ becomes

$$F_y(t) = -k\,(y_{FP} - t\,v) \left(\frac{l_0}{\sqrt{{l_0}^2 + t^2\,v^2 - 2\,y_{FP}\,t\,v}} - 1 \right) \tag{3.24}$$

To find the maximum force in the y-direction, we need compute $argmax(F_y(t))$ and estimate the force at that time instant. This is done by solving

$$\frac{\partial F_y(t)}{\partial t} = 0 \tag{3.25}$$

Solving equation 3.25, we get

$$t_{max} = \begin{pmatrix} \frac{y_{FP} - \sqrt{\left(l_0{}^3 - l_0\, y_{FP}{}^2\right)^{2/3} - l_0{}^2 + y_{FP}{}^2}}{v} \\ \frac{y_{FP} + \sqrt{\left(l_0{}^3 - l_0\, y_{FP}{}^2\right)^{2/3} - l_0{}^2 + y_{FP}{}^2}}{v} \end{pmatrix} \tag{3.26}$$

Out of the two solutions above, the first one pertains to the peak before mid-stance and the second one to the peak after mid-stance.

With $t_{max} = \frac{y_{FP} - \sqrt{\left(l_0{}^3 - l_0\, y_{FP}{}^2\right)^{2/3} - l_0{}^2 + y_{FP}{}^2}}{v}$, we compute

$$\begin{aligned} \overline{F_y} &= \frac{2}{\pi} F_y(t_{max}) \\ &= -\frac{2}{\pi}\, k l_0 \left(\frac{1}{(1 - \cos^2\alpha_0 \cos^2\beta_0)^{1/3}} - 1 \right) \\ &\quad \sqrt{(1 - \cos^2\alpha_0 \cos^2\beta_0)^{2/3} + \cos^2\alpha_0 \cos^2\beta_0 - 1} \end{aligned} \tag{3.27}$$

3.6 Results & Discussion

Using equations 3.8 and 3.14 of the analytical model, we scan over the control inputs alpha and beta similar to section 2.3 to compute the system states ψ and $d\theta$ at the apex $i+1$. Fig.3.3a depicts the reorientation of the center body ψ as observed from the return map R and Fig.3.3b depicts the reorientation predicted by the analytical model.

Upon comparison, from Fig.3.3, it is evident that the reorientation behaviors predicted by the analytical model proposed in this chapter are qualitatively similar to the reorientation behaviors computed using the full dynamics return map R. By increasing the complexity of the analytical model, the variance between both the plots

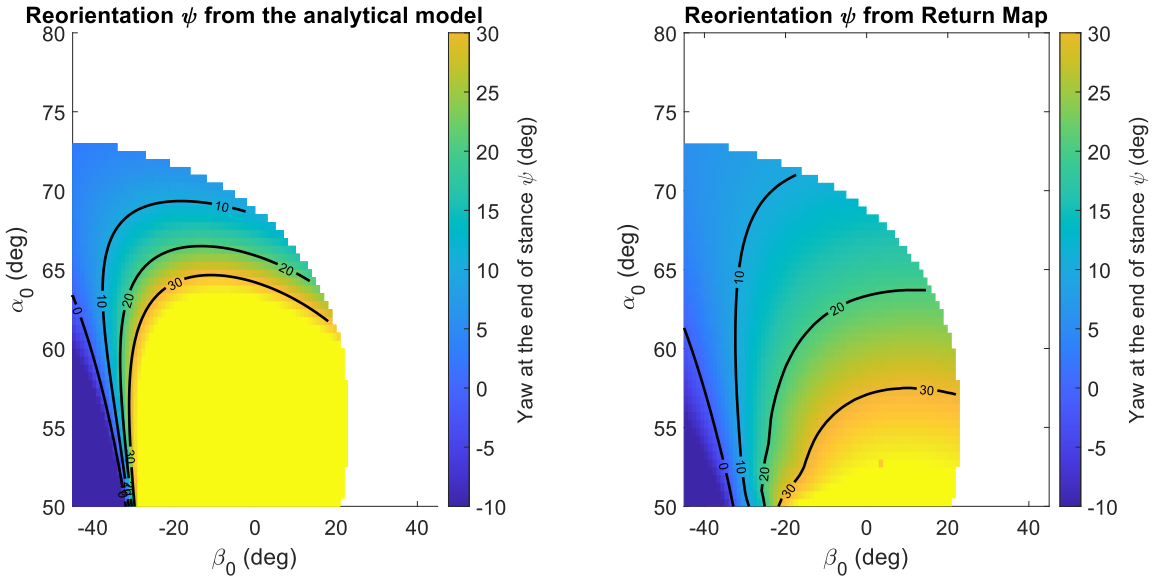


Figure 3.3: Comparison of yaw motion observed in the return map and from the analytical model

can be minimized but the aim was to create a simple model that can represent the turning dynamics of legged running systems well enough using simplified equations.

Similarly, Fig.3.3a depicts the redirection predicted by the analytical model and Fig.3.4b depicts the redirection of the CoM velocity $d\theta$ as observed from the return map R. Upon comparison, it can be deduced that the redirection angles predicted by the analytical model are qualitatively similar to the ones observed from the return map R.

By integrating the fore-aft force in 2 stages (from touch down till mid-stance and from mid-stance till end-stance), its effect on turning behaviors can be realized. Including the effect of the fore-aft force on turning dynamics and using spring leg force profiles instead of preset sinusoids, our model offers a more accurate representation of turning properties of the RBSMM than the one offered by Jindrich et. al. [26]. To

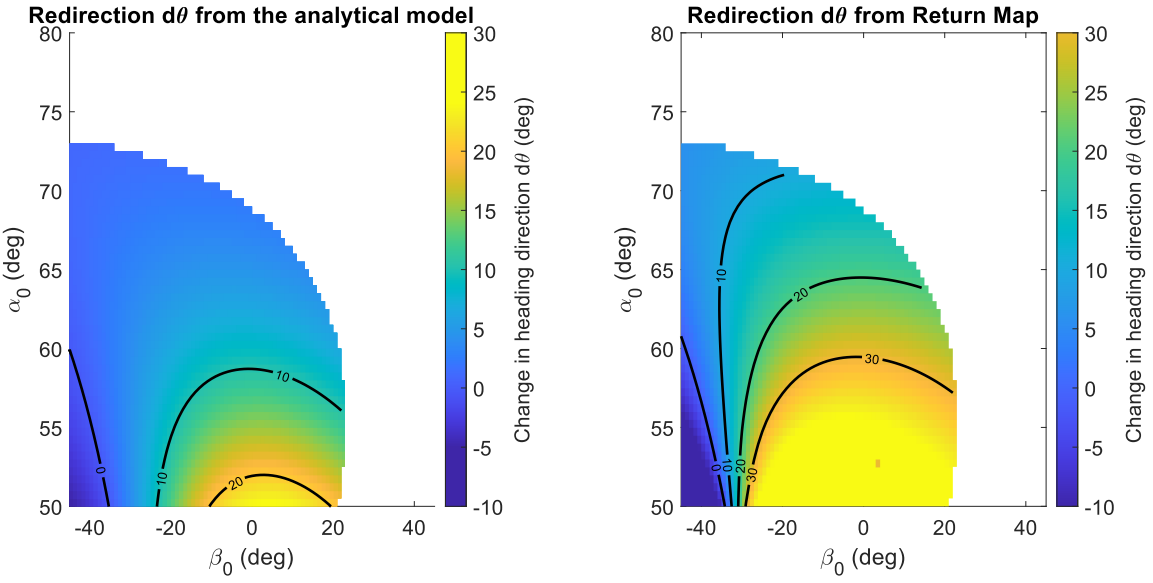


Figure 3.4: Comparison of redirection observed in the return map and from the analytical model

the best of our knowledge, this is the first work that provides a holistic model for turning dynamics in its analytical form as well as verifies its validity using a simplified running model (RBSMM) numerically.

