3D DYNAMIC MODELING OF A SPHERICAL WHEELED SELF-BALANCING MOBILE ROBOT

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Ali Nail İNAL

August 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ömer MORGÜL(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof.Dr. Uluç SARANLI(Co-supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Hitay ÖZBAY

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Melih Çakmakçı

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of Graduate School of Engineering and Science

ABSTRACT

3D DYNAMIC MODELING OF A SPHERICAL WHEELED SELF-BALANCING MOBILE ROBOT

Ali Nail İNAL
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Ömer MORGÜL
August 2012

In recent years, dynamically stable platforms that move on spherical wheels, also known as BallBots, gained popularity in the robotics literature as an alternative locomotion method to statically stable wheeled mobile robots. In contrast to wheeled platforms which do not have to explicitly be concerned about their balance, BallBot platforms must be informed about their dynamics and actively try to maintain balance. Up until now, such platforms have been approximated by simple planar models, with extensions to three dimensions through the combination of decoupled models in orthogonal sagittal planes. However, even though capturing certain aspects of the robot's motion is possible with such decoupled models, they cannot represent inherently spatial aspects of motion such as yaw rotation or coupled inertial effects due to the motion of the rigid body.

In this thesis, we introduce a novel, fully-coupled 3D model for such spherical wheeled balancing platforms. We show that our novel model captures important spatial aspects of motion that have previously not been captured by planar models. Moreover, our new model provides a better basis for controllers that

are informed by more expressive system dynamics. In order to establish the expressivity and accuracy of this new model, we present simulation studies in dynamically rich situations. We use circular paths to reveal the advantages of the new model for fast maneuvers. Additionally, we introduce new inverse-dynamics controllers for a better attitude control and investigate within simulations the capability of sustaining dynamic behaviors. We study the relation between circular motions in attitude angles and associated motions in positional variables for BallBot locomotion.

Keywords: Dynamic Modeling, Balancing Mobile Robots, Underactuated Systems, Dynamic System Control, Attitude Control

ÖZET

KÜRESEL TEKERLEKLİ KENDİNİ DENGELEYEN BİR ROBOTUN ÜÇ BOYUTLU MODELLENMESİ

Ali Nail İNAL
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Ömer MORGÜL
Ağustos 2012

Son yıllarda küresel tekerlekler üzerinde hareket eden dinamik dengeli platformlar, bir başka deyişle BallBotlar, robotik literatüründe statik dengeli tekerlekli mobil robotlara alternatif olarak popülarite kazanmıştır. Özellikle dengeleriyle ilgilenmeye ihtiyaç duymayan tekerlekli platformlara nazaran BallBot daima kendi dinamiklerinden haberdar olup, aktif olarak dengesini sağlamak zorundadır. Şu ana kadar bu tarz platformlara basit düzlemsel modellerle yaklaşılmıştır. Üç boyutlu düzleme geçmek için birbirinden ayrıştırılmış dik modeller oksal düzlemler üzerinde kombine olarak kullanılır. Her ne kadar bu tür modellerle robotun hareketinin belirli yönlerini yakalamak mümkün olsa da, sapma açısı dönüşleri dönüşleri veya sert cisim hareketlerine bağlı bağıl ataletsel etkiler gibi hareketin doğasından kaynaklanan uzaysal özelliklerini temsil edemezler.

Tezde, küresel tekerlekler üzerinde hareket eden dinamik dengeli platformlar için tamamen bağıl 3 boyutlu yeni bir model ileri sürüldü. Yeni modelin, hareketin düzlemsel model tarafından yakalanamamış önemli uzaysal yönlerini yakaladığı gösterildi. Dahası, yeni model kontrolörler için, daha etkili sistem dinamikleri tarafından daha iyi bilgilendirilmiş bir temel sağlamaktadır. Yeni modelin doğruluğunu saptamak için, dinamik açıdan zengin simulasyon çalışmaları

sunuldu. Yeni modelin hızlı manevralardaki avantajlarını göstermek için dairesel yollar kullanıldı. Bunlara ek olarak, daha iyi davranış kontrolü için yeni ters-dinamik kontrolörleri tanıtıldı, ve dinamik hareketleri devam ettirebilme yetenekleri simulasyonlarla incelendi. Dairesel hareketlerdeki tavır açısı ile pozisyon değişkenlerinde buna bağlı oluşan hareket arasındaki ilişki BallBot hareketliliği açısından incelendi.

Anahtar Kelimeler: Dinamik Modelleme, Kendini Dengeleyen Hareketli Robotlar, Gereğinden Az Tahrikli Sistemler, Dinamik Sistem Kontrolü, Davranış Kontrolü

ACKNOWLEDGMENTS

First, I would like to express my special thanks to my supervisors, Prof. Ömer Morgül and Assoc. Prof. Uluç Saranlı, for their guidance, suggestions and valuable support throughout this thesis. I am very grateful to them for our productive conversations, where I have always been inspired by their expertise.

I would like to thank Prof. Hitay Özbay and Asst. Prof. Melih Çakmakçı for reading this thesis and being a member of my thesis committee.

One of the most rewarding aspects of my M.S. study was the opportunity to work with the amazing group of people in BDRL (Bilkent Dexterous Robotics and Locomotion) group, who helped me along the way both technically and morally. I am very thankful to İsmail Uyanık, Güneş Bayır, Deniz Kerimoğlu, Özlem Gür, Sıtar Kortik, Utku Çulha, Bilal Turan and Tolga Özaslan for our wonderful late night studies and discussions.

I would like to give my special thanks to my friends outside the laboratory, Veli Tayfun Kılıç, Hasan Hamzaçebi and Görkem Seçer for always being there to listen and motivate.

I would like to thank Bilkent University Electrical and Electronics Engineering department and especially to faculty members for giving me this opportunity by teaching me well.

I am also appreciative of the financial support from the Scientific and Technological Research Council of Turkey (TÜBİTAK).

Last but not least, I would like to offer my sincere love and thanks to my parents Osman Ali and Kamile İnal, and my sisters Fatma Betül and Feyza, for their undying love, support and encouragement in my whole life.

Contents

List of Figures

List of Tables

Dedicated to My Dear Family

Chapter 1

INTRODUCTION

Robust, efficient and controllable land-based mobility is a difficult but important challenge for the robotics community. In order to solve this, many options with different morphologies are introduced including wheeled [2], tracked [3], legged [4, 5] and even leaping [6] designs. Some of these are inspired from nature, and biological models, and the remaining designs are purely engineering solutions, like spherical robots,[7]. A new alternative is added to the aforementioned robot platform designs which can actively balance on "spherical wheels" [8, 9], and named as BallBot platforms. BallBots combine the advantages of wheeled systems, for their continuous contact with the ground, with bipedal morphologies, for their desirable features, such as their compatibility with human environments [10]. Although these platforms combined such advantages they are far more complex than wheeled systems. The operation of the BallBot is inherently dynamical, and hence these robots cannot be controlled through simpler kinematic methods.

1.1 Motivation and Existing Work

Although the principles behind the BallBot morphology is partially shared by planar balancing systems, such as the Segway, the omnidirectional mobility of the Ballbot is one of its most important and novel aspects, helping increase its popularity in robotic society recently. From the first proposed actuation idea for such systems, the Inverse Mouseball Drive (IMD) design [8, 11], to later versions that used omnidirectional wheel contact with the sphere for better control affordance and reduced friction [9, 12], all Ballbots are based on the same motion principle: The manipulation of the spherical wheel through rollers attached to the actuators on the body, resulting in a nonholonomic[1], underactuated motion. Despite the fact that there is substantial research on the BallBot platform, accurate control of associated dynamics for fast maneuvers and highly dynamical motions remains to be a challenging problem. Initial studies focused on motion on linear paths, which can be reduced to a 2D model in a sagittal plane. In the literature, PI controllers on the ball velocity and an LQR controller as an outer loop for the linearized system were often used to control body attitude [8, 11]. The decoupled combination of two of these planar models in two orthogonal directions of the horizontal plane formed the basis for controlling the system, an approach we refer to as a 2.5D model in this thesis. Recent extensions of this approach uses more sophisticated control methods for both the stabilization of body attitude degrees of freedom and the design of optimal attitude trajectories to travel along desired robot paths [1, 14] but does not extend on the expressivity of the model itself. Additionally, inertial disturbances and adding loads on the robot body brings further problems related with decoupled models [15]. Such situations reveal that this model may not be suitable for dealing with dynamic situations. Even though using these behavioral primitives as a basis more

[1]For a definition of nonholonomy in a robotic sense, see e.g. [13]

complex trajectories can be obtained through planning [16], the accuracy and expressivity of underlying mathematical models has not progressed much.

The fact remains that highly dynamic and fast maneuvers are the capabilities that distinguish these platforms from traditional alternatives. However, decoupled planar models, namely 2.5D models, are bound to lose their accuracy under such conditions. Moreover, such maneuvers involve large accelerations, forcing the body to deviate from vertical axis, and hence creating significant yaw rotation and coupled inertial effects. In order to support such challenging behaviors, more realistic mathematical models for BallBot systems are needed.

1.2 Contributions

In this thesis, we introduce a novel, three-dimensional model for BallBot platforms. Our new model can capture aspects of the BallBot's motion that 2.5D models cannot. After deriving the equations of motions for our models, we use them as basis for a simulated model of the platform. In addition, we describe novel inverse-dynamics controllers for accurate control of body attitude based on our model as well. In order to illustrate the performance of these model-based controllers, we use circular body attitude trajectories that result in dynamic movements of the platform.

Simulation results that we presented show the ability of our model to capture natural yaw dynamics of this morphology. Such dynamics arise from the rolling constraint between the ground and the ball, and cannot be captured by 2.5D models. Once again, through simulations, we present a characterization of how circular trajectories in the body attitude space result in circular trajectories in positional robot coordinates. As a consequence, the potential utility of the 3D model for motion planning and execution with BallBot platforms is also illustrated.

1.3 Organization of the Thesis

In the first part of the thesis, we start with the introduction and an analysis of existing planar BallBot models, followed by an overview of existing application of this in Chapter 2. In Chapter 3, we introduce our new three dimensional model for the BallBot platform. In Section 3.1, we give a brief introduction to quaternions and introduce necessary notation for the thesis. Subsequently, with the initial analysis of the model, in Section 3.2, we also present different variations of the model for the alternative actuation mechanisms. After introducing the simulation environment in Section 3.3, we describe initial simulations to compare the new model with the planar model.

In Chapter 4, we first introduce different controllers for regulating system behavior. After presenting a simple PD controller for the new 3D BallBot model, in Section 4.1.2 and Section 4.1.3, we present the new attitude controller based on an Inverse Dynamics approach, better illustrating the potential of the new model on dynamic trajectories. Then, in Section 4.2 we discuss and denote the effects of underactuation, and introduce a circular attitude angle profile to show the capabilities of the new model at the beginning of the Section 4.3. In subsequent sections, we investigate the performances of different controllers on variations of the 3D model with detailed simulations for circular trajectories. We study the yaw dynamics of our new model in Section 4.3.4 and the relations between the external variables and the circular profile variables from the simulation results in Section 4.3.5.

Finally, in Chapter 5, we conclude the thesis with a review of our work, and summarize the related open research topics.

Chapter 2

BACKGROUND: THE PLANAR BALLBOT MODEL

This chapter introduces necessary background for the BallBot as well as a summary of existing models and methods for control.

2.1 The Planar BallBot Model

The first model intended for the control and analysis of the BallBot platform was the simplified two dimensional model. Even though the system is capable of omnidirectional motion, this simplified model reduces the BallBot to the two dimensional planar systems on median planes in order to develop simple but stable controllers. This way, the analysis was simplified and the analytic complexity of controllers were decreased. Most importantly, the nonholonomic constraints associated with ground contact were reduced to holonomic ones, further simplifying the associated derivations[1].

[1]Holonomy of the 2D model will be discussed in Section 2.3

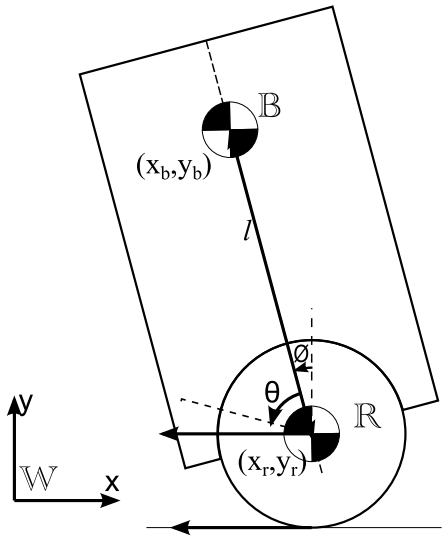

Figure 2.1: 2D BallBot model on the sagittal plane

As shown in Figure 2.1, the planar BallBot model on the sagittal plane consists of a rigid robot body attached to the center of a rigid rolling ball with an actuated rotary joint. The distance between centers of mass of the body and the ball is fixed at l. Three frames of reference are defined, an inertial world frame $\mathcal{W}$, the body frame $\mathcal{B}$ located at the center of mass of the body, and ball frame $\mathcal{R}$ located at the center of mass of the ball. The angle between the y-axis of $\mathcal{W}$ and the vertical body axis in $\mathcal{B}$ is defined to be ϕ. The angle θ is defined between the vertical axis of $\mathcal{R}$ and the vertical body axis of $\mathcal{B}$. In other words θ is defined such that $\phi + \theta$ is the angle between the y-axis of $\mathcal{W}$, and the vertical axis of $\mathcal{R}$. The position of the center of mass of the body in $\mathcal{W}$ is denoted with (x_b, y_b) and the position of the center of mass of the ball in $\mathcal{W}$ is denoted with (x_r, y_r).

Despite the disadvantageous constraints associated with this simple 2D model, such as its inability to model yawing behavior, it has been by far the most popular model in the literature for studying BallBot platforms and the stability of associated controllers.

Table 2.1: Parameters and Variables For the Free-Body Analysis of the Planar BallBot Model in Figure 2.2

Planar Ballbot Model Parameters	
m_b	Mass of the body
m_r	Mass of the ball
I_b	Inertia of the body
I_r	Inertia of the ball
g	Gravity Constant
l	The distance between center of mass of the body and center of $\mathcal{R}$
$\mathbf{r}_r$	The radius of the spherical ball
Planar Ballbot Model Variables	
x_b,y_b	Position of the body in x and y axes respectively in $\mathcal{W}$
x_r,y_r	Position of the ball in x and y axes respectively in $\mathcal{W}$
$\dot{x}_b$,$\dot{y}_b$	Linear velocity of the body in x and y axes respectively in $\mathcal{W}$
$\dot{x}_r$,$\dot{y}_r$	Linear velocity of the ball in x and y axes respectively in $\mathcal{W}$
$\ddot{x}_b$,$\ddot{y}_b$	Linear acceleration of the body in x and y axes respectively in $\mathcal{W}$
$\ddot{x}_r$,$\ddot{y}_r$	Linear acceleration of the ball in x and y axes respectively in $\mathcal{W}$
ϕ	Angle between the vertical axis of $\mathcal{B}$ and vertical axis (y axis) of inertial frame $\mathcal{W}$
θ	Angle between the vertical axis of $\mathcal{R}$ and the vertical axis of $\mathcal{B}$
$\dot{\phi}$	Angular velocity of the body with respect to $\mathcal{W}$
$\dot{\theta}$	Angular velocity of the ball with respect to body
$\ddot{\phi}$	Angular acceleration of the body with respect to $\mathcal{W}$
$\ddot{\theta}$	Angular acceleration of the ball with respect to body
τ	Input torque actuated around center of the ball in $\mathcal{W}$
$\mathbf{F}_x$,$\mathbf{F}_y$	Force applied to the body from ball
$\mathbf{F}_f$	Force applied to the ball from the ground in x axis reverse to the motion
$\mathbf{F}_{y,N}$	Normal force applied to the ball from ground

2.2 Basic Structure and Parameters

Even though there are many possible methods for deriving the dynamics of the 2D planar model, such as the Lagrangian formulation [11], we prefer to use free-body analysis on the simplified 2D model in order to better expose physical meanings of the constraints. Figure 2.2 illustrates a free-body decomposition of the system together with various forces and accelerations involved. Table 2.1 summarizes the parameters and state variables of the model.

