BODY ATTITUDE CONTROL OF A PLANAR ONE-LEGGED HOPPING ROBOT USING A NOVEL AIR DRAG ASSISTED REACTION WHEEL

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

BY

NEŞET ÜNVER AKMANDOR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2016

Approval of the thesis:

BODY ATTITUDE CONTROL OF A PLANAR ONE-LEGGED HOPPING ROBOT USING A NOVEL AIR DRAG ASSISTED REACTION WHEEL

submitted by **NEŞET ÜNVER AKMANDOR** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** ______________

Assoc. Prof. Dr. Afşar Saranlı
Supervisor, **Electrical and Electronics Eng. Dept., METU** ______________

Assoc. Prof. Dr. Yiğit Yazıcıoğlu
Co-supervisor, **Mechanical Engineering Dept., METU** ______________

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Department, METU ______________

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering Department, METU ______________

Assoc. Prof. Dr. Emre Tuna
Electrical and Electronics Engineering Department, METU ______________

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Department, METU ______________

Assist. Prof. Dr. Melih Çakmakçı
Mechanical Engineering Department, Bilkent University ______________

Date: ______________

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

Name, Last Name: NEŞET ÜNVER AKMANDOR

Signature :

ABSTRACT

BODY ATTITUDE CONTROL OF A PLANAR ONE-LEGGED HOPPING ROBOT USING A NOVEL AIR DRAG ASSISTED REACTION WHEEL

Akmandor, Neşet Ünver
M.S., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Afşar Saranlı
Co-Supervisor : Assoc. Prof. Dr. Yiğit Yazıcıoğlu

January 2016, 80 pages

In the literature, spring-loaded inverted pendulum (SLIP) model with damping has been used to represent the dynamics of legged locomotion. Based on a planar version of the model, a group of existing work focus on controlling the hip torque (between body and leg) in stance and in flight phases to generate stable planar locomotion (the SLIP-T model). Most of these studies assume an infinite body inertia such that the applied hip torque does not affect the attitude of the robot body. In practice, for any finite robot body inertia, applying time varying hip torque profiles will result in a dynamic change in the body attitude. To cancel this attitude disturbance, a compensating torque is required to be applied directly to the robot body. It is possible to use a reaction wheel to generate this torque. However, if the required torque is biased with a positive or negative direction over each stride (which is the case for hopping locomotion) the resulting wheel velocity becomes unbounded and unrealizable in practice. To solve this problem in the scope of the thesis, we propose a novel air drag assisted reaction wheel that generates a torque proportional to both wheel speed and wheel acceleration to achieve sustainable stabilization of the body attitude. We derive the dynamic model of both (regular and drag based) systems and present them in detail. Using these dynamics, we perform hybrid system simulation (with ground contact) of the planar robot system during locomotion. Under two different locomotion controllers from the literature, we demonstrate the disturbances on the body attitude and propose PD and PID based control of the regular and drag based reaction wheels to stabilize the platform. Having successful results from our simulation experiments, we also test the feasibility of the approach by conducting physical experiments to determine the required and obtainable drag torque.

Keywords: reaction wheel, spring loaded inverted pendulum (SLIP), legged locomotion, PID control, biologically inspired robotics

ÖZ

HAVA SÜRTÜNMESİ DESTEKLİ ÖZGÜN BİR TEPKİ TEKERLEĞİ İLE TEK BACAKLI DÜZLEMSEL BİR ROBOTUN GÖVDE DENGESİNİN SAĞLANMASI

Akmandor, Neşet Ünver

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Doç. Dr. Afşar Saranlı

Ortak Tez Yöneticisi : Doç. Dr. Yiğit Yazıcıoğlu

Ocak 2016 , 80 sayfa

Literatürde yer alan pek çok çalışmada sönümlemeli yay beslemeli ters sarkaç (YTS), bacaklı hareket dinamiğini modellemek için kullanılmaktadır. Bu modeli baz alan bazı çalışmalarda, robotun bacağına uygulanan torkun kontrolü ile robot kararlı bir şekilde hareket ettirilmektedir. Ancak bu çalışmalarda, gövdenin yere göre açısal serbestlik derecesinin bulunmadığı varsayılarak, gerçekte bacağa uygulanan torkun, gövdeye olan eşit şiddette ama ters yöndeki etkisi, sistemin dinamiğine katılmamaktadır. Robotun bacak hareketi sırasında gövdenin açısında oluşan bozulmayı düzeltmek için gövdeye dengeleyici bir tork verilmesi gerekmektedir. Gereken bu tork, bir tepki tekerleği ile sisteme verilebilir. Ancak her adımda, eğer sisteme sürekli aynı yönde (pozitif veya negatif) bir tork verilmesi gerekiyorsa, tepki tekerleğinin hızı belli bir yönde artmakta, bu da sistemin gerçekte uygulanabilirliğini engellemektedir. Bu sorunu çözmek adına tez kapsamında, hava sürtünmesi destekli özgün bir tepki tekerleği tasarımı ile, tekerleğin hızıyla ve ivmesiyle orantılı bir tork elde ederek, robotun gövdesi için sürdürülebilir bir denge sağladık. Her iki sistem modelinin (klasik ve hava sürtünmeli yaklaşım) içinde bulunduğu fazlara (uçuş ve yer) göre dinamik denklemleri elde ettik ve bunları tez içinde ayrıntılı olarak sunduk. Bu dinamik denklemleri kullanarak ve yer ile yaptığı etki-tepki kuvvetlerini de hesaba katarak, melez sistemin düzlemsel hareket sırasındaki fiziksel benzetimini gerçekleştirdik. Bu benzetimlerde, literatürdeki bacaklı robotların hareketini sağlayan iki farklı kontrolcü altında robotun bozulan gövde açısını düzeltmek için, hem klasik hem de hava sürtünmeli tepki tekerleği yaklaşımlarını PD ve PID metodlarını kullanarak kontrol ettik. Benzetim çalışmaları ile fikrimizin uygunluğunu başarılı bir şekilde gösterirken, fiziksel deneylerle de gereken sürtünme torklarına ulaşılabileceğini belirledik.

Anahtar Kelimeler: tepki tekerleği, yaylı ters sarkaç (YTS), bacaklı hareketlilik, PID kontrol, biyolojiden esinlenmeli robotik

To whom seeks stable balance in life...

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisors Afşar Saranlı and Yiğit Yazıcıoğlu for their endless guidance and support throughout my thesis. They always create some time for me when I needed. The conversations that we have made not only contributed to improve my academic skills but also they were truly inspiring. Actually, the main contribution of my thesis, "the drag wheel" idea, emerged during one of such brainstorming meetings. Hence, I want to thank once more Yiğit Yazıcıoğlu who is the originator of our novel concept. Besides, I am very grateful to Afşar Saranlı for his positive attitude which boosts my morale when I was low.

This is also a great opportunity to thank Uluç Saranlı who inspired and motivated me to work in legged robotics. While his research leaded my way, personally he always listened me and gave useful advices when I need a help. Moreover, I will always take his scientific enthusiasm and diligence as a model in my engineering career.