The agreement between the analytical model and the numerical results from the return map validates the RBSMM as a more realistic alternative for studying running turns than other simplified gait models. These results motivated us to proceed with building a controller to achieve stable running and synchronous turning, which we discuss in detail in the next chapter.

Chapter 4

Controller for stable running and turning with center body

In this chapter, we identify the control inputs u that drive the initial state of the system to an arbitrary target state by analysing the model behavior, represented by the return map R. Using these control inputs, we formulate a deadbeat control policy for synchronous and stable running turns on uneven terrain.

4.1 Formulation of the Deadbeat Control Policy

Similar to the 3D SMM deadbeat control policy proposed by [1], we develop a deadbeat controller that selects leg placement positions that prioritize apex height tracking against ground disturbances over redirection as apex height has a higher influence on the stability of the system. The idea is to track the target apex height and a desired

turning angle in a feed-forward fashion and a PID controller maintains the center body orientation at a desired state in a feedback manner.

We achieve the feed-forward control in two stages:

1. Given an apex height z_i, identify the set of control inputs $[\alpha\ \beta]$ that take the system to a desired apex height z^* with an error of 0.01m.
2. From the set of control inputs identified in the previous step, select one control input $[\alpha\ \beta]$ that causes the least possible error in the desired redirection angle $d\theta^*$

Fig. 4.1 depicts the 2 stages of the constrained optimization process for an arbitrary target apex height z^*=1m and redirection of $d\theta^* = 15^o$, for a system originating at an initial apex height z_i =1m. Each filled grid block in the figure denotes a leg placement position that satisfies the constraints applied through the optimization process. Initially, all the grid blocks are filled, implying that all possible leg placements under consideration result in a secondary flight phase for the given initial apex height, i.e. the system doesn't fail and fall into the ground.

In the first stage of optimization for this example, we first filter out the leg placement positions that drive the system to reach a target apex height of 1m with a $\pm 0.01m$ error. These leg placements are represented as filled blocks in the second grid of the Fig. 4.1.

In the second stage of optimization for this example, we isolate one leg placement position that drives the system to a redirection angle of 15^o with minimum error from the leg placements filtered out in the previous stage. This leg placement is depicted in the 3rd grid as a filled block. In this example, through constrained optimization,

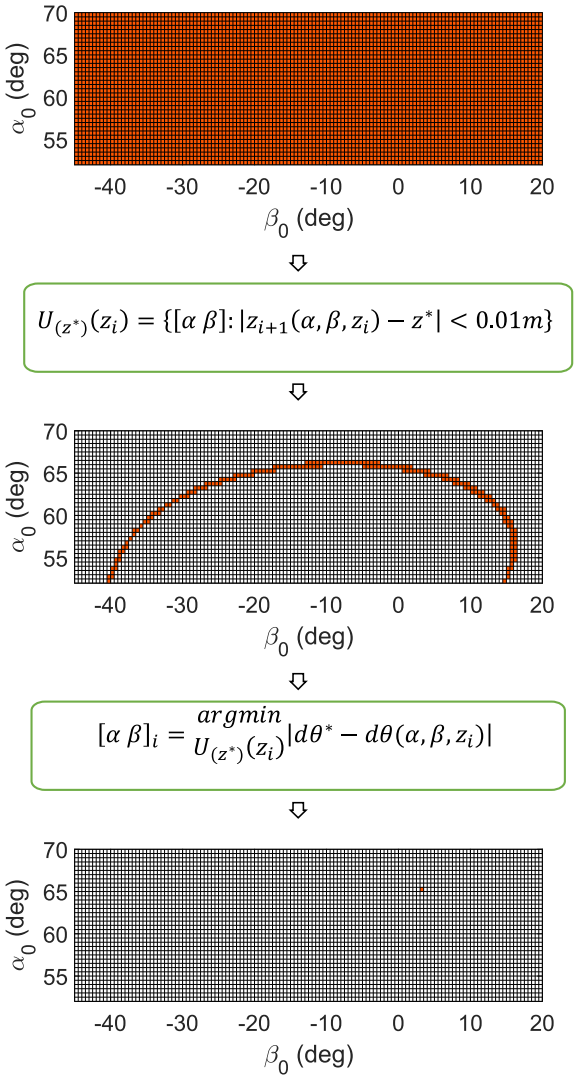


Figure 4.1: Availability of solutions after each stage of constrained optimization on the behavior function represented as orange blocks on the grid in α and β

.

we we identified an angle of attack $\alpha[z_i = 1m, d\theta^* = 15^o, z^* = 1m]$ and an angle of splay $\beta[z_i = 1m, d\theta^* = 15^o, z^* = 1m]$ that drives the system to a set of desired target states from an initial target state.

By following the constrained optimization steps as specified, we constructed the tables $\alpha[z_i, d\theta^*, z^*]$ and $\beta[z_i, d\theta^*, z^*]$. These tables are the control policies that drive a system with the initial states $(z, d\theta)_i$ into the target states $(z, d\theta)^*$. If the system's initial apex height z_i restricts the range of target apex heights and target redirections, the control policy picks the nearest neighbor height and then a nearest neighbor redirection error.

4.2 Model Behaviour

Fig. 4.2 and Fig. 4.3 shows the discretized model behavior for two different initial heights z_i . The control inputs α and β are represented as foot point locations relative to the CoM of the center body, projected onto the transverse plane.

The discrete functions $z_{i+1}[\alpha, \beta, z_i]$ and $d\theta[\alpha,\ \beta, z_i]$ are well behaved and their projection on the transverse plane forms an elliptical disk segment. Unlike the the 3D SMM behavior observed by Wu Geyer [1], the behavior of the RBSMM is not symmetrical about the frontal axis due to the off- centred hip joint through which the leg force acts on the center body (Fig. 4.2a&b and Fig. 4.3c&d).

The Foot positions that result in equal apex heights z_1 form concentric ellipse segments with decreasing heights, radially. Similarly, the configurations with equal redirections $d\theta$ form rays depicted in black in Fig. 4.2c and 4.3c. With respect to the frontal axis, outward placement of the right leg results in positive redirec-

tions(away from the sagittal plane) and inward leg placements result in negative redirections(towards the sagittal plane). However, a smaller range of negative turns is possible as opposed to positive turns and vice versa with the other leg in the consecutive stride. As a result, individual behavior goals may not always be achievable simultaneously. For instance, a target redirection of $d\theta^* = -30^o$ and a target height of $z^* = 0.95$m cannot be achieved with the right leg but a target redirection of $d\theta^* = 30^o$ and a target height of $z^* = 0.95$m can be achieved (Fig. 4.2c and Fig. 4.3c).