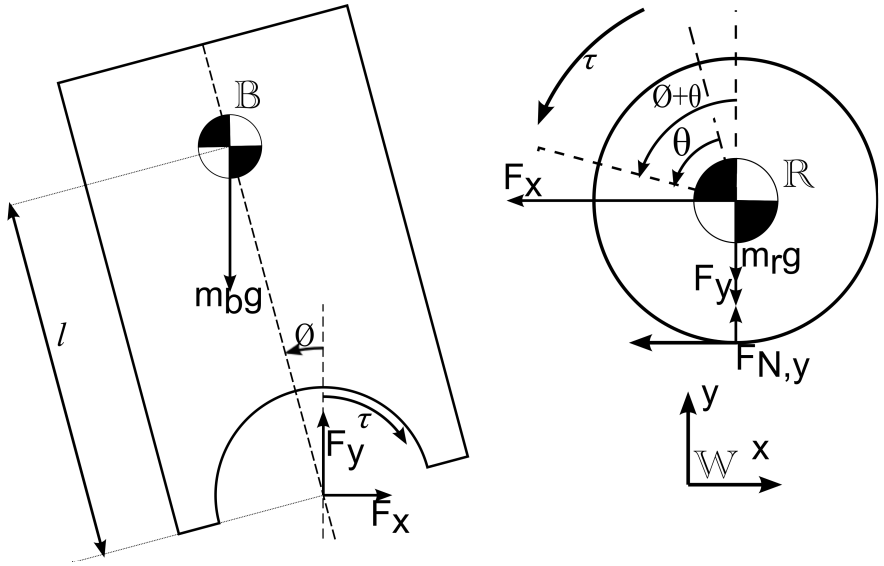

Figure 2.2: Free Body Analysis of 2D BallBot model on the sagittal plane

Choosing the generalized coordinate vector for the model to be $s := [\phi, \theta]$, we define the state of the system as

$$\mathbf{x} := \begin{bmatrix} \phi & \theta & \dot{\phi} & \dot{\theta} \end{bmatrix}^{\mathbf{T}} . \tag{2.1}$$

The first two equations for 2D model dynamics can be obtained through Newton's second law on the body for each x and y axes of the $\mathcal{W}$ as

$$m_b \ddot{x}_b = \mathbf{F}_x, \tag{2.2}$$

$$m_b \ddot{y}_b = \mathbf{F}_y - m_b g . \tag{2.3}$$

A similar derivation on the ball yields

$$m_r \ddot{x}_r = -\mathbf{F}_x - \mathbf{F}_f, \tag{2.4}$$

$$m_r \ddot{y}_r = -\mathbf{F}_y - m_r g + \mathbf{F}_{y,N} . \tag{2.5}$$

Torque balance on the rotational motion of the body results in the equation

$$I_b \ddot{\phi} = -\tau + \mathbf{F}_y l \sin\phi + \mathbf{F}_x l \cos\phi . \tag{2.6}$$

Since the body forces acting on the ball are applied to its center of mass, the rotational motion of the ball is captured by the equation

$$I_r(\ddot{\theta} + \ddot{\phi}) = \tau - \mathbf{F}_f \mathbf{r}_r . \tag{2.7}$$

In addition to these rigid body dynamics equations, the system has additional constraints as stated in [8]. The planar BallBot model assumes that there is no slip between the ball and the ground. Moreover, it is assumed that motion in the two median planes (median sagittal plane and median coronal plane) are decoupled and the equations of motion in these two planes are identical.

Planar Constraint 1: The planar model assumes that there is no-slip between the ball and the ground. Thus, the motion is pure rolling (see [7]) with the associated constraint equation taking the form

$$\dot{x}_r = -(\dot{\theta} + \dot{\phi})\mathbf{r}_r \ . \tag{2.8}$$

Taking the derivative of (2.8), the linear acceleration equation for the no-slip constraint is obtained as

$$\ddot{x}_r = -(\ddot{\theta} + \ddot{\phi})\mathbf{r}_r \ . \tag{2.9}$$

Moreover, since simplified model assumes that the motion of the ball is pure rolling, there should be no ball motion along the y axis of $\mathcal{W}$. For this reason, we consider a second equation for the no-slip constraint with

$$\ddot{y}_r = 0 \ . \tag{2.10}$$

■

Planar Constraint 2: In the 2D model, the ball center of mass must coincide with the joint location on the robot body. Due to the rigid body assumption, the distance between the center of mass of the ball and the center of mass of the body must hence remain constant at l and at the same angle relative to the body frame. In other words, the rotary joint constraint can be formulated in terms of the positions of the ball and the body as

$$\begin{bmatrix} x_b + l\sin\phi \\ y_b - l\cos\phi \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \end{bmatrix} \ . \tag{2.11}$$

Once we take the second derivative of (2.11), the linear acceleration equation for the rotary body-ball joint constraint can be written for both x and y axes in $\mathcal{W}$ as

$$\ddot{x_b} + \ddot{\phi} l \cos\phi - \dot{\phi}^2 l \sin\phi = \ddot{x}_r \tag{2.12}$$

$$\ddot{y_b} + \ddot{\phi} l \sin\phi + \dot{\phi}^2 l \cos\phi = \ddot{y}_r \ . \tag{2.13}$$

■

At this point, using the no-slip expression (2.9) in (2.12) we obtain

$$\ddot{x_b} = -(\ddot{\theta} + \ddot{\phi})\mathbf{r}_r - \ddot{\phi} l \cos\phi + \dot{\phi}^2 l \sin\phi \ . \tag{2.14}$$

Similarly for the y-axis, by using the no-slip expression (2.10) in (2.13) we obtain

$$\ddot{y_b} = -\ddot{\phi} l \sin\phi - \dot{\phi}^2 l \cos\phi \ . \tag{2.15}$$

For the ball force equation along the x-axis, we may reduce the equations by using (2.2) and (2.4) in (2.7) to yield

$$I_r \ddot{\theta} = \tau + (m_r \ddot{x}_r + m_b \ddot{x_b})\mathbf{r}_r - I_r \ddot{\phi} \ . \tag{2.16}$$

Moreover, after adding (2.9) and (2.14), the first equation takes its final form in terms of only ϕ and θ and their derivatives, taking the form

$$\begin{aligned}(I_r + (m_r + m_b)\mathbf{r}_r^2)\ddot{\theta} = \tau - \ddot{\phi}(I_r + (m_r + m_b)\mathbf{r}_r^2 + m_b \mathbf{r}_r l \cos\phi) \\ + \dot{\phi}^2 m_b \mathbf{r}_r l \sin\phi \ .\end{aligned} \tag{2.17}$$

Similar to the derivation of (2.17), we use (2.4),(2.5), (2.14) and (2.15) in (2.6) to obtain the second component in the equations of motion as

$$(I_b + m_b l^2)\ddot{\phi} = -\tau + m_b g l \sin\phi - m_b \ddot{\theta} \mathbf{r}_r l \cos\phi i - m_b \ddot{\phi} \mathbf{r}_r l \cos\phi \ . \tag{2.18}$$

As a result, combining (2.17) and (2.18), the equations of motion for the 2D BallBot model can be finalized as

$$\underbrace{\begin{bmatrix} a & a + c\cos\phi \\ c\cos\phi & b + c\cos\phi \end{bmatrix}}_{D(s)} \underbrace{\begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix}}_{\ddot{s}} + \underbrace{\begin{bmatrix} 0 & -\dot{\phi}c\sin\phi \\ 0 & 0 \end{bmatrix}}_{C(q,\dot{s})} \underbrace{\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \end{bmatrix}}_{\dot{s}} + \underbrace{\begin{bmatrix} 0 \\ -m_b g l \sin\phi \end{bmatrix}}_{G(s)} = \begin{bmatrix} \tau \\ -\tau \end{bmatrix} , \tag{2.19}$$

where, $a := (I_r + (m_r + m_b)\mathbf{r}_r^2)$, $b := (I_b + m_b l^2)$, and $c := m_b \mathbf{r}_r l$.

If the initial linear velocities of the ball and the body are identical and nonzero, and no external disturbances are present, this model will continue to move indefinitely without any dissipation. However, the real BallBot in such situations generally comes to a complete stop since inevitable damping is present in the system. For this reason, we will find it useful to incorporate a friction term into the equations of motion to model this behaviour and make the model more realistic. We accomplish this with a viscous damping constant D_v between the ball and the ground, modifying (2.17) as

$$a\ddot{\theta} + \ddot{\phi}(a + c\cos\phi) - \dot{\phi}^2 c\sin\phi = \tau - D_v\dot{\theta} \, . \tag{2.20}$$

As a result, by using (2.20) and (2.18), we obtain the following dissipative equations of motion for the 2D BallBot model as

$$\begin{bmatrix} a & a + c\cos\phi \\ c\cos\phi & b + c\cos\phi \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & -\dot{\phi}c\sin\phi \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ -m_b g l \sin\phi \end{bmatrix} = \begin{bmatrix} \tau - D_v\dot{\theta} \\ -\tau \end{bmatrix} , \tag{2.21}$$

where, a, b and c are as defined before.

As mentioned in the beginning of this section, the 2D model can be analyzed using a variety of methods. In earlier work, such as [1], [11], the Lagrangian method was used to obtain the same equations of motion as (2.19). Since this result also means that the system is integrable, as discussed in [14], the 2D planar model is a holonomic system. However, this result is not realistic since we know that the real BallBot system should be nonholonomic, similar to the example of spherical rolling ball [17]. This point will be discussed further in Section 2.3.

2.3 The Decoupled 2.5D BallBot Model

Even though the BallBot is capable of omnidirectional mobility, the 2D planar model is limited to linear trajectories in 3D space. Thus, in order to model BallBot motion on spatial trajectories, a "2.5D" model is generally used. The 2.5D model consists of combining two decoupled 2D planar model on the median sagittal planes of BallBot. This model is often assumed in the literature to be sufficiently accurate for representing basic, slow motions of the system. Not surprisingly, this is correct and satisfactory for linear, or almost linear trajectories. However, for faster nonlinear trajectories, it is likely to result in problems with predictive accuracy.

In this context, the first drawback of the 2.5D model is its inability to represent the natural yaw dynamics of the BallBot. Moreover, the 2.5D model will most likely be insufficient for fast and dynamic maneuvers, which are precisely maneuvers for which the BallBot was designed and conceptualized for. Additionally, since 2.5D model cannot represent yaw dynamics, predicting orientation of the actuation mechanism with respect to the ball, or the ball with respect to the ground is not possible within this model. This makes the modeling of interactions between such components difficult, and when we need to understand the BallBot behaviors such interaction models are very important. For the design of sufficiently accurate behavioral controllers, a mathematical model that can easily utilize the effects of such interactions would be essential. As it is the main purpose of the mobile robot platforms, thinking about the effects of adding any extensions to BallBot is important. If any extensions like arms [15] or asymmetric loads are added to the BallBot, the 2.5D model would become less accurate. Therefore, the availability of a fully coupled 3D model is a necessity for accurate behavioral controllers and practical use of the platform.

With these motivations, the development of a fully coupled 3D model for BallBot that is capable of accurately capturing all aspects of BallBot dynamics becomes necessary and will be presented in the following chapters.

Chapter 3

THE COUPLED 3D BALLBOT MODEL

As we observed in Section 2.3, controlling BallBot by using planar 2D models has some disadvantages, particularly for fast maneuvers and nonplanar motion. In order to overcome these issues, we will develop a fully coupled 3D model, which naturally incorporates body yaw dynamics as well as the behavior of nonholonomic ball rolling.

It is well known that rigid body rotations can be represented by using different coordinatizations such as Euler angles [18],[19], rotation matrices [20], [21] etc. (see [22] for a comparison and problems with representations of spatial rotation). In our work, we will use unit quaternions to represent rigid body rotations since they have the advantage of lower complexity, are free of singularities like the gimbal lock[1], and can easily be transformed into rotation matrices. In Section 3.1, we describe introductory background on quaternions that is used throughout the thesis. For more information, see e.g. [23],[24], [25],[26].

[1]See [23].

3.1 Quaternions: An Introduction

Simplistically, quaternions can be thought of as vectors in $\mathbb{R}^4$. Unit quaternions are quaternions with unit norm, namely Euclidian four-dimensional vectors of unit length. Consequently, unit quaternions can also be thought of as points on a unit hypersphere embedded in four dimensions, and have three degrees of freedom [23].

Unit quaternions have the property of capturing all of the geometry, topology and group structure of three dimensional rotations in the simplest possible way. They provide advantages in the complexity of computations compared to both Euler angles and rotation matrices, which are generally used alternatives for representing spatial rotations [27]. Quaternions do not suffer from singularities such as the gimbal lock, which is a particularly troublesome problem for Euler angles. They also allow efficient interpolation of orientation frames [26]. Moreover, quaternions can easily be transformed into rotation matrices, even though the reverse transformation is ambiguous and admits two possible unit quaternions. For all these reasons, we will find it best to represent 3D orientations by using unit quaternions in this thesis.

More formally, a unit quaternion q is defined as

$$q := \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \quad \text{where} \qquad q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \; . \tag{3.1}$$

A quaternion can also be decomposed into a constant q_0 and a three dimensional vector $v = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T$. Real numbers can be represented as quaternions[2] as being the scalar part,such as $\begin{bmatrix} q_0 & 0 & 0 & 0 \end{bmatrix}^T$. Similarly, vectors in 3D can also be represented as quaternions[3] without scalars, like $\begin{bmatrix} 0 & q_1 & q_2 & q_3 \end{bmatrix}^T$ or $(0, \mathbf{v})$.

[2]Quaternions with only scalar part are not unit quaternions unless their norm is 1.

[3]Similar to the scalar counterpart, unless unit vectors represented as quaternion, mentioned representation do not results with unit quaternions

Now consider the quaternions p and q defined as

$$q := \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T, \quad p := \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \end{bmatrix}^T . \tag{3.2}$$

Quaternion addition is similar to vector addition, consisting of adding individual coordinates, defined as

$$q + p := \begin{bmatrix} p_0 + q_0 & p_1 + q_1 & p_2 + q_2 & p_3 + q_3 \end{bmatrix}^T . \tag{3.3}$$

Conjugation of a quaternion is defined as

$$q^* := \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix}^T . \tag{3.4}$$

Note that the conjugation of a quaternion is also used to define the "inverse" of the quaternion scaled with the norm of that quaternion, i.e. $q^{-1} := q^* / \|q\|$. Thus for unit quaternions, the conjugate and inverse are the same with $q^* := q^{-1}$.

A multiplication operation $\circ$ for two quaternions p and q is defined as

$$p \circ q := \begin{bmatrix} p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3 \\ p_1 q_0 + p_0 q_1 + p_2 q_3 - p_3 q_2 \\ p_2 q_0 + p_0 q_2 + p_3 q_1 - p_1 q_3 \\ p_3 q_0 + p_0 q_3 + p_1 q_2 - p_2 q_1 \end{bmatrix} . \tag{3.5}$$

Note that quaternion multiplication is not commutative but associative. The equation given by (3.5) can conveniently be transformed into a matrix multiplication by defining the matrices P and Q, which are orthogonal[4] and associated with quaternions p and q as

$$P := \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix}, \quad Q =: \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} . \tag{3.6}$$

By using (3.5) and (3.6), it trivially follows that quaternion multiplication can be represented as

$$p \circ q := Pq = Qp. \tag{3.7}$$

[4]Note that for the quaternions without scalar parts these matrices are skew-symmetric.