During my research, ATLAS (Interdisciplinary Robotics Research Lab) and ROLAB (Laboratory of Robotics and Autonomous Systems) provided not only good working environments but also gave me a chance to work with amazing colleagues such as Orkun Öğücü, İsmail Uyanık, Ali Nail İnal, Görkem Seçer, Sıtar Kortik, Sinan Şahin Candan, Ayşe Deniz Duyul, Hasan Eftun Orhon and Sait Sovukluk. Especially, I would like to thank Mehmet Mutlu who helped me a lot during my adaptation period and my brothers/sister in arms Gökhan Koray Gültekin, Çağlar Seylan and Merve Özen for their invaluable technical supports and friendship.

I would also like to thank TÜBITAK-BIDEB (The Scientific and Technical Research Council of Turkey) for supporting my M.Sc. by 2210-C National Graduate Scholarship Programme.

I have a close-knit family. Through all my life, I have always felt their ultimate support and love. Even the ones who I have a limited relation or who has passed away, they should know that each one is very unique for me and I sincerely thank them all. Most particularly, I would like to thank to my grandmothers Ayten Akmandor and Sabiha Parlak, my grandfathers Mehmet Neşet Akmandor and Cemal Parlak, my lovely mother Serap Akmandor, my dear father İbrahim Sinan Akmandor, my beloved sister Özge Akmandor and my mellow cat Siyah for their endless love and all opportunuties that they have provided to our family.

"Let us be grateful to people who make us happy, they are the charming gardeners who make our souls blossom" said Marcel Proust. Luckily, I have such gardeners in my life and I am absolutely sure that I could not make it without them. Specifically, I thank Erhan Gül, Can Bayık and Burcu Balçık for our funny and beery friendship. I also thank Ali Peker and Semiray Aydoğan for their beautiful smiling faces that always lift my spirit. Most especially, I would like to thank my sweetheart Burcu Güneysu for her priceless support, friendship, love and every single moment that we have shared together.

Finally, I have a very special thanks to my green trousers. When no one left to support my belly, it was right there. Hence, it will always have a place in my wardrobe...

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

CoM	Center-of-mass
fbd	Free-body diagram
ODE	Ordinary Differential Equations
PD	Proportional-Derivative
PID	Proportional-Integral-Derivative
SLIP	Spring Loaded Inverted Pendulum
SLIP-T	Torque Actuated Spring Mass Hopper

CHAPTER 1

INTRODUCTION

1.1 Motivation and Existing Work

For all living creatures, we observe that mobility is essential. Taking into account the random distribution of resources on our planet, we either need to be luckily positioned or need to move so as to be closer to those resources. Thus, evolution generated a large variety of different locomotion mechanisms and techniques in biological creatures. These have evolved conforming with the environmental requirements and the most common locomotion mechanisms emerged: fins for swimming, wings for flying and legs for walking and running. When combined with biological sensors and the brain, these mechanisms are able to perform complex behaviors. For example in humans, this ranges from simple walking to artistic gymnastic movements that require high level skills and coordination of muscles leading to dynamic balance.

Some robotics research topics [8, 25, 35, 44] are inspired from nature (biomimetry or biomimicry) and they proceed to analyze the system dynamics behind naturally occurring locomotion behaviors and mechanisms. A good example is the leg mechanism of the cheetah robot [2] which is inspired by musculoskeletal structure of the fastest running animal in the world. The methodology in that work is to obtain a high fidelity mathematical model that is derived from the physical system. Since the original motion is inherently complex, researchers use some simplifications so as to retain only the essential elements. Another methodology is to mimic animal motions by gathering data from these animals and attempting to replicate these recorded motions in robotic systems [45]. Although the data gives us important information about the locomotion and gaits, they run the risk of not being entirely representative of the inherent dynamics.

In robotics, different from nature, locomotion can also be achieved with wheeled platforms. As compared to legged systems, they are relatively easy to model and control. They also require less complex sensor hardware and software. However, wheeled locomotion needs smooth pathways cleared from major obstacles [22]. Such pathways are unfortunately exceptions in nature and rough terrain is more common. Moreover, legs can be multi-functional (running, climbing, jumping) which can be very important to be able to operate in unknown environments [13]. These advantages of legged systems motivate us to contribute to the existing literature on legged robotics.

It can be observed in nature that living creatures have successfully adapted to the surrounding environmental circumstances by developing various leg morphologies and dynamically stable gaits. In the literature, we can find that the spring loaded inverted pendulum (SLIP) model is successfully used to mimic dynamical legged locomotion [15, 21]. With this model, the leg of a running human can be modeled by an inverted pendulum with a spring in the center. The body of the human is represented by a mass on top of the pendulum whose mass is much lighter then the body. Through compression and decompression, like the tendons and muscles in the human body, the spring stores and releases kinetic energy which allows the cyclic dynamical motion of the system. Each cycle of this motion is basically divided into two subsequent phases of flight and stance as illustrated in Figure 1.1.

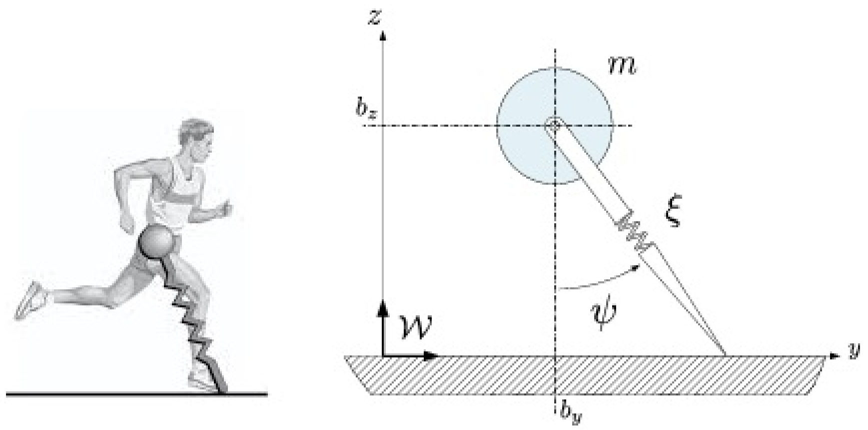

Figure 1.1: Modelling of the running with a spring-loaded inverted pendulum (SLIP) model with damping [6, 7].

With an addition of a damper accounting for the energy losses in the system, a more realistic model can also be obtained [3]. Because of SLIP approach's success in modeling the observed behavior of many legged systems, this model is widely used in the literature. One example is Raibert's single legged robots which are modeled and controlled using the SLIP model [33]. In 1980s, Raibert analyzed and also physically implemented two robots: One was able to achieve planar locomotion while the other in three dimensions. Another example uses a modified SLIP model (SLIP-T) to analyze and control planar hip torque actuated legged robot morphology. A template based control algorithm is proposed to achieve locomotion with relative stability [3, 4].

In order to achieve stable locomotion in legged robots, we need to address several challenges in control methods which should enable the regulation of the platform forward speed, apex height and position of the systems at desired values [11, 32, 33]. Furthermore, the unavoidable distortion of the robot attitude caused by the motion of the legs should be corrected by a control method. It is usually desirable to have a nearly constant body attitude because the body carries many sensors and hardware that may be sensitive to attitude changes. For example a vision system has to be pointing towards the direction of robot motion [29]. Inspired from some dynamically fast moving animals, the robot attitude can be regulated through the use of some additional body parts such as a tail [12]. Beside this naturally occurring solution to maintaining body attitude, engineering approaches not found in nature can also be used. One example uses a reaction wheel mechanism to control the body attitude of a four legged robot (quadruped) during the flight phase [11]. Reaction wheel mechanisms are efficient systems with regard to volume, mass, mechanical simplicity and reliability when compared to another engineering approach: the control moment gyroscope which also generates torque [43]. Both systems are mainly used in position control of satellites [14, 17, 23]. Because of their simplicity, the reaction wheel and similar inertial mechanism are also used as basic tools for control systems education [16, 38–40].