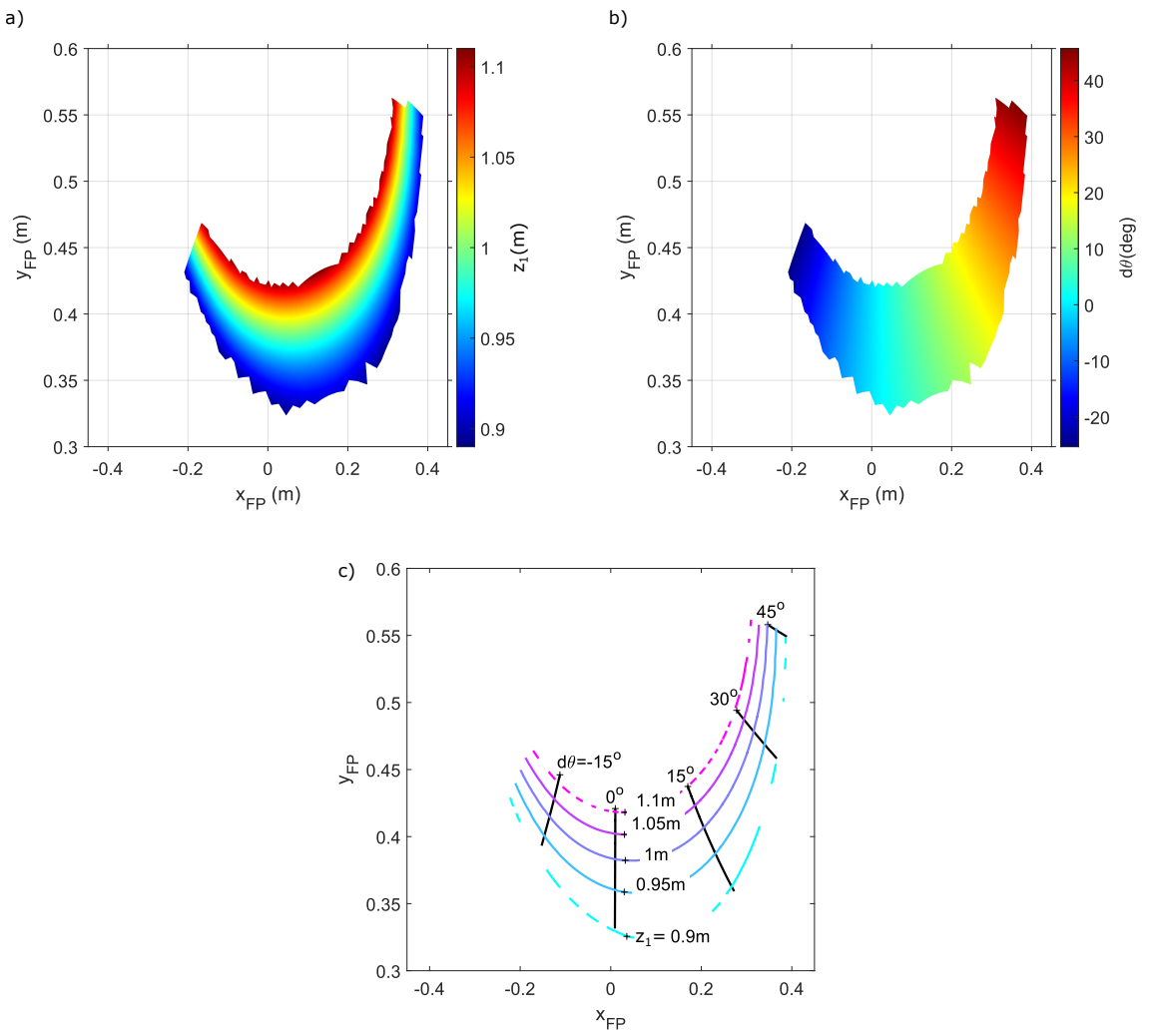


Figure 4.2: Numerical representations of model behavior for an initial apex heights z_i= 1.05m.
(a) Apex height $z_{i+1}[\alpha, \beta, z_i]$ as functions of leg placement projected onto horizontal plane.
(b) Redirection of the CoM $d\theta[\alpha, \beta, z_i]$ as functions of leg placement projected onto horizontal plane.
(c) Closeup section with apex height and redirection behaviors overlaid as contour plots. Contour lines show leg placement positions $[x_{FP}, y_{FP}]$ that produce the same apex height z_{i+1}(ring sections) or redirections $d\theta$ (rays).

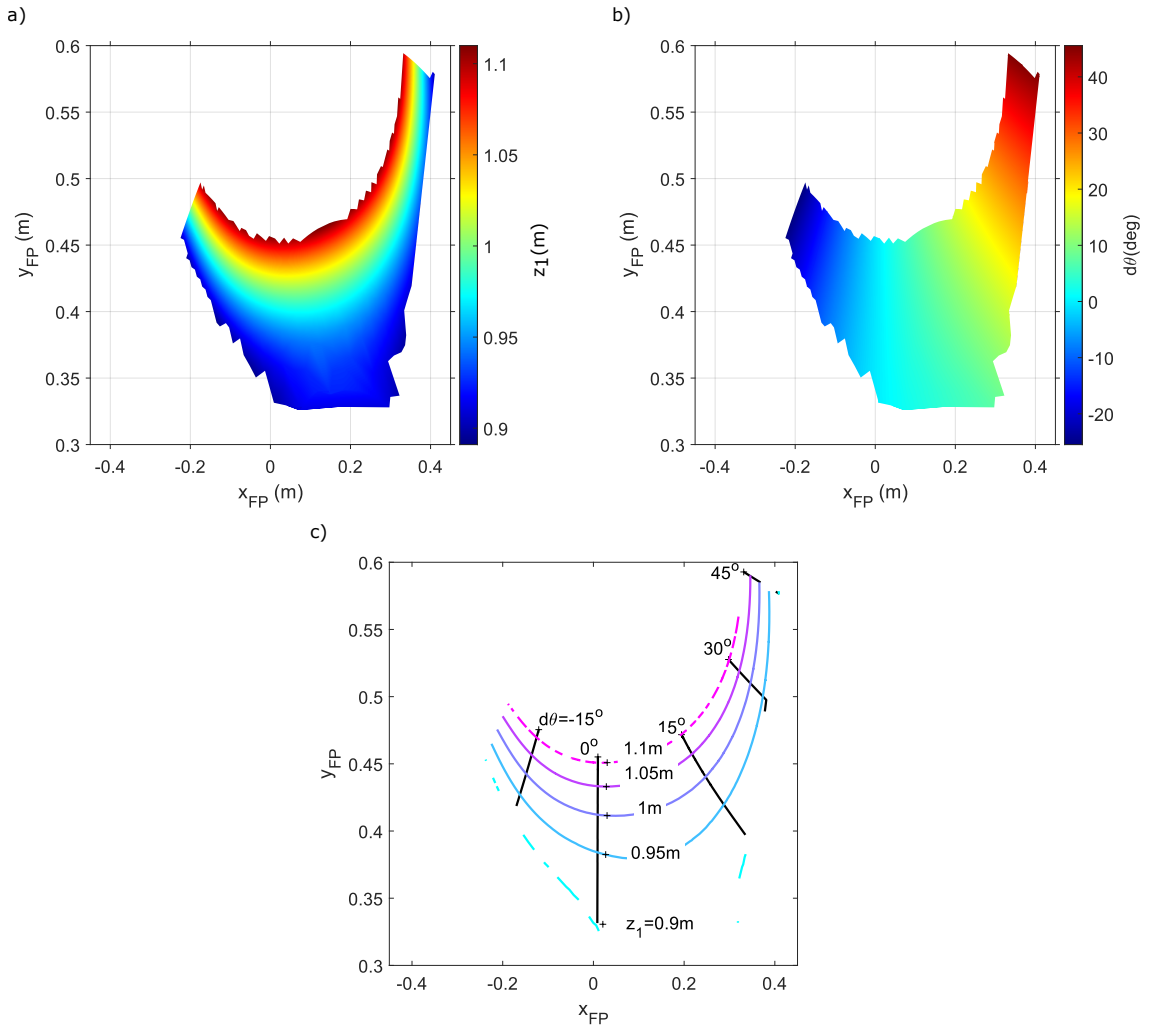


Figure 4.3: Numerical representations of model behavior for an initial apex height z_i= 0.95m.
(a) Apex height $z_{i+1}[\alpha, \beta, z_i]$ as functions of leg placement projected onto horizontal plane.
(b) Redirection of the CoM $d\theta[\alpha, \beta, z_i]$ as functions of leg placement projected onto horizontal plane.
(c) Closeup section with apex height and redirection behaviors overlaid as contour plots. Contour lines show leg placement positions $[x_{FP}, y_{FP}]$ that produce the same apex height z_{i+1}(ring sections) or redirections $d\theta$ (rays).