Note, also, that for unit quaternions we have

$$q \circ q^{-1} = p \circ p^* = e \ , \tag{3.8}$$

where $e := [1\ 0\ 0\ 0]^T$ is the multiplicative identity, also called the scalar unit quaternion.

Both quaternion multiplication matrices have a particular structure which is helpful to understand quaternion multiplication, given by

$$P = \begin{bmatrix} p_0 & -\mathbf{p}^T \\ \mathbf{p} & \mathbf{I}_3 p_0 + \Omega_{\mathbf{p}} \end{bmatrix}, \tag{3.9}$$

$$Q = \begin{bmatrix} q_0 & -\mathbf{q}^T \\ \mathbf{q} & -\Omega_q + \mathbf{I}_3 q_0 \end{bmatrix}, \tag{3.10}$$

where Ω_v denotes the skew-symmetric matrix representing vector cross product in matrix form for v as the first vector, namely $a \times b = \Omega_a b$. For a vector $a = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$, Ω_a takes the form

$$\Omega_a := \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} . \tag{3.11}$$

Finally, rotation matrices can be defined as a function of a unit quaternion q, which we denote with $R(\mathbf{q})$[5]. As mentioned before, any three dimensional vector can have a quaternion form, which we denote in this thesis with underlined vector symbols as

$$\underline{v} := \begin{bmatrix} 0 & v_1 & v_2 & v_3 \end{bmatrix}^T . \tag{3.12}$$

3.2 Basic Model Structure and Parameters

The new 3D BallBot model we introduce is shown in Figure 3.1 with various parameters in Tables 3.1 and 3.2. Similar to the 2D case, this model consists of

[5]For more information about the derivation of $R(\mathbf{q})$, please refer to [28].

Table 3.1: Parameters For the Free-Body Analysis of the 3D BallBot Model

3D BallBot Model Coordinate Frames	
$\mathcal{W}$	Inertial world frame
$\mathcal{B}$	Body frame located at the center of mass of the robot body
$\mathcal{R}$	Ball frame located at the center of mass of the rolling ball
3D BallBot Model Parameters	
m_b	Mass of the body
m_r	Mass of the ball
$\mathbf{I}_b$	Inertia matrix of the body in $\mathcal{B}$
$\mathbf{I}_r$	Inertia matrix of the ball in $\mathcal{R}$
g	Gravity Constant
$\mathbf{g}_w$	Gravity Constant in vector form $g_w = [0, 0, -g]^T$ in $\mathcal{W}$
l	The distance between center of mass of the body and center of $\mathcal{R}$
$\mathbf{r}_r$	The radius of the spherical ball
$\mathbf{p}_{b,\mathcal{B}}$	The position vector pointing from ball COM to body COM in $\mathcal{B}$
d_b	The position vector pointing from ball COM to body COM in $\mathcal{W}$
$\mathbf{r}_w$	The position vector pointing from ball COM to ground in $\mathcal{W}$
D_v	Viscous damping constant between the ball and the ground
$D_{v,p}$	Viscous damping constant in matrix form as $D_{v,p} = diag[-D_v, -D_v, 0]$

two rigid bodies connected with a spherical joint. The main body of the BallBot is a rigid body with mass m_b and inertia matrix $\mathbf{I}_b$. It is connected to the center of the roller body, a spherical ball with mass m_r, inertia matrix $\mathbf{I}_r$, and radius $\mathbf{r}_r$, through a spherical joint[6], actuated with a command torque vector τ. The directions in which τ is applied are constant in $\mathcal{B}$, the body frame located at the center of mass of the robot body and oriented in stationary alignment with the body. In addition to $\mathcal{B}$, we define $\mathcal{W}$ as an inertial world frame and $\mathcal{R}$ as the ball frame located at the center of the rolling ball. Note that the distance between $\mathcal{R}$ and $\mathcal{B}$ is fixed at l. Thus the position of the COM of the roller in $\mathcal{B}$ is always constant as $[0, 0, -l]^T$. The position of the COM for the body and the ball in $\mathcal{W}$ are represented with $\mathbf{p}_b$ and $\mathbf{p}_r$, respectively. The ground contact point for the ball in $\mathcal{W}$ is denoted with $\mathbf{p}_c$. Quaternion orientations for the body and the ball are $\mathbf{q}_b$ and $\mathbf{q}_r$, respectively.

[6]See Section 1.1 for a discussion of different actuation mechanisms for this joint.

Table 3.2: Variables For the Free-Body Analysis of the 3D BallBot Model

3D BallBot Model Variables	
τ	Input torque actuated around center of the ball defined in $\mathcal{W}$
τ_b	Input torque actuated around center of the ball defined in $\mathcal{B}$
$\tau_{r,z}$	Compensator torque actuated around center of the ball defined in $\mathcal{W}$,compensating for ball yaw constraint
$\tau_{b,z}$	Compensator torque actuated around center of the ball defined in $\mathcal{W}$,compensating for IMD constraint
$\tau_{b,IMD}$	Input torque of IMD model actuated around center of the ball defined in $\mathcal{B}$, for the pitch and roll axes
$\mathbf{p}_b$, $\mathbf{p}_r$	Position vector of the center of mass of the body and in $\mathcal{W}$ respectively
$\mathbf{p}_c$	Position vector of the ground contact point of the ball in $\mathcal{W}$
$\mathbf{q}_b$, $\mathbf{q}_r$	Quaternion representation of orientation for the body and ball respectively
$\dot{\mathbf{p}}_\mathbf{b}$, $\ddot{\mathbf{p}}_\mathbf{b}$	Linear velocity and acceleration vector of the body in $\mathcal{W}$ respectively
$\dot{\mathbf{p}}_\mathbf{r}$, $\ddot{\mathbf{p}}_\mathbf{r}$	Linear velocity and acceleration vector of ball in $\mathcal{W}$ respectively
$\mathbf{P}_b$, $\mathbf{P}_r$	Linear Momentum vector of the body and ball in $\mathcal{W}$ respectively
$\dot{\mathbf{P}}_\mathbf{b}$,$\dot{\mathbf{P}}_\mathbf{r}$	First derivative of linear momentum vector of the body and the ball in $\mathcal{W}$ respectively
$\mathbf{w}_b$, $\dot{\mathbf{w}}_\mathbf{r}$	Angular velocity and acceleration vector of the body in $\mathcal{W}$ respectively
$\mathbf{w}_b$, $\dot{\mathbf{w}}_\mathbf{r}$	Angular velocity and acceleration vector of the ball in $\mathcal{W}$ respectively
$\mathbf{L}_b$, $\mathbf{L}_r$	Angular Momentum vector of the body and ball in $\mathcal{W}$ respectively
$\dot{\mathbf{L}}_\mathbf{b}$, $\dot{\mathbf{L}}_\mathbf{r}$	First derivative of angular momentum vector of the body and ball in $\mathcal{W}$ respectively
$\mathbf{x}$	State vector of 3D model
$\dot{\mathbf{x}}$	First derivative of state vector of 3D model
$\mathbf{U}_u$,$\mathbf{U}_d$, $\mathbf{U}_r$,$\mathbf{U}_{IMD}$	Unknown vectors that contains unknown accelerations and constraints of their models respectively
M_u,M_d M_r,M_{IMD}	Unknown matrices for the equations of motion of their models respectively
N_u,N_d N_r,N_{IMD}	Constant vectors for the equations of motion of their models respectively
$\mathbf{F}_b$	Constraint force vector applied to the body by ball in $\mathcal{W}$
$\mathbf{F}_r$	Constraint force vector applied to the ball from the ground in $\mathcal{W}$

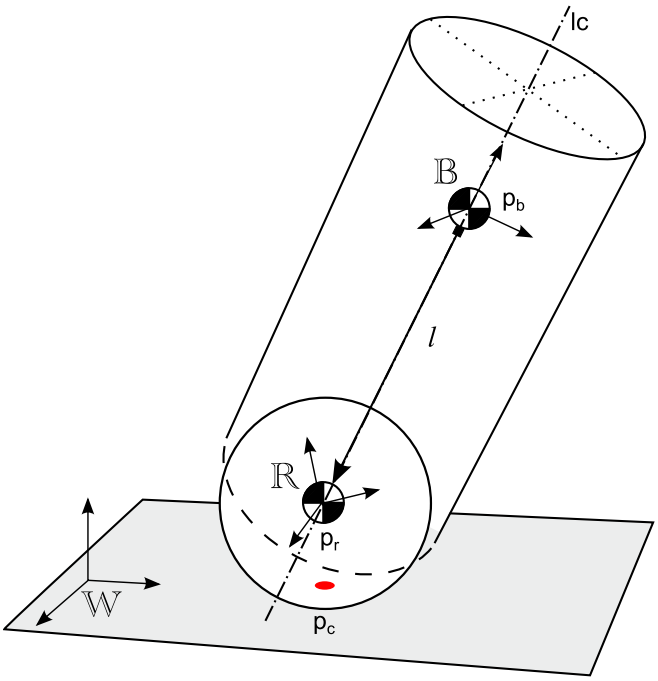

Figure 3.1: Coupled 3D BallBot model

Unlike the 2D model, the no-slip constraint for the 3D model cannot be reduced to a holonomic constraint. Consequently, direct use of the Lagrangian method for deriving the equations of motion is slightly more complex. Moreover, the incorporation of more complex surface interaction and friction models within such a derivation would be more problematic. Consequently, we prefer to use free-body diagram to obtain the equations of motion. This way, the reason behind various constraints added through the analysis can also be seen more clearly.

To begin the free-body analysis, we define the unconstrained state of the system as the position, orientation, linear and angular momenta of both the body and the ball, yielding

$$\mathbf{x} := \begin{bmatrix} \mathbf{p}_b & \mathbf{P}_b & \mathbf{q}_b & \mathbf{L}_b & \mathbf{p}_r & \mathbf{P}_r & \mathbf{q}_r & \mathbf{L}_r \end{bmatrix}^T . \tag{3.13}$$

In addition to the positions $\mathbf{p}_b$, $\mathbf{p}_r$ and orientations $\mathbf{q}_b$, $\mathbf{q}_r$, the linear and angular momenta in $\mathcal{W}$, $\mathbf{P}$ and $\mathbf{L}$ respectively, are also included in the state vector for both the body and the ball.

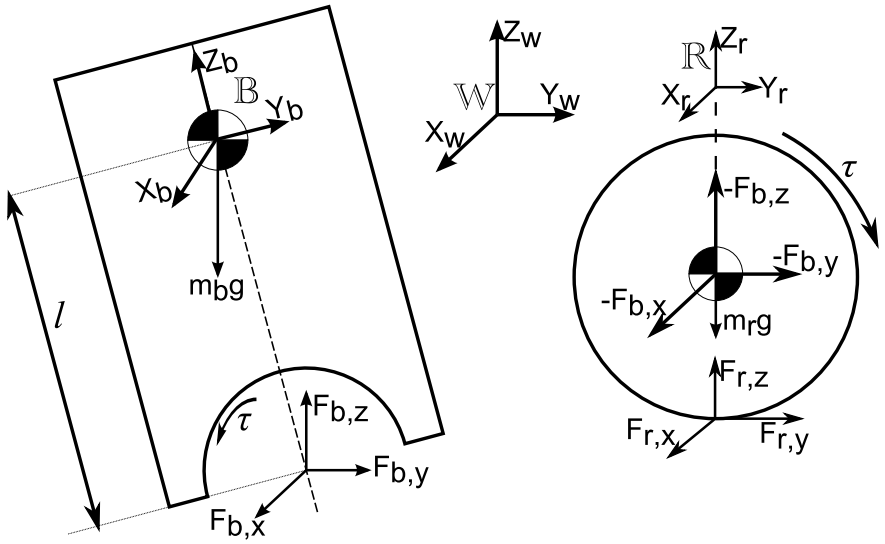

Figure 3.2: Free Body Analysis of 3D BallBot model

First two of the rigid body dynamic equations for the unconstrained model result from the application of the Newton's second law to each body for positional coordinates, taking the general form

$$\dot{\mathbf{P}} = \mathbf{F} + [0, 0, -mg]^T. \tag{3.14}$$

For the robot body, the linear force equation reads as

$$\dot{\mathbf{P}}_\mathbf{b} = m_b \ddot{\mathbf{p}}_\mathbf{b} = m_b \mathbf{g}_w + \mathbf{F}_b, \tag{3.15}$$

where, we define the gravity vector $\mathbf{g}_w := [0, 0, -g]^T$. Similarly, the force balance equations for the ball is

$$m_r \ddot{\mathbf{p}}_\mathbf{r} = m_r g_w + \mathbf{F}_r - \mathbf{F}_b \ . \tag{3.16}$$

The moment balance for the body yields the equation

$$\dot{\mathbf{L}}_\mathbf{b} = \tau + (-d_b) \times \mathbf{F}_b \quad , \tag{3.17}$$

where τ is the torque input and d_b is the position vector of the body[7] with respect to the attachment point of the body and ball in $\mathcal{W}$.

Similarly, the moment balance equation for the ball takes the form

$$\dot{\mathbf{L}}_\mathbf{r} = -\tau + (\mathbf{r}_w) \times \mathbf{F}_r \ , \tag{3.18}$$

[7]Constraint 1 is used in (3.21) as $\mathbf{p}_{body,contact} - \mathbf{p}_{body,CM} = \mathbf{p}_r - \mathbf{p}_b = -d_b$.

where $\mathbf{r}_w$ is the position of the ball ground contact point with respect to its center, always remaining constant with $\mathbf{r}_w := [0, 0, -\mathbf{r}_r]^T$ in $\mathcal{W}$.

The relation between body angular velocity and angular momentum [29] gives us the relation between angular acceleration and angular moment as

$$\dot{\mathbf{w}}_\mathbf{b} = -{\mathbf{I}_b}^{-1}\Omega_{\mathbf{w}_b}\mathbf{L}_b + {\mathbf{I}_b}^{-1}\dot{\mathbf{L}}_\mathbf{b} \ . \tag{3.19}$$

The same relation can also be given for the ball as

$$\dot{\mathbf{w}}_\mathbf{r} = -{\mathbf{I}_r}^{-1}\Omega_{\mathbf{w}_r}\mathbf{L}_r + {\mathbf{I}_r}^{-1}\dot{\mathbf{L}}_\mathbf{r} \ . \tag{3.20}$$

3.2.1 Basic BallBot Model without Yaw Constraints

Similar to the planar BallBot model, the unconstrained 3D BallBot model has two additional constraints that complement the free rigid body dynamics.