The complexity and also the energy consumption (due to the actuators) of a robotic system decrease with the reduction of the number of legs. However, the system becomes dynamically more unstable. Thus, to benefit from the advantages of fewer legs, there is an important body of work in the literature on bipedal and monopedal robot configurations and their control. For example, some research work on balancing the body of a humanoid robot use a methodology that converts inertial parts of the body into a simple inverted pendulum model combined with a reaction wheel to represent the effects of body parts. Then the simplified model is used to control the actual dynamics of the humanoid robot [24, 31]. In yet other work, researchers maintain the body balance of a humanoid robot by generating a trajectory for the legs. They calculate the Zero Moment Point (ZMP) using the changes in the inertial forces and angular momentums on the body during the motion. The trajectory for the legs is obtained to keep the robot's ZMP within the support points [26, 30, 42].

When the robot has a single leg with hopping locomotion, we obtain potentially a very energy efficient system. However, controlling the body attitude becomes difficult. During fast locomotion of such a monopod, keeping the desired body attitude is an interesting and not fully solved problem. Hence, in this thesis we aim to focus on achieving this with a novel air drag assisted reaction wheel approach.

1.2 Methodology and Contributions

We discussed our motivations in contributing to the body attitude control of a monopod robot. The particular work in [3] discusses stable locomotion of a monopod robot and proposes leg torque based actuation to supply energy to the system. This requires simpler actuator configuration but required a modification to the the classical SLIP model. Moreover, the study introduces a novel control algorithm to achieve stable locomotion for the modified system, achieving successful results. However, in order not to add extra degrees of freedom and complicate the model beyond its focus, they assume a stationary (infinite inertia) body attitude. This assumption makes the model unfeasible to implement for a practical monopod robot since every applied hip torque would have an effect on the body attitude during locomotion. In the context of the present thesis, we propose a modified version of torque actuated planar SLIP model by adding a reaction wheel assumed that it is placed exactly at the CoM of robot's body. This assumption enables us to use the existing template based controller from [3] because the reaction wheel motion does not have any effect on the locomotion dynamics. Additionally, as a contribution of the thesis, we achieve body attitude control through the use of the reaction wheel and cope with its limitations with the proposal of a "drag wheel" design.

The dynamic equations of our model depending on its hybrid phases (flight and stance) are derived using Newton-Euler methodology, so that we obtain all reaction forces between the body parts. We use these forces to detect the transition events between the phases. Since we have two actuators (hip and reaction wheel), we use different control algorithms for each from the literature to test our system. For the hip torque control we use two controllers:

1) Basic PD controller is used to keep the robot at standing position and,
2) Template based controller from [3, 4] is used for one-direction stable hopping locomotion.

For the reaction wheel controller we use PD and PID controllers to provide the necessary torque values that can compensate for the disturbances on the body caused by the hip torque actuation.

The physics based hybrid simulation results for the standing controller (that keeps the robot upright at a fixed position) show that our reaction wheel approach even with a simple PD controller is successful to compensate for the disturbances on body attitude. However, we also observe that for forward locomotion under the stable locomotion controller from [3,4], the approach become infeasible due to the unavoidable increase of the reaction wheel rotational velocity. As a main contribution of the present thesis, we overcome this problem with a novel air drag assisted reaction wheel design. We include the air drag concept [19] into our model by augmenting the reaction wheel by a simple impeller design to slow down the reaction wheel. As a result, our design enables stable locomotion compensating for the natural disturbances as well as possible external disturbances introduced. We demonstrate the feasibility of our idea with simulation results as well as physical test results.

1.3 Organization of the Thesis

The thesis is mainly divided into four parts. We start in the Introduction by giving our motivations and the background that is relevant with the thesis. Here, we also give our methodology and define our contributions to the existing robotics literature. In the second part, which includes Chapter 2 and 3, we present the background concepts such as reaction wheel pendulum and SLIP-T models that are important and addressed throughout the thesis. Chapter 4 and 5 present the main body of our contributions. In Chapter 4, we discuss the feasibility of a reaction wheel based approach for body attitude control during monopod locomotion, introducing a modified SLIP-T model an analyzing it under two different locomotion controllers. The results of the physics based hybrid simulation of the model demonstrate the feasibility of the reaction wheel approach. In Chapter 5, we present our novel air drag assisted reaction wheel concept and update the system equations accordingly. We analyze the new system, comparing its performance with the original reaction wheel results. The thesis ends with a conclusion in Chapter 6 where we summarize our work and main results and also address some possible future work.

CHAPTER 2

BACKGROUND: THE REACTION WHEEL PENDULUM

In this chapter, we review the reaction wheel pendulum from the literature by defining the system model, presenting the dynamical equations and reviewing some control methods for balancing the pendulum. Detailed analysis of the model and also physical implementation of a reaction wheel pendulum system can be found in [9]. Here, we will include the main results that will be used in the subsequent discussion and derivations.

2.1 System Model and Dynamics

The Figure 2.1 shows our system model which consists of a reaction wheel that is attached to the top end of an inverted pendulum. The bottom end of the pendulum is attached to a fixed pivot point around which the pendulum can rotate. The reaction wheel and the pendulum are represented as point masses m_{rw} and m_p respectively, at their geometric center. Table 2.1 shows all system parameters with their descriptions, which are used in the dynamical equations.

In the model, we define four state variables (Table 2.2) which are the pendulum angle θ, the reaction wheel angle β and their first derivatives with respect to time: $\dot{\theta}$, $\dot{\beta}$. In a physical implementation, we can measure these state variables using encoders. We should note that the encoder at the pivot point gives the angle θ with respect to the ground while the encoder on the reaction wheel gives the wheel angle with respect to the pendulum. Hence, if we want to get the angle β with respect to the ground, we need to add angle θ to the angle β.

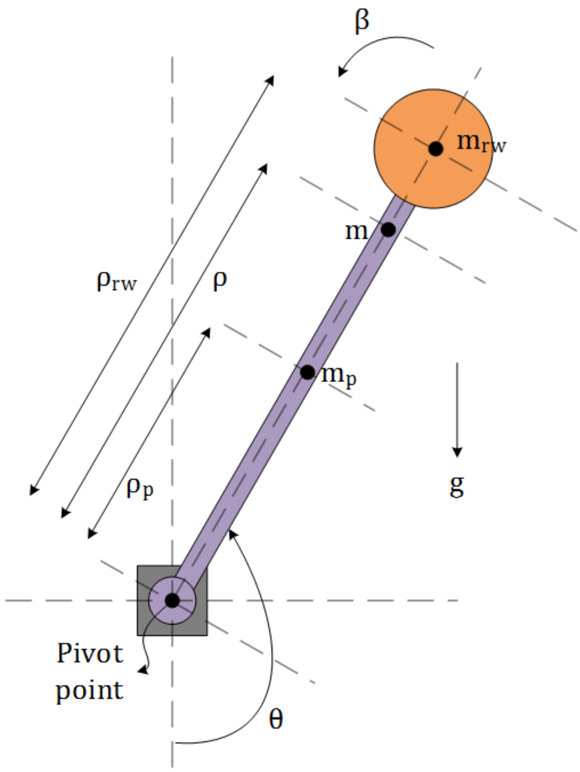

Figure 2.1: Pendulum with a reaction wheel model

Table 2.1: System Parameters of the reaction wheel pendulum model

Symbol	Definition
g	Gravity
m_{rw}	Mass of the reaction wheel
m_p	Mass of the pendulum
m	Total mass of the system including both pendulum and reaction wheel
I_{rw}	Moment of inertia of the reaction wheel with respect to its center of mass
I_p	Moment of inertia of the pendulum with respect to its center of mass
I	Moment of inertia of the system with respect to the center of mass of the system
ρ_{rw}	Distance between center of mass of the reaction wheel and the pivot point
ρ_p	Distance between center of mass of the pendulum and the pivot point
ρ	Distance between center of mass of the system and the pivot point