4.3 Time-based control policy formulation

Research in the past has exploited the ballistic flight trajectory of the 2D SMM to develop time-based control policies by embedding the deadbeat behavior across a range of apex heights instead of selecting one leg placement for one target apex state [27],[28], [29]. This results in a control policy that continuously varies the angle of attack α through the flight phase as a function of the fall-time t_i, accommodating for the variation in the landing height, helping the system achieve robust running on terrains with disturbances up to 30% of the leg length without directly being aware of the terrain disturbances. More recently, Wu Geyer [1] has designed similar time-based control policies in both α and β for the 3D SMM owing to the fact that $\alpha[z_i, d\theta^*, z^*]$ belongs to the pair of control input for α and β for any given $[z_i, d\theta^*, z^*]$[1]. This enabled the 3D SMM achieve robustness to change in ground disturbances up to 30% of the rest leg length.

A similar control policy can be constructed for the RBSMM by converting the existing apex height based control policy into fall-time-based control policy. Fig. 4.4 depicts the apex height based control policy to achieve a target height of $z^* = 1$m as contour plots. For this target height, all possible initial heights $z_i \in [0.9, 1.1]m$ can be projected to $z_{i+1} = z^*$ in a single step. From the Fig. 4.4, it can be deduced that the system can achieve steady state running ($z_i = z_{i+1} = 1m$) for desired redirections $\in [-25^o, 45^o]$.

We utilise the same transformation used by Wu Geyer [1] to build a time-based control policy that does not need terrain awareness to achieve stable running and

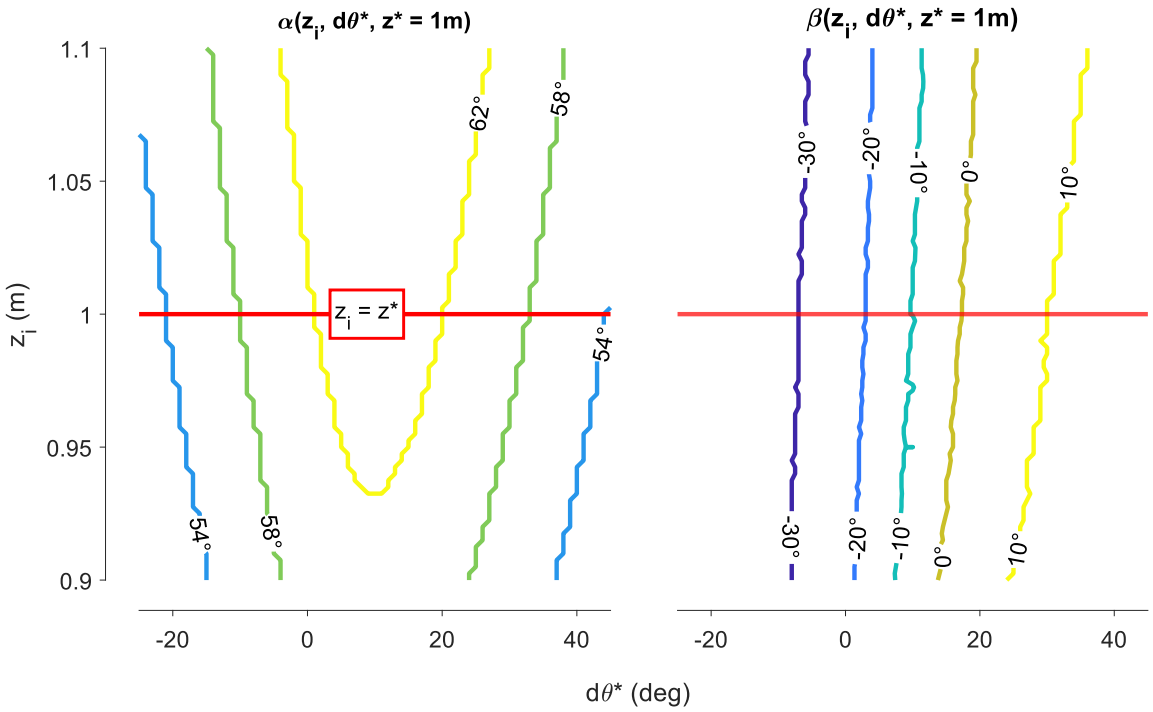


Figure 4.4: Control policy to achieve $z^* = 1$m.
Control tables $\alpha\,[z_i, d\theta^*, z^*]$ and $\beta\,[z_i, d\theta^*, z^*]$ are shown as contour plots for all possible initial heights z_i with ranges of target redirections $d\theta^*$. The red line indicates the steady-state control $z_i = z_{i+1} = z^*$.

synchronous turning. The time-based policy uses the ballistic nature of the system through the flight phase and replaces the initial apex height z_i with a function of time. The height of the CoM of the system through the flight phase can be expressed as

$$z_{flight} = z_{TD} + \frac{1}{2}gt^2 \tag{4.1}$$

where $z_{TD} = l_0 sin\alpha$, which is the CoM height at touchdown. Using this relation, the control table $\alpha[z_i, d\theta^*, z^*]$ can be transformed into a time-based control table as shown in Eq 4.2.

$$\alpha[z_i, d\theta^*, z^*] \rightarrow \alpha[t_i, d\theta^*, z^*] \tag{4.2}$$

where the time elapsed since apex $t_i = \sqrt{\frac{2}{g}(z_i - l_0\ sin\ \alpha[z_i, d\theta^*, z^*])}$.

Fig. 4.5 represents the varying angles of attack α for a system starting at an initial apex height z_i as a function of the fall-time, in the sagittal plane for ease of understanding. Even though the terrain is unobserved, the angle of attack α is varied as a function of time such that whenever the leg makes contact with the ground, the angle of attack is the appropriate one to drive the system to a target apex height of z^* with respect to the ground level at the foot point. The varying leg positions parametrized by α in the sagittal plane are depicted in the Fig. 4.5 as green dotted lines. The trajectory represented in red in the Fig. 4.5 is for the flat ground case, where the fall-time is t_i. If there is a depression ahead, the angle of attack becomes steeper as the fall-time increases. This phenomenon is depicted in the black and blue trajectories in the Fig. 4.5. Based on this law, the system can be driven to a desired apex height z^* even without terrain awareness.