Constraint 1: As before, the position of the COM of the roller must coincide with the ball-body joint. More formally, we should have

$$\underline{\mathbf{p}_b} = \underline{\mathbf{p}_r} + \mathbf{q}_b \circ \underline{\mathbf{p}_{b,\mathcal{B}}} \circ \mathbf{q}_b^* \ , \tag{3.21}$$

where $\mathbf{p}_{b,\mathcal{B}} := [0, 0, l]^T$ is the position vector pointing from ball COM to body COM in $\mathcal{B}$. Using this equation[8], we obtain

$$\begin{aligned}
\underline{\ddot{\mathbf{p}}_\mathbf{b}} &= \underline{\ddot{\mathbf{p}}_\mathbf{r}} + (\mathbf{q}_b \circ \underline{\ddot{\mathbf{p}}_{b,\mathcal{B}}} \circ \mathbf{q}_b^*). && (3.22)\\
&= \underline{\ddot{\mathbf{p}}_\mathbf{r}} + \frac{1}{2}(\underline{\dot{\mathbf{w}}_\mathbf{b}} \circ \mathbf{q}_b \circ \underline{\mathbf{p}_{b,\mathcal{B}}} \circ \mathbf{q}_b^* + \mathbf{q}_b \circ \underline{\mathbf{p}_{b,\mathcal{B}}} \circ \mathbf{q}_b^* \circ \underline{\dot{\mathbf{w}}_\mathbf{b}}^*) && (3.23)\\
&+ \frac{1}{4}\underline{\mathbf{w}_b} \circ \underline{\mathbf{w}_b} \circ d_b + \frac{1}{2}\underline{\mathbf{w}_b} \circ d_b \circ \underline{\mathbf{w}_b}^* + \frac{1}{4} d_b \circ \underline{\mathbf{w}_b}^* \circ \underline{\mathbf{w}_b}^* \quad . && (3.24)
\end{aligned}$$

[8]Note that $d_b = \mathbf{q}_b \circ \underline{\mathbf{p}_{b,\mathcal{B}}} \circ \mathbf{q}_b^*$ by definition.

For quaternion representation of vectors, we can use (3.9) and (3.10) to derive the following simplifications

$$\underline{v} \circ q + q \circ \underline{v}^* = P_v q + Q_{\underline{v}^*} q \tag{3.25}$$

$$= \begin{bmatrix} 0 & -\mathbf{v}^T \\ \mathbf{v} & \Omega_{\mathbf{v}} \end{bmatrix} q + \begin{bmatrix} 0 & \mathbf{v^T} \\ -\mathbf{v} & \Omega_{\mathbf{v}} \end{bmatrix} q \tag{3.26}$$

$$= 2 \begin{bmatrix} 0 & 0 \\ 0 & \Omega_v \end{bmatrix} q \quad . \tag{3.27}$$

Using (3.27), we obtain

$$\underline{\ddot{\mathbf{p}}_\mathbf{b}} = \underline{\ddot{\mathbf{p}}_\mathbf{r}} + \begin{bmatrix} 0 & 0_{1\times 3} \\ 0_{3\times 1} & \Omega_{\dot{\mathbf{w}}_\mathbf{b}} \end{bmatrix} d_b$$

$$+ \frac{1}{2} \begin{bmatrix} 0 & 0_{1\times 3} \\ 0_{3\times 1} & \Omega_{\mathbf{w}_b} \end{bmatrix} (\mathbf{w}_b \circ d_b + d_b \circ \mathbf{w}_b^*), \tag{3.28}$$

$$= \underline{\ddot{\mathbf{p}}_\mathbf{r}} + \begin{bmatrix} 0 & 0_{1\times 3} \\ 0_{3\times 1} & \Omega_{\dot{\mathbf{w}}_\mathbf{b}} \end{bmatrix} d_b + \begin{bmatrix} 0 & 0_{1\times 3} \\ 0_{3\times 1} & \Omega_{\mathbf{w}_b} \end{bmatrix} \begin{bmatrix} 0 & 0_{1\times 3} \\ 0_{3\times 1} & \Omega_{\mathbf{w}_b} \end{bmatrix} d_b, \tag{3.29}$$

$$= \underline{\ddot{\mathbf{p}}_\mathbf{r}} + \begin{bmatrix} 0 & 0_{1\times 3} \\ 0_{3\times 1} & \Omega_{\dot{\mathbf{w}}_\mathbf{b}} \end{bmatrix} d_b + \begin{bmatrix} 0 & 0_{1\times 3} \\ 0_{3\times 1} & \Omega_{\mathbf{w}_b}\Omega_{\mathbf{w}_b} \end{bmatrix} d_b. \tag{3.30}$$

Thus, the body and ball attachment constraint simplifies to

$$\ddot{\mathbf{p}}_\mathbf{b} - \ddot{\mathbf{p}}_\mathbf{r} + \Omega_{d_b} \dot{\mathbf{w}}_\mathbf{b} = \Omega_{\mathbf{w}_b}\Omega_{\mathbf{w}_b} d_b \ . \tag{3.31}$$

■

Constraint 2: Similar to the planar model, the 3D model also assumes that the ball undergoes pure rolling motion. In other words, there is a no-slip constraint between the ball and the ground, which can be formulated as

$$\dot{\mathbf{p}}_\mathbf{r} = \Omega_{r_w} \mathbf{w}_r \ . \tag{3.32}$$

By taking the derivative of this constraint, we obtain the constraint equation to be used in solving for the unknown forces and accelerations as

$$\ddot{\mathbf{p}}_\mathbf{r} = \Omega_{r_w} \dot{\mathbf{w}}_\mathbf{r} \ . \tag{3.33}$$

For further details on the derivation of this constraint equation, see [30].

■

In order to obtain the equations of motion for the 3D BallBot, we must solve for the unknown forces and accelerations in the system. We start by defining a vector of unknown quantities, $\mathbf{U}_u$, containing unknown accelerations and constraints of the system as

$$\mathbf{U}_u := \begin{bmatrix} \dot{\mathbf{P}}_\mathbf{b} & \dot{\mathbf{L}}_\mathbf{b} & \dot{\mathbf{w}}_\mathbf{b} & \dot{\mathbf{P}}_\mathbf{r} & \dot{\mathbf{L}}_\mathbf{r} & \dot{\mathbf{w}}_\mathbf{r} & \mathbf{F}_b & \mathbf{F}_r \end{bmatrix}^T . \tag{3.34}$$

Here, $\mathbf{F}_b$ and $\mathbf{F}_r$ are constraint forces applied by the ball to the body, and by the ground to the ball, respectively. $\dot{\mathbf{w}}_\mathbf{b}$ and $\dot{\mathbf{w}}_\mathbf{r}$, are the angular accelerations of the body and the ball in $\mathcal{W}$, respectively. The equations we presented in the previous section are all linear in the quantities collected in $\mathbf{U}_u$. Consequently, we can form a linear system of equations as a function of the unknown vector, taking the form

$$\mathbf{M}_u \mathbf{U}_u = \mathbf{N}_u \, , \tag{3.35}$$

where $\mathbf{M}_u$, is a multiplier matrix capturing the system of equations introduced above and $\mathbf{N}_u$ is the associated constant vector.

Assuming that $\mathbf{M}_u$ is invertible, which is always true for the BallBot mechanical system in the absence of problematic model components such as Coulomb friction, the solution for the $\mathbf{U}_u$ vector can be calculated as $\mathbf{U}_u = \mathbf{M}_u^{-1}\mathbf{N}_u$. At this point, using the derived equations of rigid body dynamics (3.15)-(3.20), and first constraint (3.31), and second constraint (3.33), we can rewrite the equations of motion in the form given by (3.34).

Once the unknown vector (3.34) is solved for, the equations of motion for the 3D BallBot system takes the form

$$\dot{\mathbf{x}} = \begin{bmatrix} \frac{1}{m_b}\mathbf{P}_b & m_b\ddot{\mathbf{p}}_\mathbf{b} & \frac{1}{2}\underline{\mathbf{w}_b} \circ \underline{\mathbf{q}_b} & \dot{\mathbf{L}}_\mathbf{b} & \frac{1}{m_r}\mathbf{P}_r & m_r\ddot{\mathbf{p}}_\mathbf{r} & \frac{1}{2}\underline{\mathbf{w}_r} \circ \underline{\mathbf{q}_r} & \dot{\mathbf{L}}_\mathbf{r} \end{bmatrix}^T . \tag{3.36}$$

It can be seen from (3.36) that, with τ as a three dimensional input to the system, the form of the equations of motion for the 3D unconstrained model is

$$\dot{\mathbf{x}} = f_u(x, \tau) \; . \tag{3.37}$$

With the 3D unconstrained model, if the initial linear velocity of the ball and the body are the same, and if the system is stable without external disturbances present, the model will continue to move indefinitely without any energy dissipation. However, the physical BallBot generally comes to a complete stop in such situations since there is additional dissipation within the system. This justifies the need for a friction term to be added to the systems dynamics. In order to make the system more realistic, and to support the no slip constraint of the ball-ground interaction, we add viscous damping to the model, between ball and the ground in both roll and pitch directions.

The vector of unknown quantities does not need to change for adding the damping term. Consequently, we simply define a new symbol for clarity as

$$\mathbf{U}_d := \mathbf{U}_u \; . \tag{3.38}$$

The only important change occurs in (3.18), which now takes the form

$$\dot{\mathbf{L}}_{\mathbf{r}} - (\mathbf{r}_w) \times \mathbf{F}_r = -R_b \tau_b + D_{v,p} \mathbf{w}_b \; , \tag{3.39}$$

where $D_{v,p}$ is the damping matrix and is given by

$$D_{v,p} = \begin{bmatrix} -D_v & 0 & 0 \\ 0 & -D_v & 0 \\ 0 & 0 & 0 \end{bmatrix} \; . \tag{3.40}$$

By changing (3.18) to (3.39), the equations of motion keep the same form $\dot{\mathbf{x}} = f_d(x, \tau_b)$ but with f_d computed based on the system of equations $\mathbf{M}_d \mathbf{U}_d = \mathbf{N}_d$ computed with the new set of equations. Note that we define the input τ_b in $\mathcal{B}$ for both compatibility with physical BallBot platforms and allow integration of other model components such as actuator friction.

3.2.2 BallBot Model with Yaw Constraint on Ball

Both unconstrained model and damping model do not deal with a ball-ground contact model. Thus, behavior of this contact is independent from our 3D model. When there is any angular yaw velocity on the ball, this yaw velocity is kept inside model dynamics without affecting the system. Since such a case is not possible to happen in the real platform, it is reasonable to stop the yaw rotation of the ball. In order to stop the yaw rotation of the ball, instead of adding another viscous damping component, we may add a compensator torque $\tau_{r,z}$ as a constraint in $\mathcal{W}$, which sets angular yaw velocity of the ball to zero. We introduce the following constraint to address this problem.

Constraint 3: The vertical component of the ball's angular velocity must always be zero. Thus, ball yaw constraint is defined as,

$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{w}_b = 0 \, . \tag{3.41}$$

■

Based on this constraint and the addition of a compensator torque $\tau_{r,z}$ to (3.39), we obtain

$$\dot{\mathbf{L}}_{\mathbf{r}} - (\mathbf{r}_w) \times \mathbf{F}_r - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tau_{r,z} = -R_b \tau_b + D_{v,p} \mathbf{w}_b \, . \tag{3.42}$$

The addition of the new torque component $\tau_{r,z}$ requires a new vector of unknowns for the torque constrained model, defined as

$$\mathbf{U}_r := \begin{bmatrix} \dot{\mathbf{P}}_{\mathbf{b}} & \dot{\mathbf{L}}_{\mathbf{b}} & \dot{\mathbf{w}}_{\mathbf{b}} & \dot{\mathbf{P}}_{\mathbf{r}} & \dot{\mathbf{L}}_{\mathbf{r}} & \dot{\mathbf{w}}_{\mathbf{r}} & \mathbf{F}_b & \mathbf{F}_r & \tau_{r,z} \end{bmatrix}^T . \tag{3.43}$$

By changing (3.39) to (3.42), equations of motion keep the same form $\dot{\mathbf{x}} = f_r(x, \tau_b)$, but with f_r computed based on the new system of equations $\mathbf{M}_d \mathbf{U}_d = \mathbf{N}_d$. As we add (3.41) to the equations of motion of the system, $\mathbf{M}_r$ and $\mathbf{N}_r$ can

be formed from updated equations of motion, and with the new unknown vector $\mathbf{U}_r$ resulting system of equations can be given as

$$\mathbf{M}_r\mathbf{U}_r = \mathbf{N}_r \ . \tag{3.44}$$

Assuming that $\mathbf{M}_r$ is invertible, as expressed in Section 3.2.1, the solution for the $\mathbf{U}_r$ vector can be calculated as $\mathbf{U}_r = \mathbf{M}_r^{-1}\mathbf{N}_r$.

3.2.3 The Inverse Mouse-ball Drive BallBot Model

The last constraint is added to support the first instantiation of BallBot platforms, where the input τ has only two actuated components in $\mathcal{B}$ along the pitch and roll axes. In other words, we have the input vector $\tau_{b,IMD} = \begin{bmatrix}\tau_{bx} & \tau_{by} & 0\end{bmatrix}^T$. Since the torque in the yaw axis of $\mathcal{B}$ is not actuated with the *Inverse Mouseball Drive* (IMD) and is not allowed to slip sideways, the yaw angular velocity of the body in $\mathcal{B}$ is dictated by the yaw angular velocity of the ball in $\mathcal{B}$. We capture this property of the IMD with the following constraint:

Constraint 4: The relative angular yaw velocity of the body with respect to ball in $\mathcal{B}$ must be zero, as captured by the constraint

$$\begin{bmatrix}0 & 0 & 1\end{bmatrix} R_b^T(\mathbf{w}_b - \mathbf{w}_r) = 0. \tag{3.45}$$

By taking derivative of this equation, we obtain a new constraint on the set of unknowns for the IMD constrained model as

$$0 = \begin{bmatrix}0 & 0 & 1\end{bmatrix} (\dot{R_b}^T(\mathbf{w}_b - \mathbf{w}_r) + R_b^T(\dot{\mathbf{w}}_\mathbf{b} - \dot{\mathbf{w}}_\mathbf{r})) \tag{3.46}$$

$$= \begin{bmatrix}0 & 0 & 1\end{bmatrix} (-{R_b}^T\Omega_{\mathbf{w}_b}(\mathbf{w}_b - \mathbf{w}_r) + R_b^T(\dot{\mathbf{w}}_\mathbf{b} - \dot{\mathbf{w}}_\mathbf{r})) \ . \tag{3.47}$$

Defining ${R_{bz}}^T := \begin{bmatrix}0 & 0 & 1\end{bmatrix} R_b^T$, the IMD constraint equation becomes

$${R_{bz}}^T\dot{\mathbf{w}}_\mathbf{b} - {R_{bz}}^T\dot{\mathbf{w}}_\mathbf{r} = {R_{bz}}^T\Omega_{\mathbf{w}_b}(\mathbf{w}_b - \mathbf{w}_r) \ . \tag{3.48}$$

■

Note, however, that the unactuated torque in the yaw axis of $\mathcal{B}$ is still needed as the compensator torque $\tau_{b,z}$ for the IMD constraint but becomes an unknown quantity. When the compensator torque $\tau_{b,z}$ is explicitly shown in (3.42) and (3.17) respectively, we obtain two new equations for the dynamic model

$$\dot{\mathbf{L}}_{\mathbf{r}} - (\mathbf{r}_w) \times \mathbf{F}_r + R_b \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tau_{b,z} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tau_{r,z} = -R_b \tau_{b,IMD} + D_{v,p} \mathbf{w}_b \,, \tag{3.49}$$

$$\dot{\mathbf{L}}_{\mathbf{b}} + (d_b) \times \mathbf{F}_b - R_b \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tau_{b,z} = R_b \tau_{b,IMD} \,. \tag{3.50}$$

After adding $\tau_{b,z}$, a new vector of unknown quantities, $\mathbf{U}_{IMD}$, is defined for the torque constrained model as

$$\mathbf{U}_{IMD} := \begin{bmatrix} \dot{\mathbf{P}}_{\mathbf{b}} & \dot{\mathbf{L}}_{\mathbf{b}} & \dot{\mathbf{w}}_{\mathbf{b}} & \dot{\mathbf{P}}_{\mathbf{r}} & \dot{\mathbf{L}}_{\mathbf{r}} & \dot{\mathbf{w}}_{\mathbf{r}} & \mathbf{F}_b & \mathbf{F}_r & \tau_{r,z} & \tau_{b,z} \end{bmatrix}^T . \tag{3.51}$$

By changing (3.42) to (3.49) and (3.17) to (3.50), equations of motion preserve the same form $\dot{\mathbf{x}} = f_{IMD}(x, \tau_{b,IMD})$ with f_{IMD} computed from the new system of equations. Once we add (3.48) to the equations of motion of the system, $\mathbf{M}_{IMD}$ and $\mathbf{N}_{IMD}$ can be formed from updated equations as

$$\mathbf{M}_{IMD} \mathbf{U}_{IMD} = \mathbf{N}_{IMD}. \tag{3.52}$$

Assuming that $\mathbf{M}_{IMD}$ is invertible, the solution for the $\mathbf{U}_{IMD}$ vector can be calculated as $\mathbf{U}_{IMD} = \mathbf{M}_{IMD}^{-1} \mathbf{N}_{IMD}$.