Table 2.2: State Variables of the reaction wheel pendulum model

Symbol	Definition
θ	Pendulum angle with respect to the ground coordinate system (vertical) in counter clockwise direction
β	Reaction wheel angle with respect to the pendulum's coordinate system in counter clockwise direction
$\dot{\theta}$	Pendulum angular velocity with respect to the ground coordinate system in counter clockwise direction
$\dot{\beta}$	Reaction wheel angular velocity with respect to the pendulum's coordinate system in counter clockwise direction

In the literature, there are two methods that are very common to derive the dynamical equations of a mechanical system. First one is the Euler-Lagrange method where we find the total energy of the system (kinetic and potential) and then derive the equations with respect to the generalized coordinates [41]. On the other hand, in Newton-Euler methodology we divide the system into rigid bodies, and for each rigid body we obtain the free-body-diagram with forces acting on it and the resulting accelerations [5]. If we require the information for reaction forces between rigid bodies, we must use the Newtonian approach; but Lagrangian is an easier method to get the dynamical equations and capture the motion of the unconstrained degrees of freedom in the system.

In order to understand the reaction wheel concept and derive the system model, in this chapter we use the Lagrangian method, because of its derivational simplicity. However, in later chapters since we need the force information between the rigid bodies, we derive the dynamical equations using the Newton-Euler methodology. As a consequence, in the present thesis one can review both methodologies applied on very similar systems.

Using the Lagrangian method, let us define θ and β as the generalized coordinates of the system.

$$q = [\theta, \beta]^T \tag{2.1}$$

We calculate the total kinetic energy of the system by adding the individual kinetic energies of the pendulum and reaction wheel respectively as [9]:

$$K = \frac{1}{2}I_{rw}(\dot{\beta} + \dot{\theta})^2 + \frac{1}{2}I\dot{\theta}^2 \tag{2.2}$$

where

$$I = (I_p + m_{rw}\rho_{rw}^2 + m_p\rho_p^2) \tag{2.3}$$

Note that, while calculating the kinetic energy of the pendulum, we should consider the added mass m_{rw} of the reaction wheel as well. Hence, in the second term of the kinetic energy equation 2.2, the moment of inertia I is calculated using both masses and the distance of the center of mass from the pivot point as shown in equation 2.3.

When there are no external forces, ideally system could be stabilized either upright or hanging down. However, due to the fact that the system is in an unstable equilibrium point in the upright position, even very small disturbances affect its stability. On the other hand, the hanging down position is asymptotically stable under the influence of gravity. Hence, in our model let us define this hanging down position as having zero potential energy and calculate the total potential energy of the system at any other angle based on this definition as [9]:

$$P = mg\rho(1 - \cos\theta) \tag{2.4}$$

where

$$\rho = \frac{(m_{rw}\rho_{rw} + m_p\rho_p)}{(m_{rw} + m_p)} \tag{2.5}$$

Note that because of our model definition, the motion of the reaction wheel has no effect on the overall potential energy.

Subtracting total potential energy 2.4 from the total kinetic energy 2.2 of our system, we find the Lagrangian function as:

$$L = K - P \tag{2.6}$$

The equations of motion are expressed in terms of the Lagrangian function in the following form:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q_k}}) - \frac{\partial L}{\partial q_k} = \tau_k, \quad where\ k = 1, 2. \tag{2.7}$$

Taking the partial derivative of the Lagrangian function with respect to the system variables, θ and β we find:

$$\begin{aligned}\frac{\partial L}{\partial \dot{\theta}} &= I_c\dot{\theta} \\ \frac{\partial L}{\partial \theta} &= -mg\rho_c sin(\theta) \\ \frac{\partial L}{\partial \dot{\beta}} &= I_{rw}(\dot{\beta}+\dot{\theta}) \\ \frac{\partial L}{\partial \beta} &= 0\end{aligned} \tag{2.8}$$

If external forces are applied to the system, these forces/torques must be incorporated into the Lagrange equations. In our model, only an external torque τ is applied to the reaction wheel. Based on Newton physics, the applied τ generates a torque around the pivot point of the pendulum with equal magnitude but in the opposite direction. Finally, with this torque accounted, we obtain

$$\begin{aligned}\ddot{\theta} &= \frac{-\tau - mg\rho sin(\theta)}{I} \\ \ddot{\beta} &= \frac{\tau}{I_{rw}} + \frac{\tau + mg\rho sin(\theta)}{I},\end{aligned} \tag{2.9}$$

as the final system dynamical equations.

2.2 Control Methods

2.2.1 Controlling Pendulum Angle

In order to stabilize the pendulum at the upright position, let us linearize the dynamical equations 2.9 around $\theta = \pi$. Here we follow the notation from [9] and to avoid a constant π term which needs to be included in the linearized system equations, let us define $\bar{\theta}$ as the angle deviation from π. Linearization of the system equation 2.9 around $\bar{\theta} = 0$ and using $\bar{\theta}$ in the equations, we obtain

$$\begin{aligned}\ddot{\bar{\theta}} &= \frac{-\tau + mg\rho\bar{\theta}}{I} \\ \ddot{\beta} &= \frac{\tau}{I_{rw}} + \frac{\tau - mg\rho\bar{\theta}}{I}.\end{aligned} \tag{2.10}$$

A simple PD controller can be used to stabilize the pendulum at the upright position, so we give the torque input τ as:

$$\tau = -k_{pp}\bar{\theta} - k_{dp}\dot{\bar{\theta}} \tag{2.11}$$

where k_{pp} and k_{dp} are proportional and derivative coefficients, respectively. Using this control input 2.11 in the linearized equations 2.10, the closed loop system for the pendulum angle dynamics become:

$$\ddot{\bar{\theta}} - \frac{k_{dp}}{I}\dot{\bar{\theta}} - (\frac{mg\rho}{I} + k_{pp})\bar{\theta} = 0 \tag{2.12}$$

When the pendulum initially has a non-zero angle $\bar{\theta}_0$, the system can end up in two configurations. If the controller can eliminate all steady-state error while converging to the top equilibrium, the reaction wheel velocity will converge to a constant value. However, in case the controller cannot eliminate this steady-state error, the right hand side of the equation 2.12 is not equal to zero but exhibits a residual constant torque as given by

$$\ddot{\bar{\theta}} - \frac{k_{dp}}{I}\dot{\bar{\theta}} - (\frac{mg\rho}{I} + k_{pp})\bar{\theta} = \tau_d. \tag{2.13}$$

For this case, the controller, which tries to unsuccessfully counter this residual torque, would cause the reaction wheel velocity to increase in an unbounded manner. Both of these behaviors can be observed analytically by taking the Laplace transform of the equations for the pendulum and reaction wheel. The details can be found in [9]. In physical experiments, starting from an initial angle $\bar{\theta}_0$ and having a nonzero τ_d would cause a saturation for the motor velocity and consequently, the basic PD controller 2.11 would fail to balance the pendulum at the upright position.