The α control table is only half of the joint pair of control inputs [α β]. The

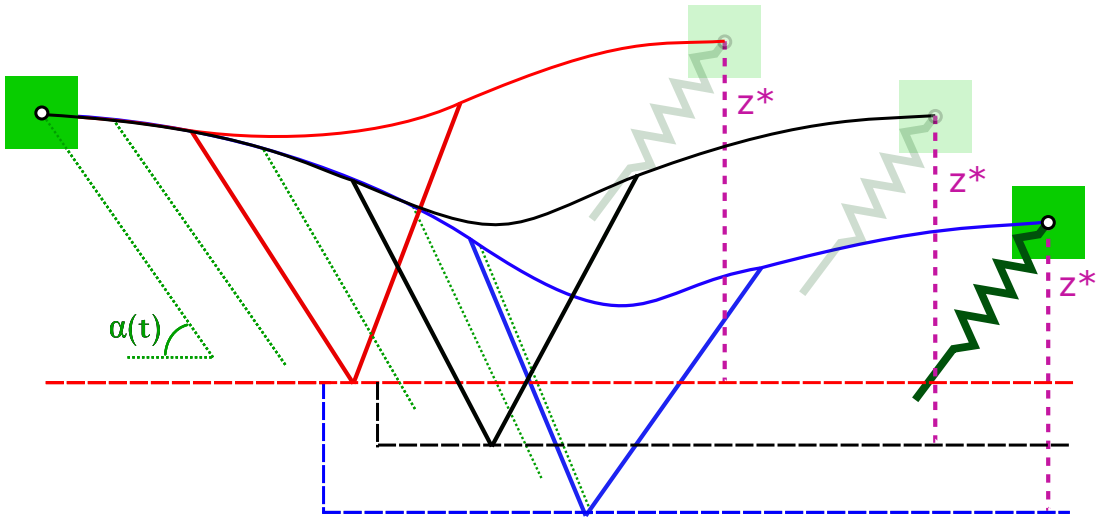


Figure 4.5: Apex height based leg placement control policy implemented as a time law on terrain with uncertainty

β control table can also be transformed in the same way. The transformed control tables for a targeted apex height $z^* = 1m$ are shown in the Fig.4.6. The red line indicates the steady state control of $z_i = z_{i+1} = z^*$.

$$\beta[z_i, d\theta^*, z^*] \rightarrow \beta[t_i, d\theta^*, z^*] \tag{4.3}$$

The controller has no memory of the older system states before the current step. Each step of a continuous run will be treated as a completely isolated step with the final system states of the previous step as the initial system states of the current step. So it is vital that the initial system states at each step match with the original initial system states used while computing the return map. Deviations from this cause the controller to deviate from its expected behavior.

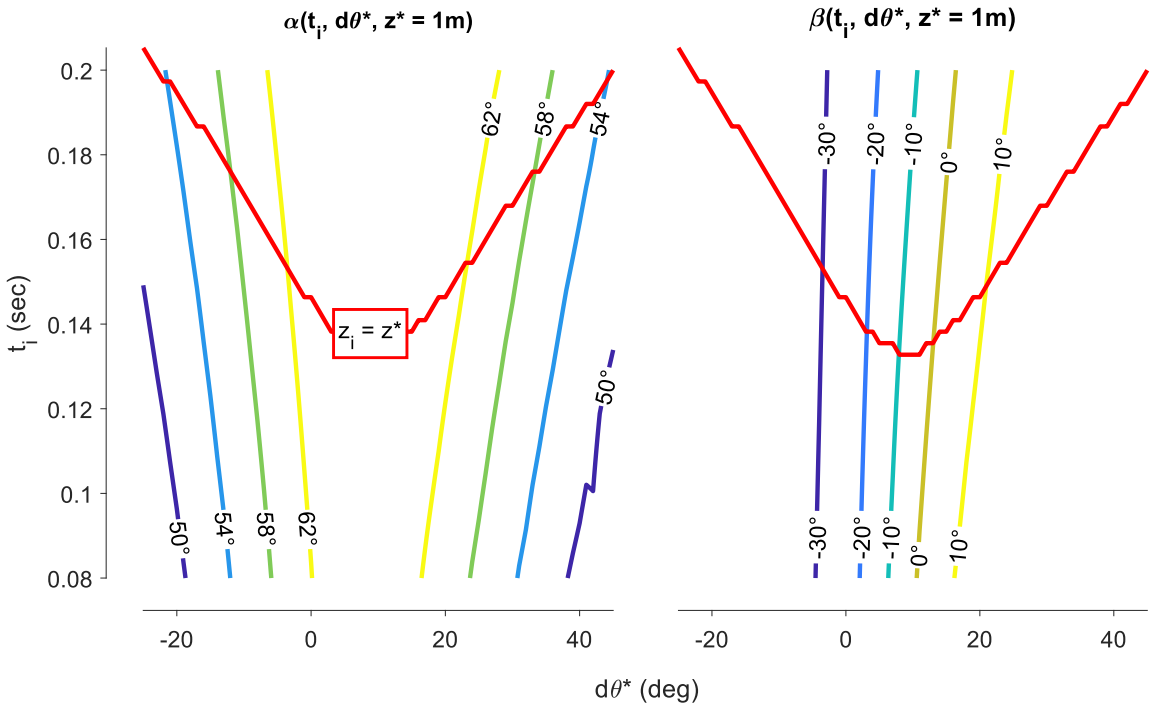


Figure 4.6: Control policy to achieve $z^* = 1$m.
Control tables $\alpha\,[t_i, d\theta^*, z^*]$ and $\beta\,[t_i, d\theta^*, z^*]$ are shown as contour plots for different fall times t_i with ranges of target redirections $d\theta^*$. The red line indicates the steady-state control $z_i = z_{i+1} = z^*$.

4.4 Results & Discussion

4.4.1 Performance of the control policy in simulation

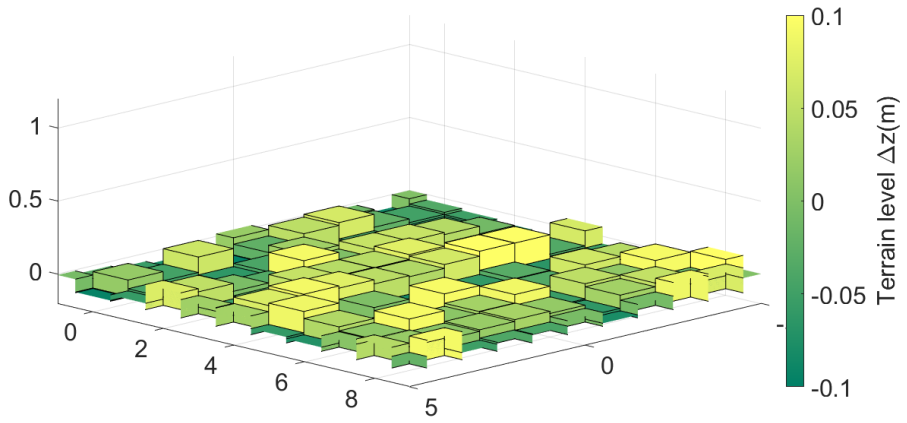


Figure 4.7: Random terrain with disturbances up to $\pm 0.1m$ to test the performance

The time-based control policy constructed as per Eq. 4.2 and 4.3 was tested for robustness to changes in terrain height. A terrain with random disturbances of $\Delta z = \pm 0.1m$ (up to 20% of the rest leg length) distributed uniformly across the ground as 1mx1m tiles were generated. The generated terrain is shown in the Fig. 4.7. The system was initialized at an apex height $z_0 = 1m$ and a forward velocity of $v_0 = 5ms^{-1}$ with its center body perfectly aligned with the global frame. At every consecutive apex, a new target redirection $d\theta^*$ was generated at random and the system was supposed to track the redirection requirement with minimum error, synchronously while maintaining its stability against the rough terrain. The performance of this controller against the aforementioned disturbances in terrain is presented in Fig. 4.8.