3.3 Simulation Environment for 3D Models

Our simulations for the following sections and chapters are based on numerical integration of different forms of the 3D BallBot equations of motion. In each case, we solve the system of equation for the unknown forces and accelerations $\mathbf{U}$ and then compute the derivative of the system state. We used ode45 of MATLAB

for numerical integration with relative tolerance of 10^{-5} and maximum time step of $10^{-1}s$ to ensure accuracy.

In order to make sure that the constraints on accelerations defined in Section 3.2.1, Section 3.2.2 and Section 3.2.3 do not drift in time due to the numerical integration errors, we periodically reset them to the closest possible state that satisfies the constraints once every second in simulated time. Moreover, we normalize quaternions in the state vector to ensure they have unit norm within each evaluation step to prevent other numerical problems due to ways in which ode45 computes derivatives of the state. As a result of these countermeasures, constraint errors never grew beyond 10^{-10} in magnitude in our simulations.

Table 3.3: Kinematic and dynamic parameters in MKS units for BallBot simulations, chosen to be compatible with [1]

m_b	$\mathbf{I}_b^{xx}$	$\mathbf{I}_b^{yy}$	$\mathbf{I}_b^{zz}$	m_r	$\mathbf{I}_r$	l	$\mathbf{r}_r$
51.66	12.59	12.48	0.66	2.44	0.018	0.69	0.106

In all our simulations in subsequent sections, we use the dynamic and kinematic parameters given in Table 3.3, which are chosen based on the experimental BallBot presented in [1].

3.4 Planar 3D Model Trajectories Verified

In theory, the 3D model operating on linear trajectories must reduce to the 2D model on the associated sagittal plane. In this section, we use this fact to partially verify the correctness of our 3D model. Figure 3.3 illustrates an example comparison between the body attitude differences of the 2D planar model and our new 3D model without resetting. As the trajectory chosen for the example is a linear path, the resulting body attitude trajectories for 2D and 3D models are the same. This is expected since yaw dynamics are never excited and constitute

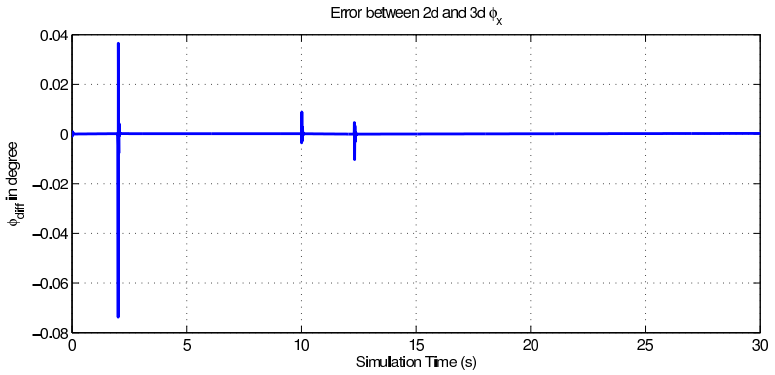

Figure 3.3: The difference in the attitude trajectories of the 2D planar and 3D BallBot models with an attitude reference trajectory that first accelerates and then stops the BallBot system

the main difference between the models. The body attitude for both 2D and 3D models are the same. The minor differences in the graph arise when the controller applies torque input and results in minor numerical differences between the models.

Chapter 4

CONTROLLERS AND SIMULATIONS

We have so far introduced a number of 3D BallBot models with different constraints and properties. In subsequent sections, controllers for these models will be presented, followed by associated simulations using both the 3D model with ball yaw constraint and 3D model with the IMD constraint to observe differences in performance and accuracy.

4.1 Control of BallBot Attitude

Regardless of its nonholonomic ball-ground contact constraints, BallBot is an underactuated system for its motion in $\mathcal{W}$. For this reason, the position and orientation of the ball cannot be directly controlled with the available control inputs. Any controller for the motion of the BallBot must control and use body states to manipulate ball dynamics towards the desired behavior. Section 4.2 elaborates further on this issue. Controllers for the new 3D models in the following sections are proposed under such considerations.

4.1.1 Pure PD Control

We begin our investigation of possible control strategies with a simple PD controller for BallBot body attitude angles that does not make explicit use of system dynamics. The structure of this controller is shown in Figure 4.1. Controller constants are tuned manually by considering convergence and stability properties of system trajectories.

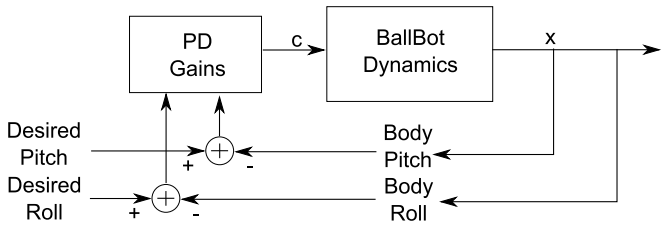

Figure 4.1: Detailed block diagram for pure PD control of the BallBot

Even though models can approximately follow desired trajectories with this controller to some extent, the torque requirements and resulting steady state errors are too high to be practically feasible. The resulting attitude tracking errors for the actual BallBot may not be suppressible with actuator limitations for fast maneuvers and highly dynamical motions. The performance of this controller will be studied in more detail in subsequent sections so we now proceed with the descriptions of model-based, more accurate and high performance controllers.

4.1.2 Inverse Dynamics Control Based on the 2.5D Model

As we mentioned in Section 4.1.1, more accurate controllers that are informed by the dynamics of the system are required to handle inertial effects that become significant at high speeds and dynamic maneuvers. Consequently, we now describe an inverse dynamic controller based on the 2.5D model to control the BallBot's body attitude within a simulated 3D model. We choose 2.5D model, since most of the current platforms uses 2.5D model for their controllers. This way, the efficiency of these existing control methods can be characterized.

In principle, the idea is to use the dynamics of the 2.5D model to compute a control torque vector that will stabilize a desired attitude angle trajectory. This controller effectively cancels gravitational and inertial effects on the body attitude, so that it provides accurate attitude control for the BallBot.

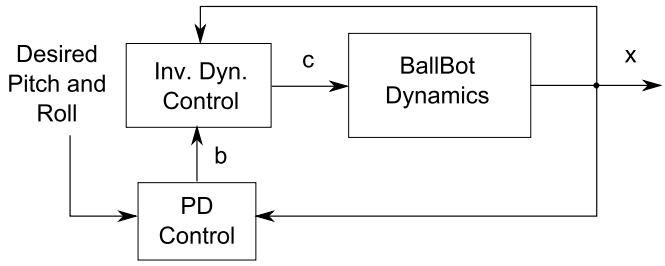

Figure 4.2: Block diagram for Inverse Dynamics Control of the BallBot attitude angles, supported by stabilizing PD feedback.

As illustrated by the structure of this control strategy shown in Figure 4.2, our main focus for the inverse dynamics controller is cancelling out accelerations on the body attitude degrees of freedom due to the BallBot dynamics, subsequently replacing them with Proportional Derivative (PD)feedback. Using this method, body attitude angles can be stabilized around desired trajectories, generated by a suitably chosen planner, such as the method described in [16].

We accomplish this cancellation by treating torque control inputs as unknowns and introducing additional constraints that force attitude dynamics to be reduced to simple, second order stable subsystems.We introduced these new constraints for the 2D planar model, using the desired angular acceleration of body attitude angle as the reference angular acceleration for both decoupled axes of 2.5D model. This idea is captured for both sagittal planes by the constraints

$$\ddot{\phi}_i = \ddot{\phi}_{desired} \ , \tag{4.1}$$

where i is either the pitch or roll degree of freedom. The solution of (2.21) with (4.1) now gives us the solution for τ that will accomplish the desired cancellation. Stable control is obtained by adding PD control onto this solution.

As a result of this inverse dynamics cancellation, the feedback PD constants can now be much smaller, decreasing actuator requirements and hence increasing practical feasibility of the controller. The resulting motion should also be smoother and more robust compared to the pure PD controller of the preceding section. Nevertheless, as our simulation results will show later, differences between the 2.5D and 3D models result in nonzero tracking errors, suggesting that more accurate control should be possible by using the 3D model itself as a basis for the controller.

4.1.3 Inverse Dynamics Control Based on the 3D Model

As mentioned in the previous section, using the 2.5D model as a basis for controlling the 3D model suffers from inaccuracies in tracking desired attitude angles. Consequently, we now consider using the 3D model directly for inverse dynamics control. This way, we can more accurately cancel out accelerations of the system for high speeds and dynamically challenging motions.

Similar to the 2.5D inverse dynamics controller, we begin by considering torque control inputs as unknowns for both $\mathbf{U}_r$ in (3.43) for the ball yaw constraint model, and $\mathbf{U}_{IMD}$ in (3.51) for the IMD constraint model as

$$\mathbf{U}'_r := \begin{bmatrix} \dot{\mathbf{P}}_\mathbf{b} & \dot{\mathbf{L}}_\mathbf{b} & \dot{\mathbf{w}}_\mathbf{b} & \dot{\mathbf{P}}_\mathbf{r} & \dot{\mathbf{L}}_\mathbf{r} & \dot{\mathbf{w}}_\mathbf{r} & \mathbf{F}_b & \mathbf{F}_r & \tau_{r,z} & \tau_b \end{bmatrix}^T , \tag{4.2}$$

$$\mathbf{U}'_{IMD} := \begin{bmatrix} \dot{\mathbf{P}}_\mathbf{b} & \dot{\mathbf{L}}_\mathbf{b} & \dot{\mathbf{w}}_\mathbf{b} & \dot{\mathbf{P}}_\mathbf{r} & \dot{\mathbf{L}}_\mathbf{r} & \dot{\mathbf{w}}_\mathbf{r} & \mathbf{F}_b & \mathbf{F}_r & \tau_{r,z} & \tau_{b,z} & \tau_{b,IMD} \end{bmatrix}^T . \tag{4.3}$$

Additional constraints necessary to ensure unique solutions for these unknowns for roll and pitch axes are given by

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \dot{\mathbf{L}}_\mathbf{b} = \begin{bmatrix} \tau_x^* \\ \tau_y^* \end{bmatrix} , \tag{4.4}$$

where τ_x^* and τ_y^* are the outputs of the PD gain control as described in Section 4.1.2.

For ball yaw constraint model, a third additional constraint is needed since τ_b has 3 inputs instead of 2 as in $\tau_{b,IMD}$. The attitude acceleration constraints for this model hence become

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\mathbf{L}}_\mathbf{b} = \begin{bmatrix} \tau_x^* \\ \tau_y^* \\ \tau_z^* \end{bmatrix} , \tag{4.5}$$

where τ_z^* is the yaw output of the PD gain.

Solutions for the controller can then be obtained by solving the augmented constraint equation $\mathbf{U}_r' = (\mathbf{M}_r')^{-1}\mathbf{N}_r'$ for the ball yaw constraint model and $\mathbf{U}_{IMD}' = (\mathbf{M}_{IMD}')^{-1}\mathbf{N}_{IMD}'$ for the IMD constraint model respectively. Those equations yield control torques that cancel out dynamics on the body attitude coordinates effectively, substituting PD feedback instead. This way, there is only decoupled stabilizing torques left in the equations. We use this controller in subsequent sections to illustrate various features of the 3D model with respect to its ability to capture interesting behaviors on positional variables.

4.2 Shape Variables vs. External Variables

Configuration variables of BallBot as an underactuated system can be divided into two groups. Configuration variables that appear in the inertia matrix are called shape variables, and remaining configuration variables are called external variables [31]. It is interesting to note that underactuated systems can be classified according to whether shape variables fully or partially actuated or unactuated. For details on the notion of shape and external variables, please refer to [32].

In this context, BallBot systems are considered to be shape-accelerated underactuated balancing system. For the 2D planar model, the body angle ϕ serves as the shape variable whereas the ball angle θ and the related ball positions

(x_r,y_r) are external variables [16]. Similar to the 2D planar model, the body attitude angles become shape variables of the system and cannot be directly controlled. In general, controllers for underactuated balancing systems generally aim to track desired trajectories for external variables while protecting balance. However, since shape-accelerated underactuated balancing systems have constraints on the accelerations of these external variables depending on the values of shape variables and their derivatives, using desired shape variables for tracking is preferable. For this reason, by controlling accelerations of external variables of BallBot with respect to desired shape variables, we control shape variables trajectories indirectly on our 3D model. For planning trajectories of external variables, we investigate the results of these shape variable trajectories in subsequent sections.

4.3 Tracking Circular Attitude Angle Profiles

Most of our simulations in this thesis focus on circular trajectories in body attitude coordinates (with period t_{cycle} and amplitude A_{max}). These circular trajectories lead BallBot to exhibit dynamically dexterous capabilities, which we cannot expect from linear paths. Such circular trajectories are important for locomotion of the BallBot as a base for circular planning, and it is helpful for excitation of yaw motion in 3D models, in particular IMD constraint model. Note that intuitively, we should expect such circular attitude trajectories to result in external variables following similarly circular paths for the ball COM as we mentioned in Section 4.2.

In order to prevent falling and obtain smooth transients, our simulations begin at $t = 0$ from an upright body posture with $\mathbf{q}_b = [1, 0, 0, 0]^T$. Subsequently, the body attitude is commanded to follow a pattern spiraling out for a duration of

t_{setup} to reach its periodic circular path until a final time, $t = t_{final}$, is reached, chosen to be large enough to ensure convergence to steady-state.

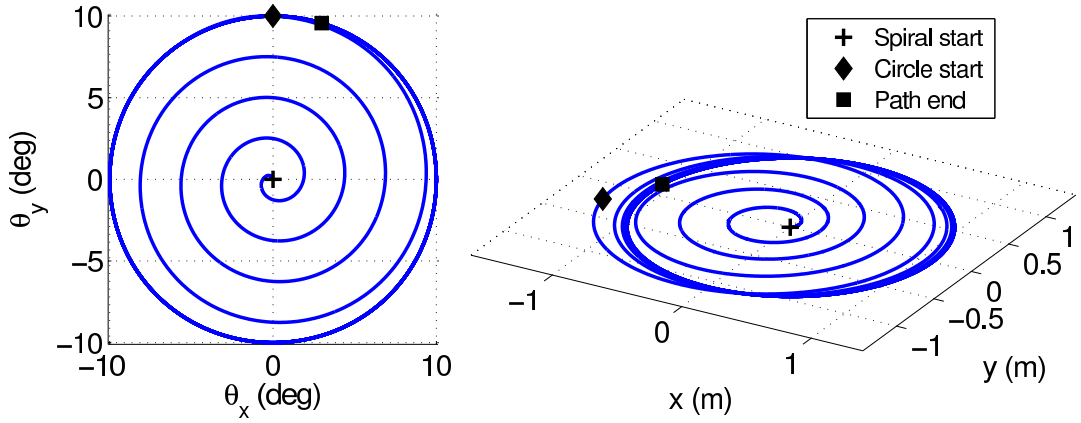

Figure 4.3: An example simulation with The 3D BallBot model, starting from an upright posture and spiraling out to a circular attitude trajectory. Left: Body attitude trajectory, Right: Ball trajectory in $\mathcal{W}$. This example has an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$.