2.2.2 Controlling Pendulum Angle and Reaction Wheel Velocity

To avoid the saturation limit that the reaction wheel could reach, in this section we present a method which controls both pendulum angle and reaction wheel velocity. But before the presentation of the control method, first we need to show the controllability of the system with single input τ, based on the states that we define.

$$
\begin{aligned}
&X = [x_1, x_2, x_3, x_4]^T \\
&where; \\
&x_1 = \theta \qquad\qquad x_3 = \beta \\
&x_2 = \dot{\theta} \qquad\qquad x_4 = \dot{\beta}
\end{aligned}
\tag{2.14}
$$

Writing the equation 2.10 in state space form, we get:

$$
\begin{bmatrix} \dot{\bar{\theta}} \\ \ddot{\bar{\theta}} \\ \dot{\beta} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mg\rho}{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-mg\rho}{I} & 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} \bar{\theta} \\ \dot{\bar{\theta}} \\ \beta \\ \dot{\beta} \end{bmatrix} + \tau \begin{bmatrix} 0 \\ \frac{-1}{I} \\ 0 \\ \frac{I.I_{rw}}{I + I_{rw}} \end{bmatrix}
\tag{2.15}
$$

Hence, the controllability matrix is:

$$
C(A, B) = \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{I} & 0 & \frac{-mg\rho}{I^2} \\ \frac{-1}{I} & 0 & \frac{-mg\rho}{I^2} & 0 \\ 0 & \frac{I.I_{rw}}{I + I_{rw}} & 0 & 0 \\ \frac{I.I_{rw}}{I + I_{rw}} & 0 & 0 & 0 \end{bmatrix}
\tag{2.16}
$$

Since the matrix $C(A, B)$ is full rank, the system is controllable. Hence, using single variable input τ we can define our next control law as [9]:

$$
\tau = -k_{pp}\bar{\theta} - k_{dp}\dot{\bar{\theta}} - k_{pp}((\dot{\beta} + \dot{\bar{\theta}}) - w_d)
\tag{2.17}
$$

which tries to keep the reaction wheel speed at desired level w_d while balancing the pendulum at the inverted position.

Note that this analysis is based on the linearization of the original nonlinear system around the top equilibrium. Therefore, the method presented here can only be applied when the system is to be balanced on this equilibrium point. It can be easily observed that if we attempt to balance the system at a non-zero angle (non-vertical configuration), we will not be able to keep the reaction wheel velocity bounded. However, in

Chapter 5 we show that balancing the pendulum at a constant non-zero angle can be achieved using the drag wheel concept.

2.3 Simulation of the Model and SimMechanics Verification

We simulate the reaction wheel pendulum model using both Matlab [28] and SimMechanics [1] environments. In the literature, SimMechanics is highly used and accurate to simulate multibody mechanical systems by constructing them from blocks of bodies, joints, constraints and force elements [1]. The dynamical equations are formulated using these blocks and the motion of the complete system is generated by solving these dynamical equations. Figure 2.2 shows the pendulum with reaction wheel model generated by SimMechanics' blocks. The model consists of a stationary base, two rotational joints, pendulum, reaction wheel and controller block which supplies the required torque to control the pendulum angle.

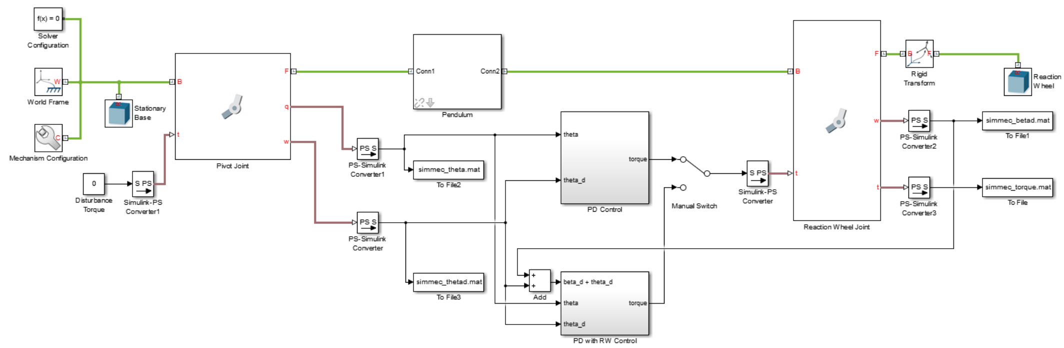

Figure 2.2: Pendulum with reaction wheel model generated by SimMechanics' blocks

In Matlab simulation, first the states of the reaction wheel pendulum model are initialized as shown in the flowchart (Figure 2.3). At each iteration, the successive states are found by solving the dynamical equations of the system using an ordinary differential equation solver (ode15s), while the controller regulates the system to reach the desired states.

The system parameters and initial state values are explicitly given in Table 2.3. Accordingly, simulations are performed starting from 10^0 offset from the inverted vertical position of the pendulum. Snapshots from animations for both implementations

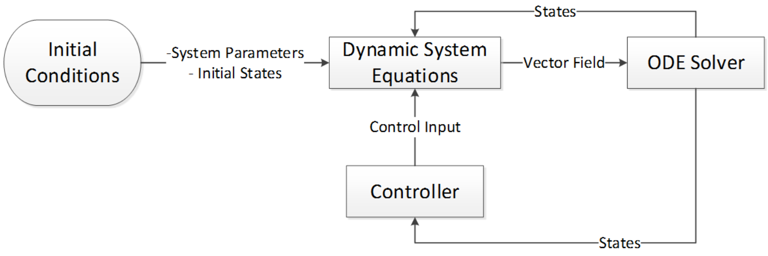

Figure 2.3: Flowchart of the Matlab Simulation for the reaction wheel pendulum

are shown in Figure 2.4.

Table 2.3: (a) Initial States and (b) System Parameters

(a)

Symbol	Initial Value and Unit
θ	$\frac{190\pi}{180}$ rad
β	0 rad
$\dot{\theta}$	0 $\frac{rad}{s}$
$\dot{\beta}$	0 $\frac{rad}{s}$

(b)

Symbol	Value and Unit
g	9.81 $\frac{m}{s^2}$
m_{rw}	1 kg
m_p	0.03 kg
I_{rw}	0.0002 $kg.m^2$
I_p	0.0062556 $kg.m^2$
ρ_{rw}	0.5 m
ρ_p	0.25 m

2.3.1 Controlling Pendulum Angle

In Section 2.1, we derive the dynamical equations of the pendulum with reaction wheel system and provide a simple PD controller in order to keep the pendulum at the upright position. Based on the mathematical analysis, here we simulate the system using Matlab and SimMechanics. We present some simulation results in order to compare two implementations.

The top graph in Figure 2.5 shows the angular change of the pendulum with respect time for both implementations. Starting at an initial angle, PD controller brings and keeps the pendulum at 180^o successfully. The bottom graph shows that the difference of the angle values between Matlab and SimMechanics is on the order of 10^{-11}. The difference is too small that it is even lesser than the maximum step length, which is adjusted as 10^{-3}, for the ODE solver.