Fig. 4.8a shows the aerial view of the random navigation trail created for 200 steps of running (500m long). At every consecutive apex, a new target redirection $d\theta^*$ in the range of -25^o to 45^o was generated, which are marked as red points and the resulting CoM trajectory as a blue line, respectively with a maximum terrain disturbance of $\Delta z_g = 0.2m$. The occurrence of changes in terrain experienced along this path between consecutive steps is shown in Fig. 4.8b.

Fig. 4.8c shows the error in target apex heights reached by the system in the presence of terrain irregularities for a run of 200 steps without stumbling. Although the total energy is varying constantly due to the active torques applied, the system remains stable throughout the run with a mean absolute error of 0.0127m in apex height, where the maximum allowed error for apex height tracking is 0.02m. Fig. 4.8d shows the error in target redirections reached by the system in the presence of terrain irregularities. The error becomes larger with increasing terrain irregularity, though it stays under $\pm 3^o$ in the majority of scenarios. Hence, the controller can achieve terrain blind synchronous running turns while maintaining gait stability and is robust to irregularities in terrains up to 20cm, which is 20% of the rest leg length l_0.

4.4.2 Discussion

Using the single step apex return map generated using the RBSMM, we developed functional relationships between leg placement positions and synchronous turns, which were shown in Fig. 4.2 and Fig. 4.3. These relationships differ significantly from the ones observed in Fig. 2 of [1]. When the authors of [1] attempted to implement

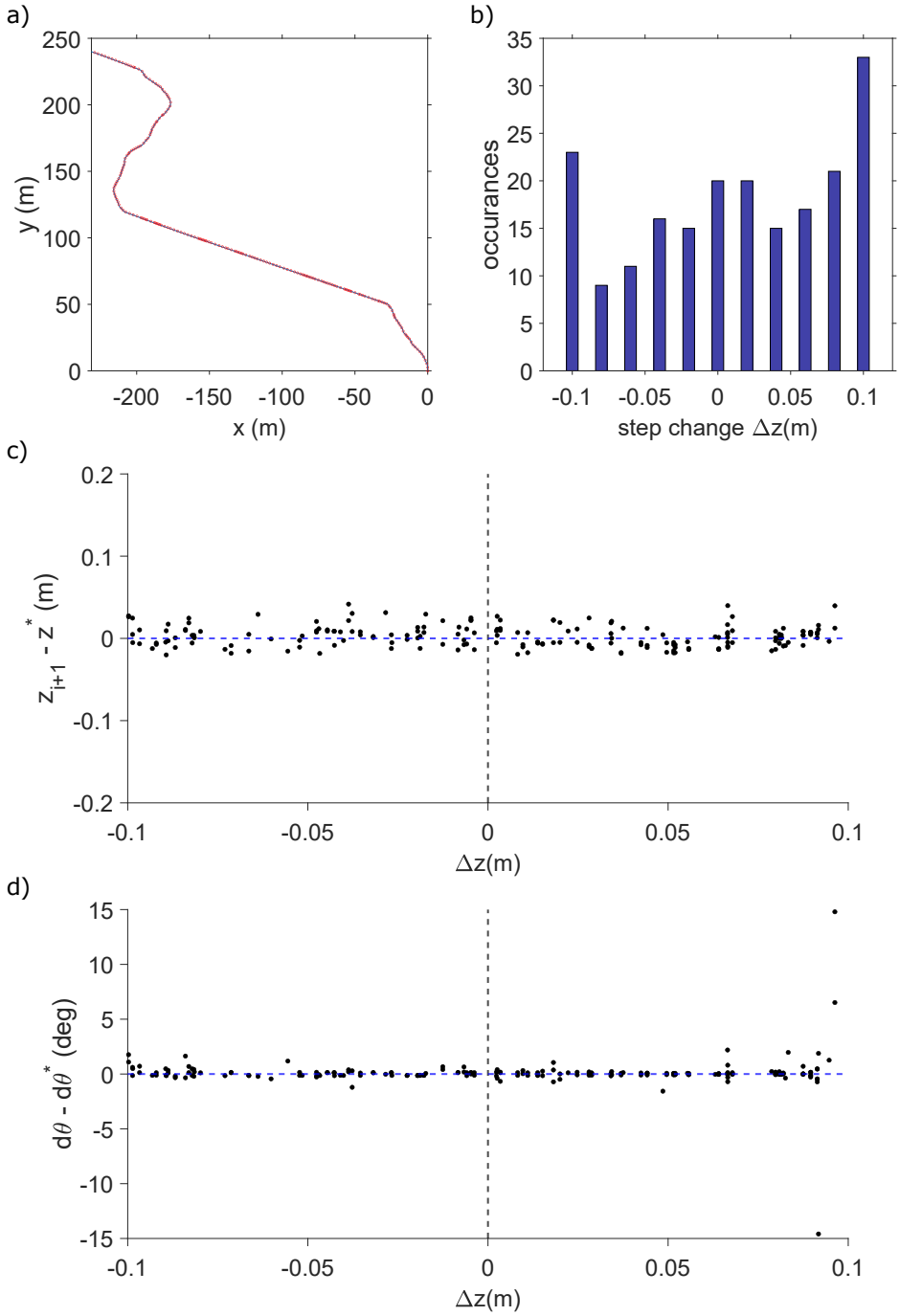


Figure 4.8: Performance of time-based control with target height $z^* = 1$m.
(a) Aerial view of random navigation trail with 200 steps with way points (red points), resulting in the CoM trajectory (blue line), respectively.
(b) Histogram of encountered step changes.
(c) Error in apex height $\epsilon_z = z_{i+1} - z^*$ as functions of ground level z_g.
(d) Corresponding redirection errors $\epsilon_{d\theta} = d\theta - d\theta^*$.