Figure 4.3 illustrates an example for this attitude profile and the resulting robot motion in $\mathcal{W}$. As it can be seen, external variable trajectories converge to circular paths as expected. Even though the system is symmetric in median axes, the center of the circular path in $\mathcal{W}$ slightly drifts in the initial stage before converging to a single circle. This is a result of the initial nonzero velocity due to the spiraling out pattern and the viscous damping term included in (3.39). Viscous damping term eliminates the average translational velocity of the system when attitude trajectories are tracked with accuracy.

4.3.1 Performance Under Pure PD Control

In this section, we use pure PD control for tracking circular attitude angle trajectories and investigate its performance. In Figure 4.4, simulations with the ball yaw constraint 3D model are shown for an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$. We chose reasonable torque constants and obtained acceptable tracking performance. Trajectory tracking errors for

these simulations do not converge to zero, namely there is steady state error for pure PD controllers on the model.

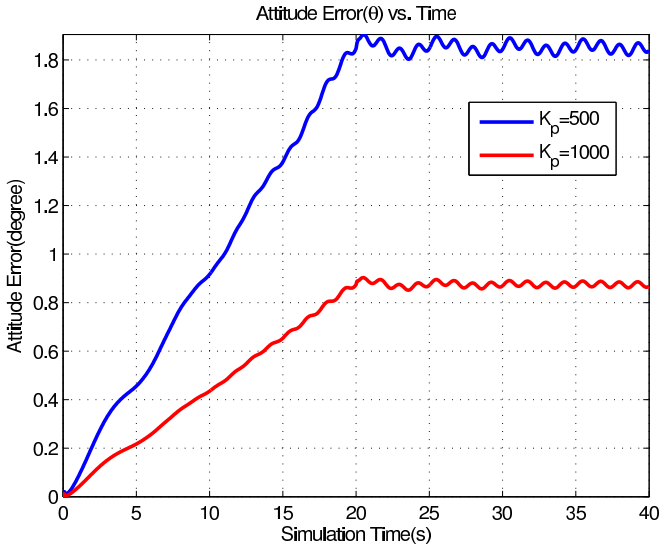

Figure 4.4: Attitude errors for an example simulation with The 3D ball yaw constraint BallBot model under pure PD control following a circular attitude angle trajectory. This example has an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$, with ($K_p = 500$, $K_d = 25$) and ($K_p = 1000$, $K_d = 50$).

In order to decrease the attitude error, we can always use controller gains. Corresponding simulation results for a typical run are shown in Figure 4.5. Even though the attitude error decreases with respect to the previous simulation, it still does not converge to zero even for higher, somewhat unrealistic PD constants for real systems with limited actuation capabilities. As a result, using behavioral components in the attitude controllers would be preferable for the ball yaw constraint 3D model in order to obtain higher performance controllers for dynamically challenging motions.

For the IMD constraint model, the PD controller shows a trajectory tracking performance similar to the performance of ball yaw constraint 3D model with smaller torque constants. We simulated the IMD constraint model with two different K_p values, for which simulation results are shown in Figure 4.6. Due to the IMD constraint on input torques, the ball yaw constraint model has slightly

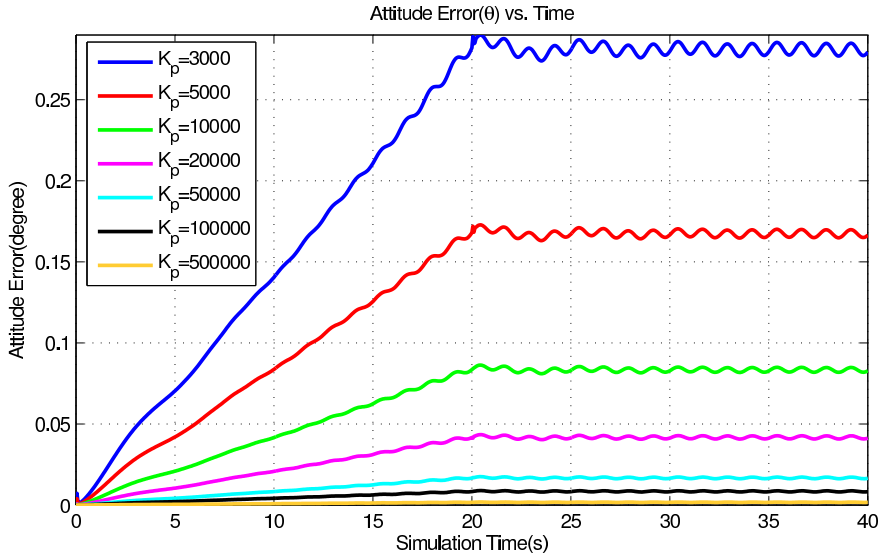

Figure 4.5: Attitude tracking errors for an example simulation with the 3D ball yaw constraint model under pure PD control on a circular trajectory. This example has an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$, with ($K_p = 3000$, $K_d = 60$), ($K_p = 5000$, $K_d = 70$), ($K_p = 10000$, $K_d = 100$), ($K_p = 20000$, $K_d = 150$), ($K_p = 50000$, $K_d = 200$), ($K_p = 100000$, $K_d = 300$), and ($K_p = 500000$, $K_d = 500$).

better tracking for the same values of the K_p constant. The yaw axis torque on the system cannot be actuated for the IMD constraint model.

Additionally, as we expected increasing the input torque constants improve the tracking performance of the IMD constrained 3D model to some degree as well. However, even after increasing input torque constants to unreasonably high values, like in Figure 4.7, attitude error never converges to zero.

For the aforementioned reasons, by adding some behavioral components to controller with our inverse dynamics controllers, we will investigate the tracking performance of both models.

4.3.2 Performance Under 2.5D Inverse Dynamics Control

In order to suppress inertial effects on high speeds and dynamic maneuvers, we introduced the inverse dynamic controller in Section 4.1.2. By using the 2.5D

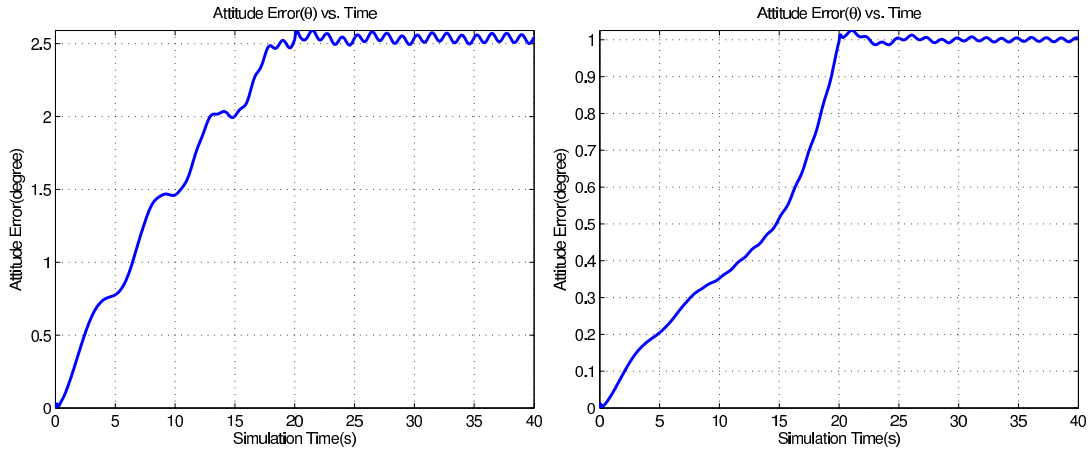

Figure 4.6: Attitude tracking errors for example simulations with the 3D IMD constraint model under pure PD control on a circular trajectory. This example has an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$, with ($K_p = 300$, $K_d = 15$) at left and ($K_p = 1000$, $K_d = 50$) at right.

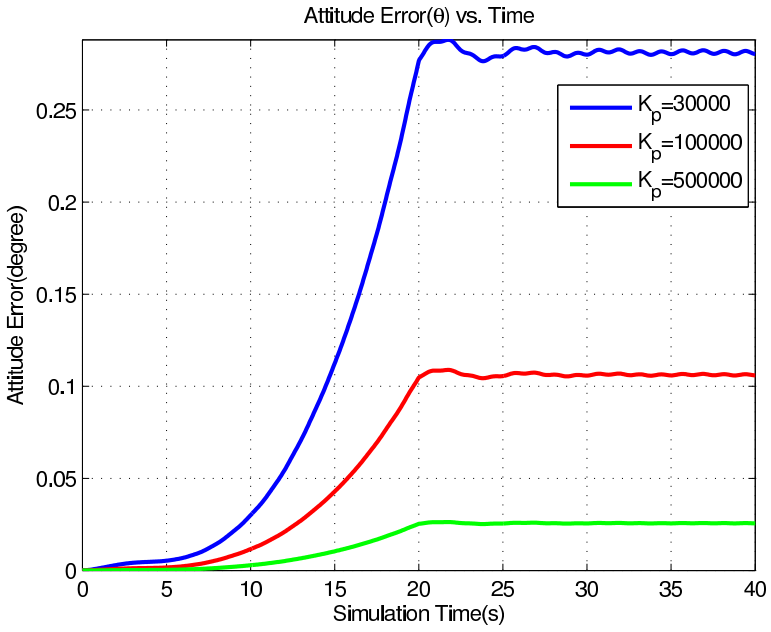

Figure 4.7: Attitude tracking errors for example simulations with the 3D IMD constraint model under pure PD control on a circular trajectory. This example has an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$, with ($K_p = 30000$, $K_d = 400$), ($K_p = 100000$, $K_d = 500$), and ($K_p = 500000$, $K_d = 600$) .

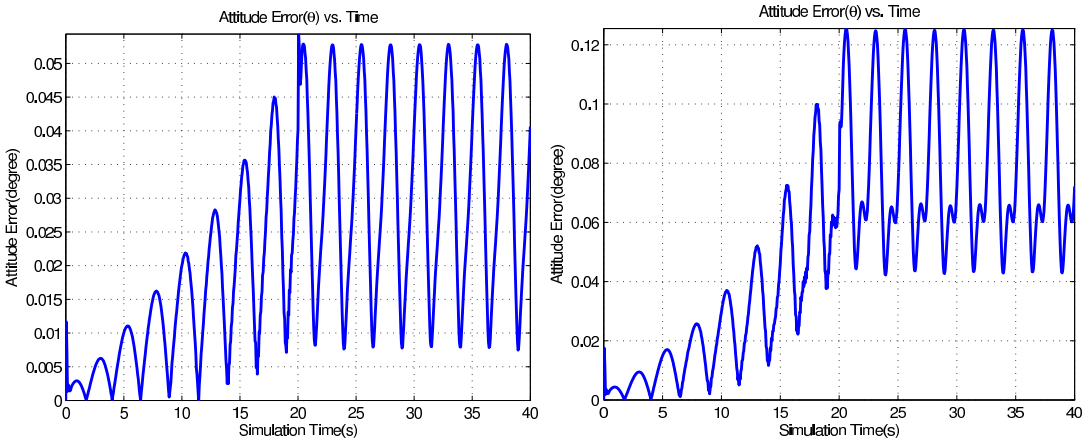

Figure 4.8: Attitude tracking errors for example simulations with the 3D ball yaw constrained model under 2.5D inverse dynamics control on a circular trajectory. This example has an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$ at left and $A_{max} = 15deg$ at right.

inverse dynamics controller, we achieve a much better and stable performance compared to pure PD controllers for the yaw constrained 3D BallBot model in highly dynamic cases. This can be observed from the simulation results in Figure 4.8. The tracking performance of the controller is naturally affected by the chosen trajectory, like its amplitude and period. Additionally, in all cases the attitude error, though bounded, saturates and does not converge to zero, since the 2.5D model does not exactly provide an inverse dynamics for the 3D ball yaw constrained model.

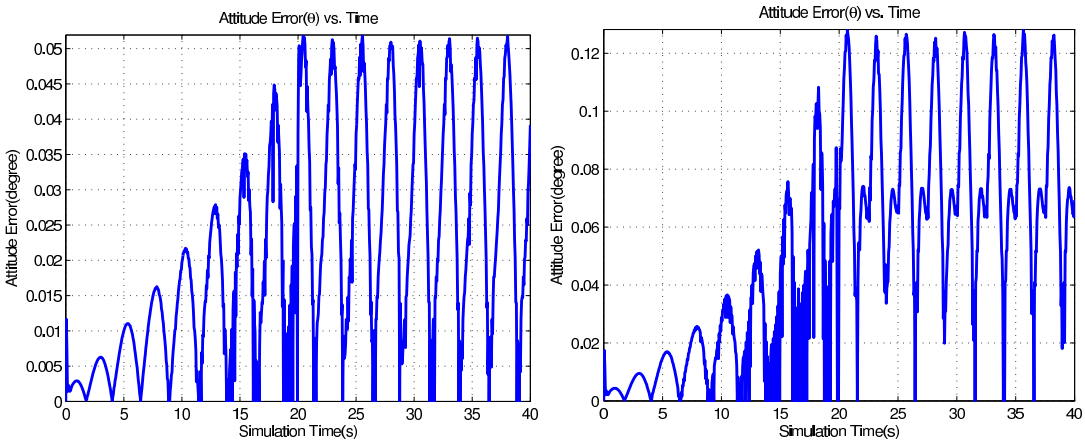

Figure 4.9: Attitude tracking errors for example simulations with the 3D IMD constraint model under 2.5D inverse dynamics control on a circular trajectory. This example has an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$ at left and $A_{max} = 15deg$ at right.

Similarly, the 3D IMD constrained BallBot model is also improved with the 2.5D attitude controller. Figure 4.9 shows attitude errors for the simulation of the model with 2.5D inverse dynamics controller for period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$ and $A_{max} = 15deg$. The trajectory parameters naturally effect the tracking performance, and this model also does not converge to zero for any circular trajectory with the 2.5D inverse dynamics controller since instead of the exact model we used 2.5D model for the desired attitude profile. Thus, we will use a better inverse dynamics controller for the system which can give the exact inverse dynamics of the system.

4.3.3 Performance Under 3D Inverse Dynamics Control

Since all our controllers are used on the 3D models, we expect that using 3D models in the inverse dynamics controller would improve the performance. Thus, for each 3D model, we use the inverse dynamics of the model in the controller to achieve better results.

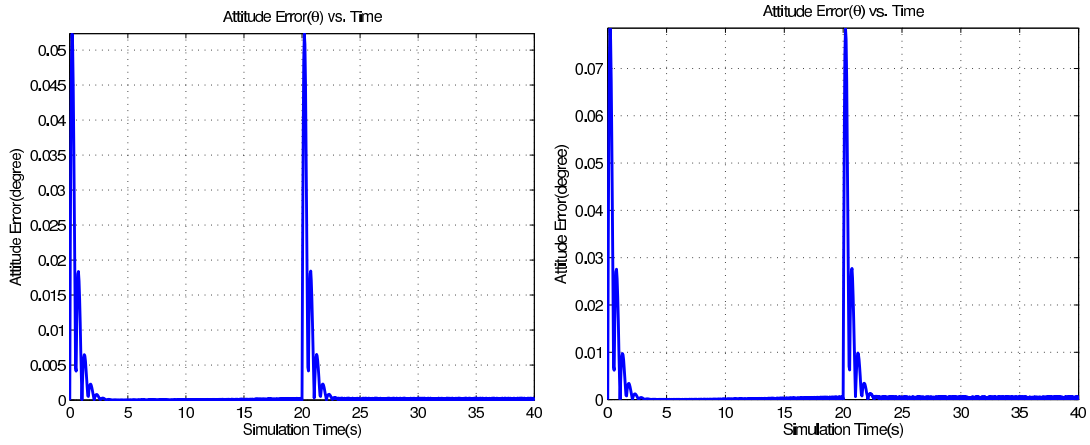

Figure 4.10: Attitude tracking errors for example simulations with the 3D ball yaw constrained model under 3D inverse dynamics control on a circular trajectory. This example has an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$ at left and $A_{max} = 15deg$ at right.