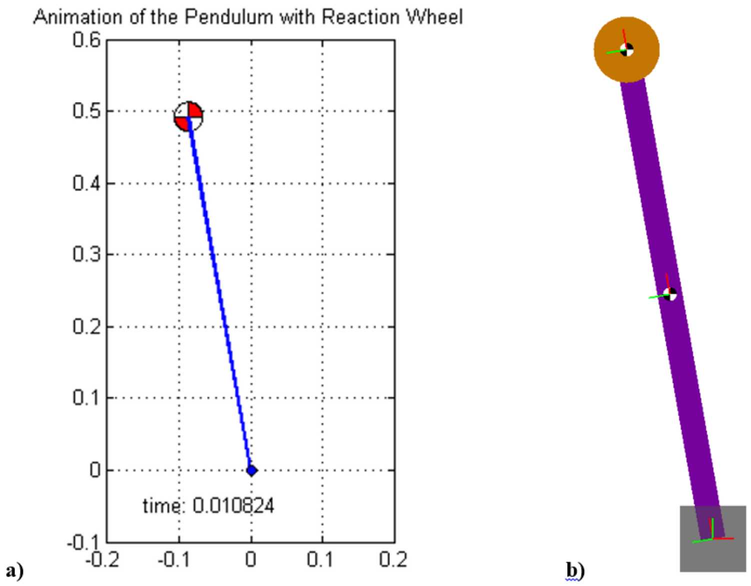

Figure 2.4: Snapshots from animations of pendulum with reaction wheel using a) Matlab and b) SimMechanics

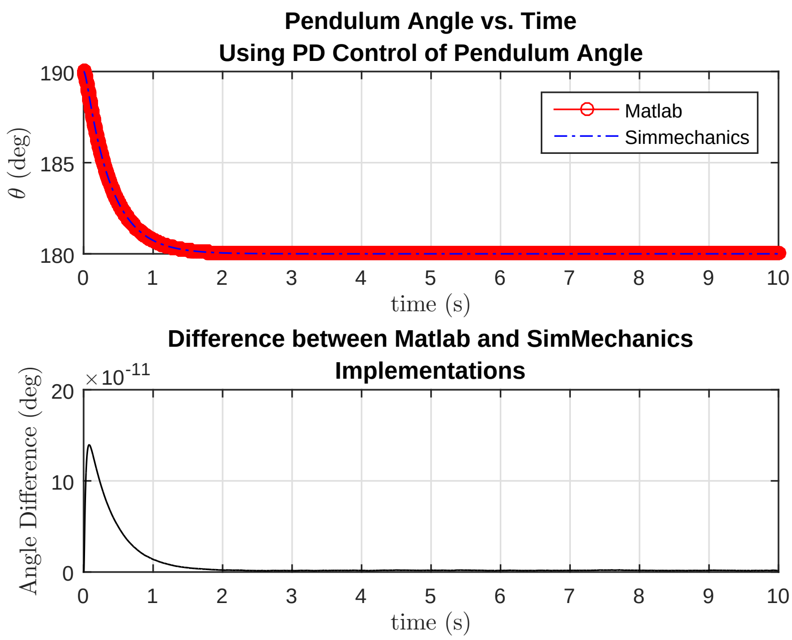

Figure 2.5: (top) Pendulum angle with respect to time and (bottom) the difference between Matlab (red) and SimMechanics (blue) simulations

The top graph in Figure 2.6 shows the angular velocity values of the pendulum angle with respect time for both implementations. The bottom graph indicates that the difference of the angular velocity values between Matlab and SimMechanics is on the order of 10^{-11}.

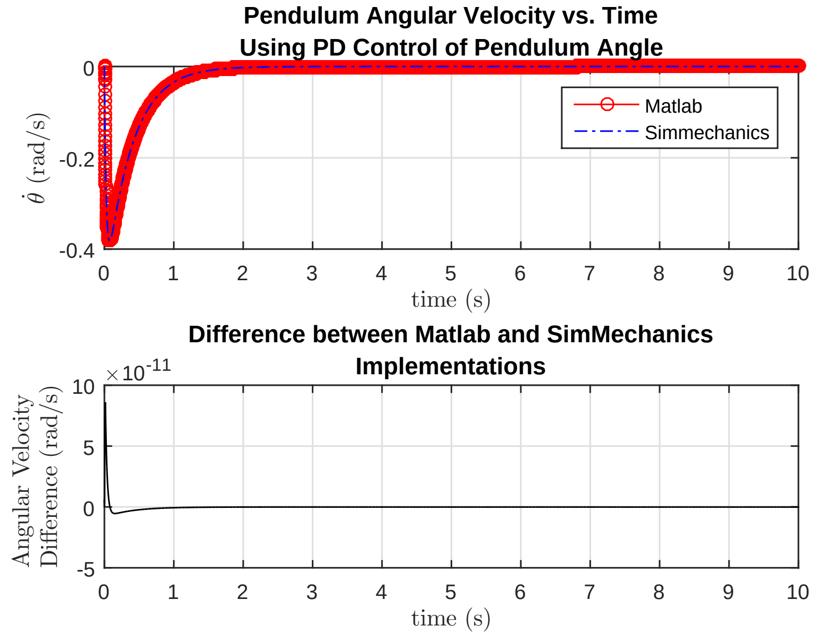

Figure 2.6: (top) Angular velocity of the pendulum with respect to time and (bottom) the difference between Matlab (red) and SimMechanics (blue) implementations

The initial 10^o angular offset from the upright position (which is $\theta = 180^o$) causes the reaction wheel to gain a constant angular velocity at the steady state (top graph in the Figure 2.7). Comparing both implementations based on the reaction wheel angular velocity data, again we observe that the difference between Matlab and SimMechanics is very small (on the order of 10^{-7}).

In order to simulate the physical experiment case in which the controller cannot eliminate a steady-state error, a small disturbance torque is externally given into the system to disturb the pendulum angle continuously. We observe that even under the continuous disturbance, the PD controller brings the pendulum to the desired angle (Figure 2.8). However, limitlessly the reaction wheel velocity increases to compensate the angular effect caused by the disturbance torque (Figure 2.9).