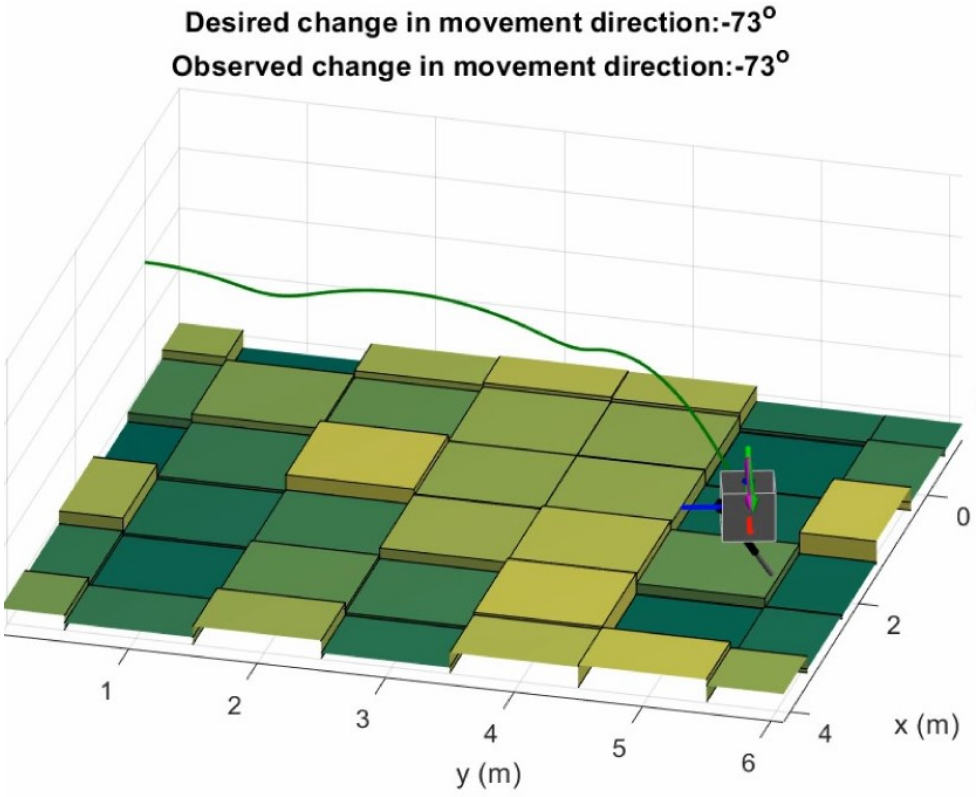


Figure 4.9: Maximum redirection $d\theta^* = 70^o$ in 2 consecutive steps

the deadbeat controller they formulated on a real-world sagittal planar biped, the robot required additional controllers to counteract the pitching caused by [1]'s leg placement strategy. These additional controllers interfered with the original deadbeat controller, so additional modifications were necessary [30]. By incorporating the effects of the center body control on the linear dynamics of the feed-forward system while computing the return map, the controller proposed in this chapter can be more readily transferred on to a real robot.

Using these functional relationships, we developed a controller to achieve synchronous running turns at high speeds, while maintaining gait stability. The controller is deadbeat for target redirection in the range of -25^o to 45^o while using the right leg in stance and -45^o to 25^o while using the left. While the maximum redirection that can be achieved in a single step is 45^o, the maximum redirection that can be achieved in 2 consecutive steps is limited to 70^o (Fig. 4.9). Multiple steps are needed if the desired redirection lies outside the specified redirection ranges.

We also embedded this control law in time to derive a terrain agnostic time-based control policy that can execute stable running and synchronous turning on terrain with uncertainties up to 20% the rest leg length (20cm). If the terrain has uncertainty, the controller selects control inputs from the control tables $\alpha[t_i, z^*, d\theta^*]$ and $\beta[t_i, z^*, d\theta^*]$, as a function of t_i, the period of free fall since apex till touchdown. If the uncertainty is positive, i.e. there is an increased elevation in the ground level from the previous step, the controller returns a shallower angle of attack α and returns a steeper α if the uncertainty is negative, for a desired turning angle $d\theta^*$. A corresponding β is also selected similarly. For instance, if the system with an initial apex height

$z_i = 1m$ is attempting to track an apex height of $z^* = 1m$ and achieve a redirection of $d\theta^* = -10^o$ simultaneously on flat ground, the time of flight t_i becomes 0.1733secs and the controller returns an α of 58.5^o. If there is an elevation of $\Delta z_g = +10cm$ in the ground level at the spot of landing, the effective apex height becomes $z_i = 0.9m$, making the time of flight t_i 0.1570secs and the controller returns α of 56^o. Similarly, and the controller returns an α of 60^o if there is a depression of $\Delta z_g = -10cm$ for a time of flight $t_i = 0.2184$secs.

Chapter 5

Conclusions

5.1 The big picture

From this work, we gained a deeper understanding of the influence of rotational dynamics of the center body on a running system's turning dynamics and stability. To the best of our knowledge, this is the first attempt at modeling rotational dynamics of a spring-legged running system's center body, unlike the point-mass models like the SMM and LIPM. From the proposed RBSMM, we draw functional relationships between leg placement position in the transverse plane, the system stability, the redirection of the runner's CoM velocity and the reorientation of its center body(4.2 and Fig. 4.3). Using this information, we developed a time-based running and steering controller to successfully achieve robust running and synchronous turning of a bipedal running system(RBSMM) on terrains with large uncertainties up to 20% of the rest leg length l_0.

5.2 Comparison to existing controllers

Simplified running models are generally a lower-order representation of the full-order dynamics of running. Due to this, the dynamics omitted out for simplicity have to be accounted for, by using additional control strategies, which oftentimes interferes with the simplified model's assumed dynamics [31], [30]. Alternatively, explicitly modeling the full-order dynamics of a system is complex and time-consuming. The proposed RBSMM offers the right balance between the system complexity required to account for the additional dynamics and the simplicity needed for easy planning. In fact, the planning problem with the RBSMM is minutely more complex than the planning problem of the 3D SMM waypoint navigation [1] while accounting for the rotational dynamics of the center body. This system can be used more readily to build advanced controllers for bipedal robot running because the model more closely mimics a bipedal running system with a center body with an off-centered hip joint and alternate leg placement for each consecutive stride.

5.3 Key contributions

In this thesis, we make 3 major contributions:

1. In chapter 2, we introduced the rigid body spring mass model(RBSMM), a simplified running model that accounts for the rotational dynamics of running systems alongside linear dynamics. Unlike the 3DSMM turning controls proposed in [1], completely passive control of the RBSMM is not possible, due to the highly underactuated nature of the system. So we introduced a decentralized

control method and used it to control the apex height tracking and redirection passively in a feed-forward manner and the active feedback control employed to regulate the body reorientation to be synchronous with the redirection. We also develop functional relationships between leg placement, redirection of the CoM velocity and the reorientation of the center body using the RBSMM.

2. In chapter 3, we derived an algebraic relationship between leg placement, redirection of the CoM velocity and the reorientation of the center body using a simplified analytical model for running turns by modeling forces and kinematics during turning maneuvers. It was also shown that the turning dynamics observed in the return map are qualitatively similar to the ones computed from the analytical model, which validates the RBSMM's ability to approximate turning dynamics of running systems more accurately than other simplified gait models.

3. In chapter 4, we developed functional relationships between leg placement and synchronous turning through constrained optimization on the single step apex return map. Using these relationships, we built a deadbeat controller to achieve stable running and synchronous turning of the RBSMM, which is robust to terrain uncertainties of up to 20% of the rest leg length and demonstrated its performance in simulation. We also embedded this controller in a time-based control policy which does not require terrain awareness to produce stable running and turning gaits.

5.4 Scope for future research

Through this work, we attempt to fill the lapses in understanding of the influence leg placement in turning behaviors of running systems. We draw functional relationships between the foot placement position and the turning dynamics of the RBSMM. We also use these relationships to build robust controllers for terrain unaware running and turning on terrains with uncertainties without complicating the planning problem too much. The results show great promise in simulation. It remains to be seen how well these control policies transfer to physical robot hardware, and it opens up exciting avenues for future research.

The analytical model proposed in chapter 3 can be used to draw parametric insights into the turning dynamics of legged running systems.

While the controller proposed in chapter 4 attempts to restrict roll and pitch motions of the center body to zero, this is not a mandate. When humans execute running turns, a small amount of rolling in the trunk was observed [32]. Even for straight-line running, while or controller regulates the yaw of the center body to be zero (to align the body y-axis with the global y-axis), humans yaw their pelvis in clockwise and counter-clockwise directions alternatively through their running gait[33]. So, by relieving the feedback controller slightly and slowing the center body to roll pitch and yaw by small amounts, might reduce the effort currently being made to execute turning maneuvers. Allowing the center body to rotate too much might affect the stability of the system as well. So a right balance of the center body's freedom to rotate and the stability must be determined.

SMM based controllers proposed in the past for robots did not account for the rotational dynamics and the angular momentum of the center body. So these methods resorted to supplementary controllers that interfere with these controllers built based on the lower-order dynamics of the running system. The proposed controllers in this thesis were built by incorporating the rotational dynamics of the center body and a method to regulate them in the model design itself.