Figure 4.10 illustrates the simulation results for 3D ball yaw constrained BallBot model and 3D inverse dynamics controller, on a circular trajectory with

period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$ and $A_{max} = 15deg$. The attitude error of the system converges to zero at the steady state, and the tracking error at the higher speeds gives better results compared to 2.5D inverse dynamics controller for the 3D ball yaw constrained model.

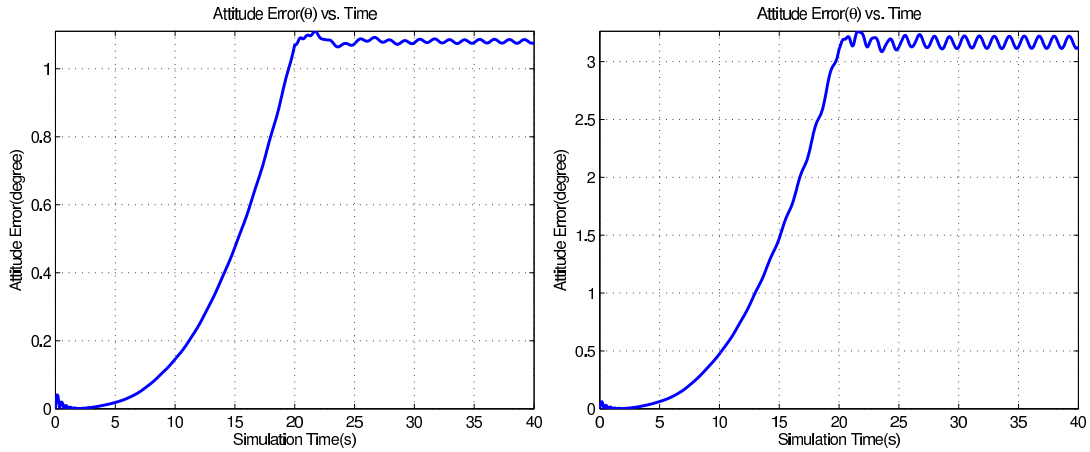

Figure 4.11: Attitude tracking errors for example simulations with the 3D IMD constrained model under 3D inverse dynamics control on a circular trajectory. This example has an attitude reference with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$ at left and $A_{max} = 15deg$ at right.

For the 3D IMD constrained BallBot model and 3D inverse dynamics controller, Figure 4.11 shows the results of the simulation on a circular trajectory with period $t_{cycle} = 5s$ and amplitude $A_{max} = 10deg$ and $A_{max} = 15deg$. The attitude error of the system is not improved as compared to 2.5D inverse dynamics controller for the 3D IMD constrained model. On the contrary, a serious performance loss in the trajectory tracking occurs. The reason is that the PD controller does not include a feedforward model of the reference trajectory, thus the poor performance of the controller dominates the steady-state tracking errors. Although including a feedforward term for the attitude reference trajectory for such a simple circular trajectory would be possible, in general when task-level trajectory planning on robot position is used to obtain the desired attitude angles, it may not be possible. Thus, we choose not to use a compensation term for the desired trajectory accelerations.

Note that, in our simulations, and the examples we used here have speeds up to $3.5m/s$ linear ball velocity, which is higher than what has been studied in the existing literature. Additionally, we expect to have feedback policies for the external variables, such as ball position, which will increase the robustness of the system, and eliminate the possible impact of the steady-state attitude tracking error on the behavior. Applying our model and inverse dynamics controllers to such high level trajectory planning projects is left as the future work.

4.3.4 Yaw Dynamics

As we mention before, our 3D models has the advantage of the ability to model any natural yaw dynamics. Similar to the yaw rotation observed when a conic object is rolling on the ground, the body is expected to rotate in yaw direction, when the body left the vertical axis, as a result of the nonholonomic rolling constraints and yaw constraints between body and the ball. However, for slow motions and linear motions, which are the main focus of the studies in the literature, such rotations are negligible.

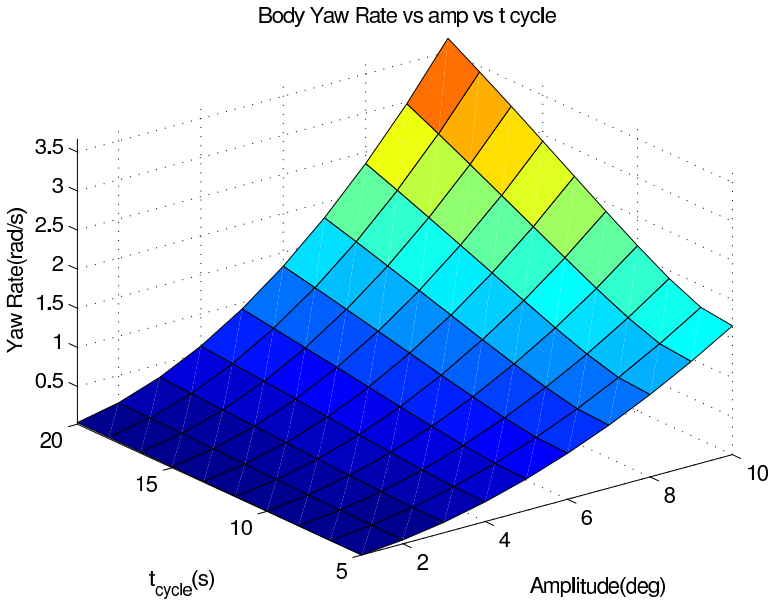

Figure 4.12: Dependence of the yaw rate to the period and amplitude of attitude angle reference trajectories.

Figure 4.12 illustrates the relation between BallBot's yaw angular velocity and the period and amplitude of the attitude reference trajectory. As it can be observed there is significant yaw rotation associated with circular motion of the platform. The yaw motion increases in magnitude as either the period or the amplitude of the attitude reference trajectory increase. Our 3D model is the first model capable of including this behavior into dynamics. Though, it can be measured with inertial sensing and can be compensated with independent feedback controllers, this inability to model such behaviors in the system dynamics would cause inaccuracies in the motion planning on the long run.

4.3.5 Characterizing External Variable Trajectories

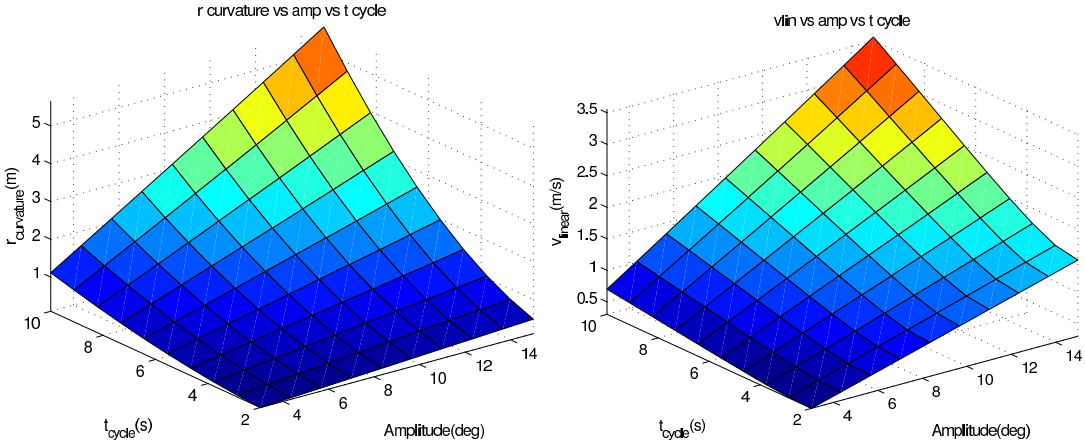

Figure 4.13: Dependence of the circular external variable trajectory parameters to the period and amplitude of the attitude reference trajectory. Left: radius of the circular path, Right: linear ball velocity along the circular path for 3D ball yaw constrained model.

Underactuated nature of the BallBot is one of the interesting feature of its morphology. Due to this fact, ball attitude, that is important for high level trajectory planning, can only be controlled indirectly through attitude angles. Our assumption, mentioned in Section 4.2, that circular trajectories in attitude angle space would lead to circular trajectories can be observed in our simulations. To further understand the relation between shape variables and external variables of the system, we ran simulations for a range of different attitude angle trajectories,

for the period t_{cycle} and the maximum attitude angle A_{max}, and found the corresponding radii of the circular reference trajectories. The reference trajectories for body attitude degrees of freedom hence become the following expression;

$$\theta_x = A_{max} \sin(2\pi t / t_{cycle}) \tag{4.6}$$

$$\theta_y = A_{max} \cos(2\pi t / t_{cycle}). \tag{4.7}$$

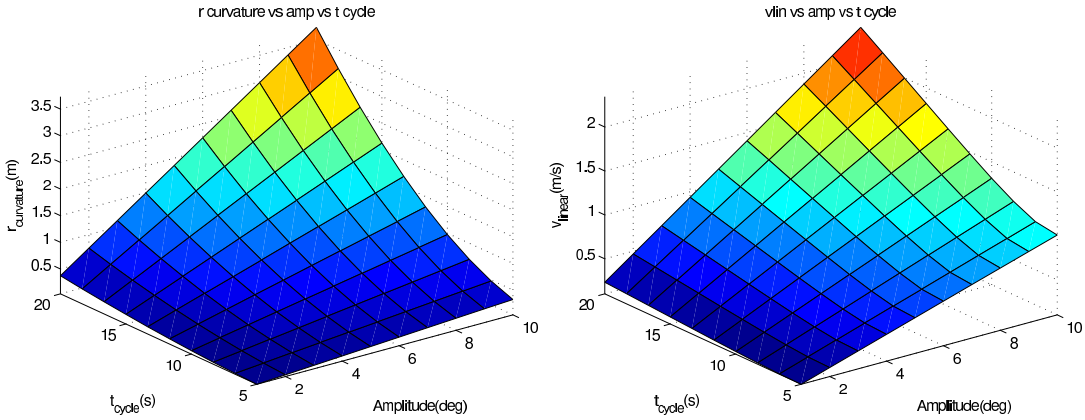

Figure 4.14: Dependence of the circular external variable trajectory parameters to the period and amplitude of the attitude reference trajectory. Left: radius of the circular path, Right: linear ball velocity along the circular path for 3D IMD constrained model.

Figure 4.13 and Figure 4.14 illustrates the results of the simulations with radius and linear velocity as the function of the reference trajectory period and amplitude, with the ranges $A_{max} \in [3, 15] deg$ and $t_{cycle} \in [2, 10] s$, and $A_{max} \in [1, 10] deg$ and $t_{cycle} \in [5, 20] s$, for 3D ball yaw constrained model and 3D IMD constrained model, respectively. Radii and linear velocities of the steady-state external variable trajectories are chosen as the parameterizations of the ball path in $\mathcal{W}$ [1]. Note that, radius and linear velocity associated with these external variables are independent of startup time and initial states.

Additionally, the radius of the steady state can be calculated by the basic assumption that the system is in the steady state. This way radius can become a linear function of velocity, with period as the proportion constant. Thus, both

[1]Radii are calculated by using averaged steady state radius of curvature, for further details of radius of curvature, see [33]

linear velocity of the ball and radius become nonlinear functions of period and amplitude. Using these functions for the radius of period $t_{cycle} = 5.56s$ and amplitude $A_{max} = 12.33deg$ point, the linear velocity is calculated as $1.838m/s$, and in the Figure 4.13, simulation gives $1.832m/s$, and the error in the calculation is the result of damping, which is neglected.

Chapter 5

CONCLUSIONS

In this thesis, we proposed an accurate, three dimensional model of the BallBot platform, which uses rolling on spherical wheels as the means of mobility. Our model use several constraint equations to capture interactions between the ground and the ball, and also between the ball and the robot body. Unlike the earlier modeling attempts that uses decoupled planar approximations as basis, important aspects of robot motion such as significant yaw rotations can be captured using our model.

Two different inverse-dynamics controllers are proposed, one based on planar approximations and 2.5D model, and another one based on our novel 3D controller. Their capability of sustaining dynamic behaviors such as circular trajectories in the workspace in a robust and stable fashion have been shown through simulation studies. These studies showed that tracking performance for shape variables of these controllers are on the acceptable levels. The relation between circular motions in shape variables and the characterized associated motions in external variables are investigated.

With these results at hand, as future work, we are ultimately aiming for dynamically dexterous behavioral controllers and motion planners for the BallBot

platform, using ideas as given in [34, 35]. In order to provide this, on the contrary to previous works on the subject, our goal is using dynamic properties of this system to its full possibilities. For this reason, through experiments, we first will validate this model. We also plan to work on some possible extensions of this model by considering more realistic friction models to increase its accuracy. In this context, we expect that the proposed 3D model and controllers based on this model will be important for the new possible accurate motion models for external variables of the system which are otherwise only indirectly controllable.

APPENDIX A

Quaternion Derivations

For a point 'a' at the top of the rigid rod as can be seen in Figure A.1, a_B is the position of the point in body frame coordinates and constant and a_W is the position of the point in world frame(inertial frame) coordinates, and rotation relation between a_W and a_B in quaternion coordinate frame is given as follows;

$$a_W = q \circ a_B \circ q^* \tag{A.1}$$

Quaternion multiplication is $q_3 = q_1 \circ q_2 = f(q_1, q_2)$. Then, derivative of q_3 is;

$$\frac{dq_3}{dt} = \frac{\partial f}{\partial q_1} \circ \dot{q}_1 + \frac{\partial f}{\partial q_2} \circ \dot{q}_2 \tag{A.2}$$

Thus for $a_W = q \circ a_B \circ q^*$, the following expression holds;

$$\frac{d}{dt}(q \circ (a_B \circ (q^*))) = \frac{\partial f}{\partial q} \circ \dot{q} + \frac{\partial f}{\partial q_2}\bigg|_{a \circ q^*} \circ (\frac{\partial f}{\partial q} \circ \cancelto{0}{\dot{a}} + \frac{\partial f}{\partial q_2}\bigg|_{q} \circ \dot{q}^*) \quad [1] \tag{A.3}$$

$$= \frac{\partial f}{\partial q} \circ \dot{q} + \frac{\partial f}{\partial q_2}\bigg|_{a \circ q^*} \circ \frac{\partial f}{\partial q_2}\bigg|_{q} \circ (\dot{q})^* \tag{A.4}$$

Fact 1: $(\dot{q^*})$ or $(\dot{q}^*)$ and $(\dot{q})^*$ are equal.

[1] Since a is a constant point on the rigid body.