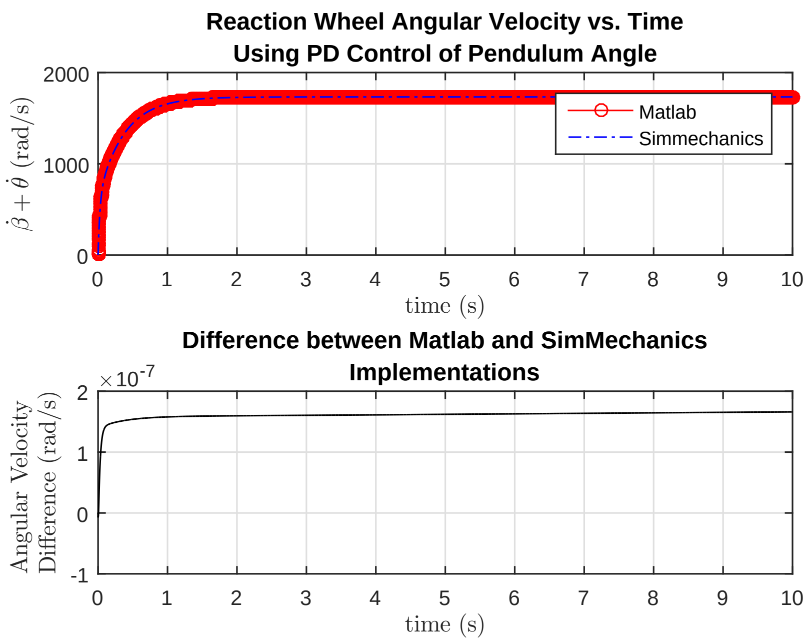

Figure 2.7: Angular velocity of the reaction wheel with respect to time and the difference between Matlab and SimMechanics implementations

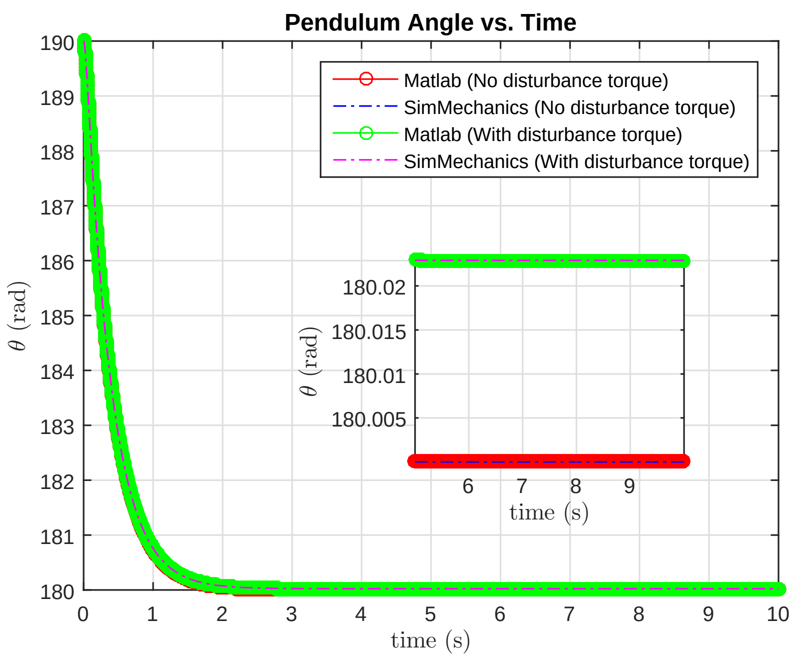

Figure 2.8: Pendulum angle with respect to time with (green and magenta) or without (red and blue) some disturbances on the pendulum

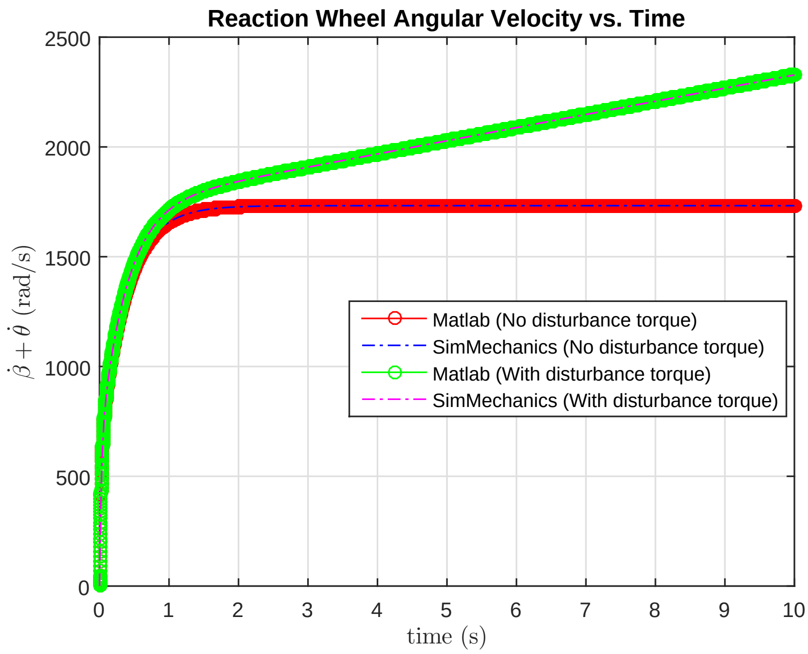

Figure 2.9: Angular velocity of the reaction wheel with respect to time with (green and magenta) or without (red and blue) some disturbances on the pendulum

2.3.2 Controlling Pendulum Angle and Reaction Wheel Velocity

In the previous section we observe that if there is a residual torque which affects on the pendulum angle, the reaction wheel velocity increases in an unbounded manner in order to counter this torque. As we discuss in Section 2.2.2, to keep the reaction wheel velocity at some desired level, we update the PD controller by adding the reaction wheel velocity with an appropriate coefficient (equation 2.17). Using the updated PD controller in the simulations, the pendulum is balanced at the upright position (Figure 2.10) as well as the reaction wheel velocity is brought to a constant level (Figure 2.11).

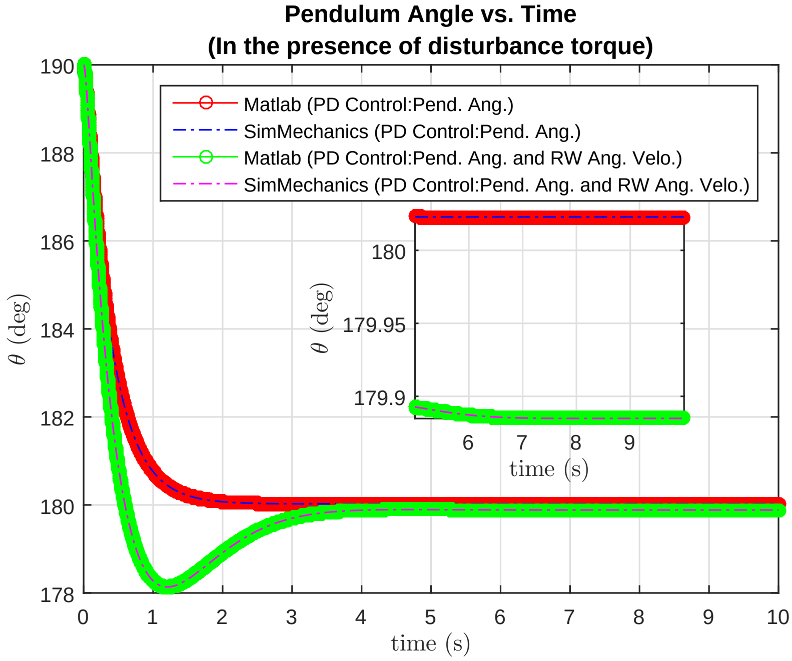

Figure 2.10: Pendulum angle with respect to time, comparing control methods in the presence of some disturbances on the pendulum