The controller that regulates the center body orientation requires 3 state inputs: the roll(ϕ), pitch(ω) and yaw(ψ) angles, which can be extracted from an IMU. Similarly, the system also needs to calculate the time of flight for which it needs a clock and a way to identify apex and touchdown conditions. The apex can be identified as a point where the vertical velocity of the system is zero, which can be acquired from the IMU as well. As for the identification of touchdown condition, actuators in the leg with torque feedback or a force sensor in the robot's feet will suffice. Hence, there is no need for global sensors or terrain awareness for the controller to work.

Bibliography

[1] Albert Wu and Hartmut Geyer. The 3-d spring–mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments. *IEEE Transactions on Robotics*, 29(5):1114–1124, 2013. 1.1, 1.2, 1.3, 2.1, 2.3.1, 2.5, 4.1, 4.2, 4.3, 4.3, 4.4.2, 5.2, 1

[2] Marc H Raibert, H Benjamin Brown Jr, Michael Chepponis, Jeff Koechling, and Jessica K Hodgins. Dynamically stable legged locomotion. Technical report, Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab, 1989. 1.1, 1.2

[3] Gerardo Bledt and Sangbae Kim. Extracting legged locomotion heuristics with regularized predictive control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 406–412. IEEE, 2020. 1.2

[4] Jessica K Hodgins. Three-dimensional human running. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3271–3276. IEEE, 1996. 1.2

[5] AD Perkins and KJ Waldron. Control of bipedal turning while running. In *Advances in Robot Kinematics: Motion in Man and Machine*, pages 301–308. Springer, 2010. 1.2

[6] Miles A Townsend. Biped gait stabilization via foot placement. *Journal of biomechanics*, 18(1):21–38, 1985. 1.2

[7] Martijn Wisse, Christopher G Atkeson, and Daniel K Kloimwieder. Swing leg retraction helps biped walking stability. In *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pages 295–300. IEEE, 2005. 1.2

[8] Shuuji Kajita, Mitsuharu Morisawa, Kanako Miura, Shin'ichiro Nakaoka, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro, and Kazuhito Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4489–4496. IEEE, 2010. 1.2

[9] Ryosuke Tajima, Daisaku Honda, and Keisuke Suga. Fast running experiments

involving a humanoid robot. In *2009 IEEE international conference on robotics and automation*, pages 1571–1576. IEEE, 2009. 1.2

[10] Reinhard Blickhan. The spring-mass model for running and hopping. *Journal of biomechanics*, 22(11-12):1217–1227, 1989. 1.2

[11] Sean G Carver, Noah J Cowan, and John M Guckenheimer. Lateral stability of the spring-mass hopper suggests a two-step control strategy for running. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(2):026106, 2009. 1.2, 2.1

[12] Raffaele M Ghigliazza, Richard Altendorfer, Philip Holmes, and D Koditschek. A simply stabilized running model. *SIAM review*, 47(3):519–549, 2005. 1.2, 2.1

[13] F Peuker and A Seyfarth. Adjusting legs for stable running in three dimensions. In *6th World Congress of Biomechanics (WCB 2010). August 1-6, 2010 Singapore*, pages 3–6. Springer, 2010. 1.2, 2.1

[14] Justin E Seipel and Philip Holmes. Running in three dimensions: Analysis of a point-mass sprung-leg model. *The International Journal of Robotics Research*, 24(8):657–674, 2005. 1.2, 2.1

[15] R Altendorfer, RM Ghigliazza, P Holmes, and Daniel E Koditschek. Exploiting passive stability for hierarchical control. In *Fifth International Conference on Climbing and Walking Robots (CLAWAR)*, pages 177–184, 2002. 1.2

[16] Andre Seyfarth, Hartmut Geyer, Michael Günther, and Reinhard Blickhan. A movement criterion for running. *Journal of biomechanics*, 35(5):649–655, 2002. 1.2

[17] Frank Peuker, Christophe Maufroy, and André Seyfarth. Leg-adjustment strategies for stable running in three dimensions. *Bioinspiration & biomimetics*, 7(3):036002, 2012. 1.2

[18] NJ Kohut, DW Haldane, D Zarrouk, and RS Fearing. Effect of inertial tail on yaw rate of 45 gram legged robot. In *International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machinces*, pages 157–164, 2012. 1.2

[19] Nicholas J Kohut, Andrew O Pullin, Duncan W Haldane, David Zarrouk, and Ronald S Fearing. Precise dynamic turning of a 10 cm legged robot on a low friction surface using a tail. In *2013 IEEE International Conference on Robotics and Automation*, pages 3299–3306. IEEE, 2013. 1.2

[20] Thomas Libby, Aaron M Johnson, Evan Chang-Siu, Robert J Full, and Daniel E Koditschek. Comparative design, scaling, and control of appendages for inertial reorientation. *IEEE Transactions on Robotics*, 32(6):1380–1398, 2016. 1.2

[21] Carlos S Casarez and Ronald S Fearing. Steering of an underactuated legged robot through terrain contact with an active tail. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2739–2746. IEEE, 2018. 1.2

[22] Devin L Jindrich and ROBERT J Full. Many-legged maneuverability: dynamics of turning in hexapods. *Journal of experimental biology*, 202(12):1603–1623, 1999. 1.2

[23] Brigitte Demes, Kristian J Carlson, and Theresa M Franz. Cutting corners: the dynamics of turning behaviors in two primate species. *Journal of experimental biology*, 209(5):927–937, 2006. 1.2

[24] Angela L Ridgel, Blythe E Alexander, and Roy E Ritzmann. Descending control of turning behavior in the cockroach, blaberus discoidalis. *Journal of Comparative Physiology A*, 193(4):385–402, 2007. 1.2

[25] Joshua L Proctor and P Holmes. The effects of feedback on stability and maneuverability of a phase-reduced model for cockroach locomotion. *Biological cybernetics*, 112(4):387–401, 2018. 1.2

[26] Devin L Jindrich, Thor F Besier, and David G Lloyd. A hypothesis for the function of braking forces during running turns. *Journal of biomechanics*, 39(9):1611–1620, 2006. 1.2, 2, 2.10, 2.5, 3, 1, 3.1, 3.1, 3.2, 3.4, 3.6

[27] J Schmitt and J Clark. Modeling posture-dependent leg actuation in sagittal plane locomotion. *Bioinspiration & biomimetics*, 4(4):046005, 2009. 4.3

[28] Andre Seyfarth and Hartmut Geyer. Natural control of spring-like running–optimized self-stabilization. In *Proceedings of the Fifth International Conference on Climbing and Walking Robots. Professional Engineering Publishing Limited*, pages 81–85, 2002. 4.3

[29] André Seyfarth, Hartmut Geyer, and Hugh Herr. Swing-leg retraction: a simple control model for stable running. *Journal of Experimental Biology*, 206(15):2547–2555, 2003. 4.3

[30] Albert Wu. *The theory, implementation, and evaluation of spring mass running on ATRIAS, a bipedal robot*. PhD thesis, Carnegie Mellon University, 2017. 4.4.2, 5.2

[31] William Martin. *Optimal control of compliant bipedal gaits and their implementation on robot hardware*. PhD thesis, Carnegie Mellon University, 2019. 5.2

[32] Aftab E Patla, Allan Adkin, and Tonya Ballard. Online steering: coordination

and control of body center of mass, head and body reorientation. *Experimental brain research*, 129(4):629–634, 1999. 5.4

[33] Nidhi Seethapathi and Manoj Srinivasan. Step-to-step variations in human running reveal how humans run without falling. *ELife*, 8:e38371, 2019. 5.4