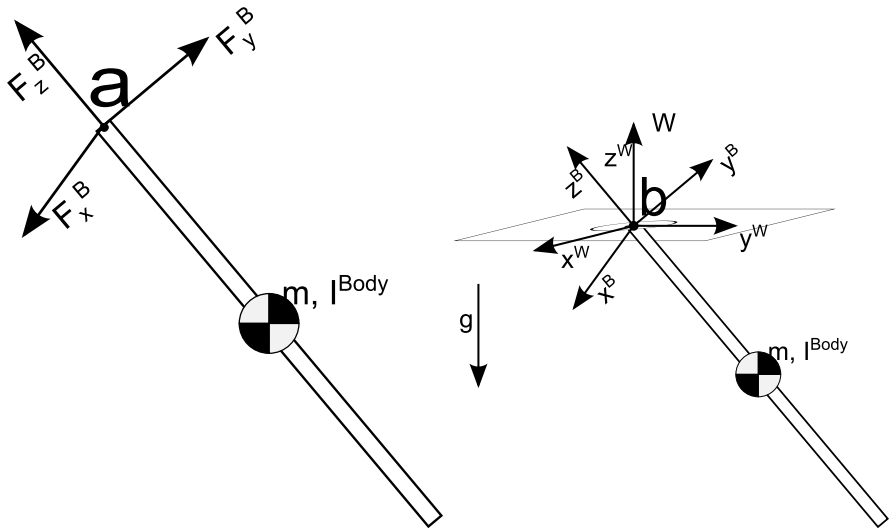

Figure A.1: Spherical Pendulum Model

Proof: Since $q^* = Cq$ [2], Then;

$$\dot{q} = \lim_{\Delta x \to 0} \frac{q(t+\Delta x) - q(t)}{\Delta x} \qquad and, \tag{A.5}$$

$$(\dot{q^*}) = \lim_{\Delta x \to 0} \frac{q^*(t+\Delta x) - q^*(t)}{\Delta x} \tag{A.6}$$

$$= \lim_{\Delta x \to 0} \frac{Cq(t+\Delta x) - Cq(t)}{\Delta x} \tag{A.7}$$

$$= C \lim_{\Delta x \to 0} \frac{q(t+\Delta x) - q(t)}{\Delta x} \tag{A.8}$$

$$= C\dot{q} = (\dot{q})^* \tag{A.9}$$

■

Fact 2: Quaternion conjugation over quaternion multiplication and addition operations are given below for $a = b \circ c$,$d = b + c$;

$$a^* = c^* \circ b^* \tag{A.10}$$

$$d^* = b^* + c^* \tag{A.11}$$

[2] as C is a diagonal matrix with elements $[1 - 1 - 1 - 1]$ and the conjugation matrix for quaternions

Proof: Transpose of the conventional matrix representation gives us the conjugate quaternion's conventional representation. Namely;

$$P_p^T = p^* \tag{A.12}$$

Thus we can show that, for $a = b \circ c$ then

$$a^* = (b \circ c)^* = (P_b P_c)^T = P_c^T P_b^T = c^* \circ b^* \tag{A.13}$$

and for $d = b + c$ then

$$d^* = (b + c)^* = (P_b + P_c)^T = P_b^T + P_c^T = b^* + c^* \tag{A.14}$$

■

Fact 3: Derivative of multiplication rule is valid for quaternion multiplication.

Proof: For $M(p, q) = p \circ q = Pq = Qp$

$$\frac{d}{dt}(p \circ q) = \frac{\partial M}{\partial p} \circ \dot{p} + \frac{\partial M}{\partial q} \circ \dot{q} \tag{A.15}$$

Since there is no dependent term in M for both representation $\frac{\partial Qp}{\partial p} = Q$(derivative of matrix multiplication);

$$\frac{d}{dt}(p \circ q) = Q\dot{p} + P\dot{q} \tag{A.16}$$

$$= \dot{p} \circ q + p \circ \dot{q} \tag{A.17}$$

Also, since quaternion multiplication has derivative, an alternative method for proof would be;

$$\dot{q} = \lim_{\Delta x \to 0} \frac{q(t + \Delta x) - q(t)}{\Delta x} \qquad and, \tag{A.18}$$

$$p \dot{\circ} q = \lim_{\Delta x \to 0} \frac{p(t + \Delta x) \circ q(t + \Delta x) - p(t) \circ q(t)}{\Delta x} \tag{A.19}$$

$$= \lim_{\Delta x \to 0} \frac{p(t + \Delta x) \circ q(t + \Delta x) - p(t) \circ q(t)}{\Delta x} + \lim_{\Delta x \to 0} \frac{(p(t + \Delta x) \circ q(t) - p(t + \Delta x) \circ q(t))}{\Delta x} \tag{A.20}$$

$$= \lim_{\Delta x \to 0} \frac{p(t + \Delta x) \circ q(t + \Delta x) - p(t + \Delta x) \circ q(t)}{\Delta x} + \lim_{\Delta x \to 0} \frac{-p(t) \circ q(t) + p(t + \Delta x) \circ q(t)}{\Delta x} \tag{A.21}$$

and, since quaternion multiplication is distributive over addition, Then;

$$
\begin{aligned}
p \dot{\circ} q &= \lim_{\Delta x \to 0} \frac{p(t+\Delta x) \circ (q(t+\Delta x) - q(t))}{\Delta x} \\
&+ \lim_{\Delta x \to 0} \frac{(p(t+\Delta x) - p(t)) \circ q(t)}{\Delta x} && \text{(A.22)} \\
&= \lim_{\Delta x \to 0} \frac{p(t+\Delta x) \circ (q(t+\Delta x) - q(t))}{\Delta x} \\
&+ \lim_{\Delta x \to 0} \frac{(p(t+\Delta x) - p(t)) \circ q(t)}{\Delta x} && \text{(A.23)} \\
&= p(t) \circ \left(\lim_{\Delta x \to 0} \frac{(q(t+\Delta x) - q(t))}{\Delta x} \right) \\
&+ \left(\lim_{\Delta x \to 0} \frac{(p(t+\Delta x) - p(t))}{\Delta x} \right) \circ q(t) && \text{(A.24)} \\
&= \dot{p} \circ q + p \circ \dot{q} && \text{(A.25)}
\end{aligned}
$$

■

We may as well write the following expression for derivative of a_W, assuming a will be constant w.r.t. body frame,[3]

$$
\begin{aligned}
\frac{da_W}{dt} &= \frac{d}{dt}(q \circ a_B \circ q^*) && \text{(A.26)} \\
&= \dot{q} \circ (a \circ q^*) + q \circ (\overset{0}{\cancel{\dot{a}}} \circ q^* + a \circ \dot{q}^*) && \text{(A.27)} \\
&= \dot{q} \circ a \circ q^* + q \circ a \circ \dot{q}^* && \text{(A.28)} \\
&= \dot{q} \circ q^* \circ a + a \circ q \circ \dot{q}^* && \text{(A.29)}
\end{aligned}
$$

Then derivative of $\dot{a}_W$ is given as below;

$$
\begin{aligned}
\frac{d}{dt}(\frac{da_W}{dt}) &= \frac{d}{dt}(\dot{q} \circ a \circ q^* + q \circ a \circ \dot{q}^*) && \text{(A.30)} \\
&= \ddot{q} \circ a \circ q^* + \dot{q} \circ a \circ \dot{q}^* + \dot{q} \circ a \circ \dot{q}^* + q \circ a \circ \ddot{q}^* && \text{(A.31)} \\
&= \underbrace{\ddot{q} \circ a \circ q^*}_{Q_{a \circ q^*} \ddot{q}} + \underbrace{\dot{q} \circ a \circ \dot{q}^* + \dot{q} \circ a \circ \dot{q}^*}_{2\dot{q} \circ a \circ \dot{q}^*} + \underbrace{q \circ a \circ \ddot{q}^*}_{P_{q \circ a} C \ddot{q}} && \text{(A.32)}
\end{aligned}
$$

Thus, $\ddot{a}_W$ equation becomes;

$$
(P_{q \circ a} C + Q_{a \circ q^*}) \ddot{q} = \ddot{a}_W - 2\dot{q} \circ a \circ \dot{q}^* \qquad \text{(A.33)}
$$

[3]Since,the proof of $\dot{q} = \frac{1}{2} w \circ q$ is done only for constant a case, dynamic case should be calculated for a dynamic COM condition.

To calculate the (A.33)i, $\ddot{q}$ is given as follows;

$$\dot{q} = \frac{1}{2} w_w \circ q \quad \text{[4]} \quad \text{and,} \tag{A.34}$$

$$\ddot{q} = \frac{1}{2}\frac{d}{dt}(w_w \circ q) \tag{A.35}$$

$$= \frac{1}{2}\dot{w}_w \circ q + \frac{1}{2} w_w \circ \dot{q} \tag{A.36}$$

Then $\ddot{a}_W$ can be computed as the following expression;

$$\ddot{a}_W = (\frac{1}{2}\dot{w}_w \circ q + \frac{1}{2} w_w \circ \dot{q}) \circ a \circ q^* + 2\dot{q} \circ a \circ \dot{q}^* + q \circ a \circ (\frac{1}{2}\dot{w}_w \circ q + \frac{1}{2} w_w \circ \dot{q})^* \tag{A.37}$$

Using the properties mentioned at equations (A.11) and (A.10), the following holds;

$$\begin{aligned} \ddot{a}_W &= (\frac{1}{2}\dot{w}_w \circ q + \frac{1}{2} w_w \circ \dot{q}) \circ a \circ q^* + 2\dot{q} \circ a \circ \dot{q}^* \\ &\quad + q \circ a \circ (\frac{1}{2} q^* \circ \dot{w}_w^* + \frac{1}{2}\dot{q}^* \circ w_w^*) && \text{(A.38)} \\ &= \frac{1}{2}\dot{w}_w \circ q \circ a \circ q^* + \frac{1}{4} w_w \circ w_w \circ q \circ a \circ q^* \\ &\quad + \frac{1}{2} w_w \circ q \circ a \circ q^* \circ w_w^* && \text{(A.39)} \\ &\quad + \frac{1}{2} q \circ a \circ q^* \circ \dot{w}_w^* + \frac{1}{4} q \circ a \circ q^* \circ w_w^* \circ w_w^* && \text{(A.40)} \end{aligned}$$

Since $a_W = q \circ a_B \circ q^*$, then the final expression for the second derivative of the expression becomes as given in below,

$$\begin{aligned} \ddot{a}_W &= \frac{1}{2}\dot{w}_w \circ a_W + \frac{1}{2} a_W \circ \dot{w}_w^* \\ &\quad + \frac{1}{4} w_w \circ w_w \circ a_W + \frac{1}{2} w_w \circ a_W \circ w_w^* + \frac{1}{4} a_W \circ w_w^* \circ w_w^* \end{aligned} \tag{A.41}$$

[4] Also in body coordinates this becomes $\dot{q} = \frac{1}{2} q \circ w_B$

Bibliography

[1] U. Nagarajan, A. Mampetta, G. A. Kantor, and R. L. Hollis, "State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 998 –1003, may 2009.

[2] P. S. Schenker, T. L. Huntsberger, P. Pirjanian, E. T. Baumgartner, and E. Tunstel, "Planetary rover developments supporting mars exploration, sample return and future human-robotic colonization," *Autonomous Robots*, vol. 14, pp. 103–126, Mar. 2003.

[3] B. M. Yamauchi, "Packbot: a versatile platform for military robotics," in *Proceedings of SPIE: Unmanned Ground Vehicle Technology VI*, vol. 5422, pp. 228–237, September 2004.

[4] U. Saranli, M. Buehler, and D. E. Koditschek, "RHex: A simple and highly mobile robot," *International Journal of Robotics Research*, vol. 20, pp. 616–631, July 2001.

[5] R. Playter, M. Buehler, and M. Raibert, "BigDog," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6230 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, June 2006.

[6] E. Hale, N. Schara, J. Burdick, and P. Fiorini, "A minimally actuated hopping rover for exploration of celestialbodies," in *Robotics and Automation,*

2000. Proceedings. ICRA '00. IEEE International Conference on, vol. 1, (San Francisco, CA), pp. 420–427, April 2000.

[7] D. Liu, H. Sun, Q. Jia, and L. Wang, "Motion control of a spherical mobile robot by feedback linearization," in *Proc. of the World Congress on Intelligent Control and Automation*, pp. 965 –970, june 2008.

[8] T. Lauwers, G. Kantor, and R. Hollis, "One is enough!," in *Robotics Research* (S. Thrun, R. Brooks, and H. Durrant-Whyte, eds.), vol. 28 of *Springer Tracts in Advanced Robotics*, pp. 327–336, Springer Berlin / Heidelberg, 2007.

[9] M. Kumagai and T. Ochiai, "Development of a robot balanced on a ball - first report, implementation of the robot and basic control," *Journal of Robotics and Mechatronics*, vol. 22, no. 3, pp. 348–355, 2010.

[10] U. Nagarajan, G. Kantor, and R. L. Hollis, "Human-robot physical interaction with dynamically stable mobile robots," in *Proc. of the ACM/IEEE int. Conf. on Human robot interaction*, HRI '09, (New York, NY, USA), pp. 281–282, ACM, 2009.

[11] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, (Orlando, FL.), pp. 2884–2889, May 2006.

[12] M. Kumagai and T. Ochiai, "Development of a robot balancing on a ball," in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pp. 433 –438, oct. 2008.

[13] R. Murray, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton: CRC Press, 1994.

[14] U. Nagarajan, G. Kantor, and R. Hollis, "Trajectory planning and control of an underactuated dynamically stable single spherical wheeled mobile robot,"

in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 3743 –3748, May 2009.

[15] E. M. Schearer, "Modeling dynamics and exploring control of a single-wheeled dynamically stable mobile robot with arms," Tech. Rep. CMU-RI-TR-06-37, Carnegie Mellon University, August 2006.

[16] U. Nagarajan, G. Kantor, and R. Hollis, "Hybrid control for navigation of shape-accelerated underactuated balancing systems," in *Proc. of the IEEE Conf. on Decision and Control*, pp. 3566 –3571, dec. 2010.

[17] V. A. Joshi, R. N. Banavar, and R. Hippalgaonkar, "Design and analysis of a spherical mobile robot," *Mechanism and Machine Theory*, vol. 45, no. 2, pp. 130 – 136, 2010.

[18] J. Ginsberg, *Engineering Dynamics*. Cambridge: Cambridge University Press, 2008.

[19] P. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*. Englewood Cliffs: Prentice-Hall, 1988.

[20] M. Spong, *Robot Modeling and Control*. Chichester: John Wiley & Sons, 2006.

[21] K. S. Fu, R. Gonzalez, and C. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. Mcgraw-Hill Book Company, 1987.

[22] R. Jazar, *Theory of Applied Robotics: Kinematics, Dynamics, and Control*. Berlin: Springer, 2010.

[23] A. Hanson, *Visualizing Quaternions*. San Diego: Morgan Kaufmann, 2006.

[24] J. Vince, *Geometric algebra for computer graphics*. London: Springer, 2010.

[25] C. Crane, *Kinematic analysis of robot manipulators*. Cambridge, U.K. New York, NY, USA: Cambridge University Press, 2008.

[26] R. Goldman, *Rethinking quaternions theory and computation.* San Rafael, Calif: Morgan & Claypool, 2010.

[27] K. Shoemake, "Animating rotation with quaternion curves," *SIGGRAPH Comput. Graph.*, vol. 19, pp. 245–254, July 1985.

[28] D. H. Eberly and K. Shoemake, "Chapter 10 - quaternions," in *Game Physics*, pp. 507 – 544, San Francisco: Morgan Kaufmann, 2004.

[29] D. Baraff, "An introduction to physically based modeling: Rigid body simulation i - unconstrained rigid body dynamics," in *In An Introduction to Physically Based Modelling, SIGGRAPH '97 Course Notes*, p. 97, 1997.

[30] J. T. Wen, "Control of nonholonomic systems," in *The Control Handbook* (W. Levine, ed.), pp. 1359–1368, Boca Raton: CRC Press, 1996.

[31] U. Nagarajan, "Dynamic constraint-based optimal shape trajectory planner for shape-accelerated underactuated balancing systems," in *Proc. of the Robotics, Science and Systems Conference*, (Zaragoza, Spain), June 2010.

[32] R. Olfati-Saber, *Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2001.

[33] M. Weir, *Thomas' calculus.* Boston: Pearson Addison Wesley, 2005.

[34] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *TheInternational Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[35] P. Reist and R. Tedrake, "Simulation-based lqr-trees with input and state constraints," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 5504–5510, may 2010.