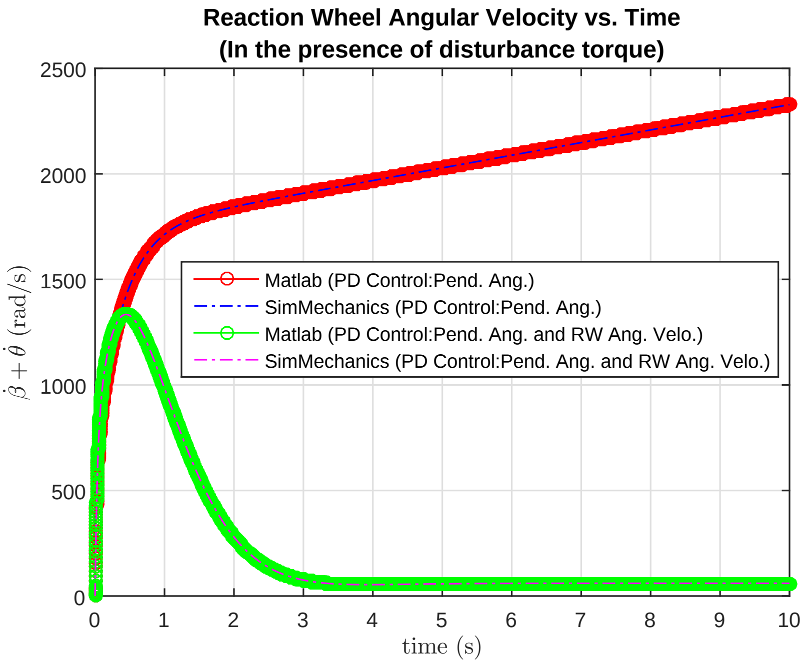

Figure 2.11: Angular velocity of the reaction wheel with respect to time, comparing control methods in the presence of some disturbances on the pendulum

CHAPTER 3

BACKGROUND: THE PLANAR TORQUE ACTUATED SPRING MASS HOPPER

In this chapter, we review the planar torque actuated spring mass hopper (SLIP-T) model and its locomotion controller introduced by Ankaralı [3, 4]. In chapters 4 and 5, we use this SLIP-T model with the same controller but extent it with the reaction wheel. Basically, SLIP-T is a modified version of the well-known spring loaded inverted pendulum (SLIP) model. The summary of both models, derivation of dynamic equations for SLIP-T and also the necessary background for the template control of SLIP-T model are presented below.

3.1 System Model and Dynamics

3.1.1 The Planar SLIP Model with Damping

The SLIP model (Figure 3.1) is basically a system with a point mass (body) attached to a massless inverted pendulum (leg) with a spring at the middle. Due to the energy loss on the spring during its motion, the spring is modeled realistically as having a damping factor parallel to the spring stiffness. Based on contact/no contact of the bottom point of the pendulum (toe) with the ground, the continuous dynamic locomotion of the system consists of two hybrid stages: flight and stance.

The flight is the phase where the leg has no contact with the ground. Hence, based on the initial states, the system follows a trajectory which is a simple ballistic flight of the point mass. This phase can be divided into two sub-phases: ascent and descent.

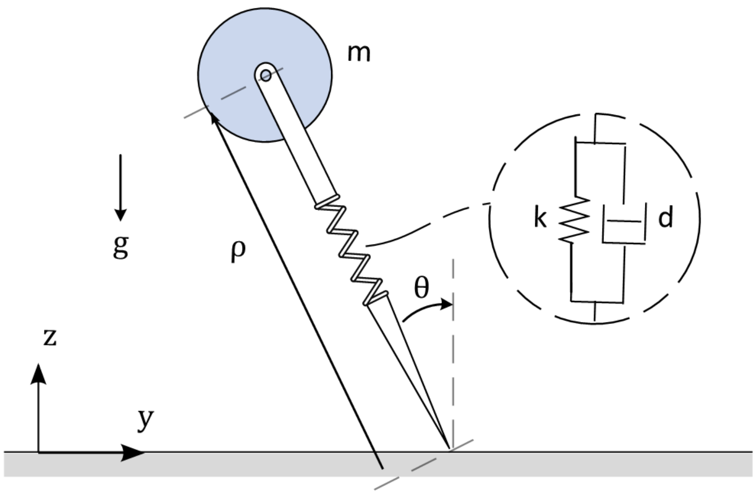

Figure 3.1: Basic SLIP model with damping [3]

At the end of decompression of the spring, the system lifts off from the ground and the ascent stage continues until the most reachable height (apex) is achieved. After apex, the system starts to descent upon gravity. When the toe has a contact with the ground, the stance phase starts.

The stance phase can be also analyzed in two sub-phases: compression and decompression. The compression phase starts with the touchdown of the toe to the ground. Then, the energy of the system compresses the spring up to a bottom point where the spring cannot be compressed more. After that bottom point, the potential energy stored on the spring is converted into kinetic energy which decompresses the spring till the lift-off event. Starting with an apex state, Figure 3.2 shows a complete one-stride of the system with the transition events that divide phases and sub-phases.

3.1.2 The Planar Model and Dynamics of The Torque Actuated Spring Mass Hopper

SLIP-T model (3.3) is structurally very similar to the SLIP model which is introduced in the previous section. The main difference between models is that the leg in SLIP-T model can be actuated by a hip torque, although in SLIP model the actuation of the leg is completely passive. Hence, this controllable torque can supply the necessary decompression during stance and it can bring the leg to the desired angle

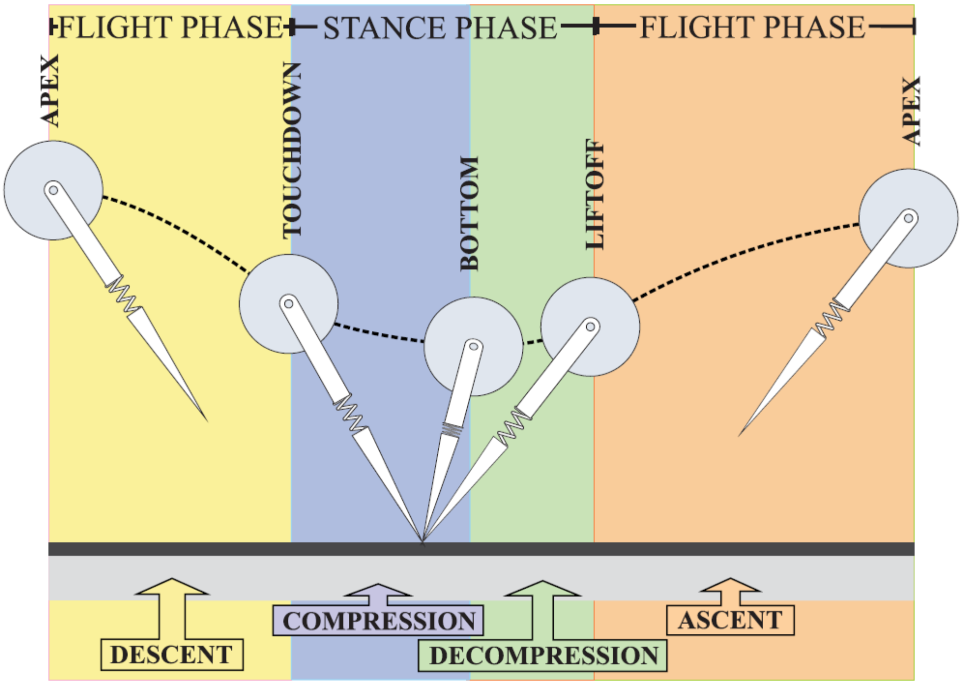

Figure 3.2: One-stride of the SLIP system with the the transition events (boundaries) that divide phases (shaded regions) [3]

during flight. In the model, in order to keep the dynamical equations mathematically

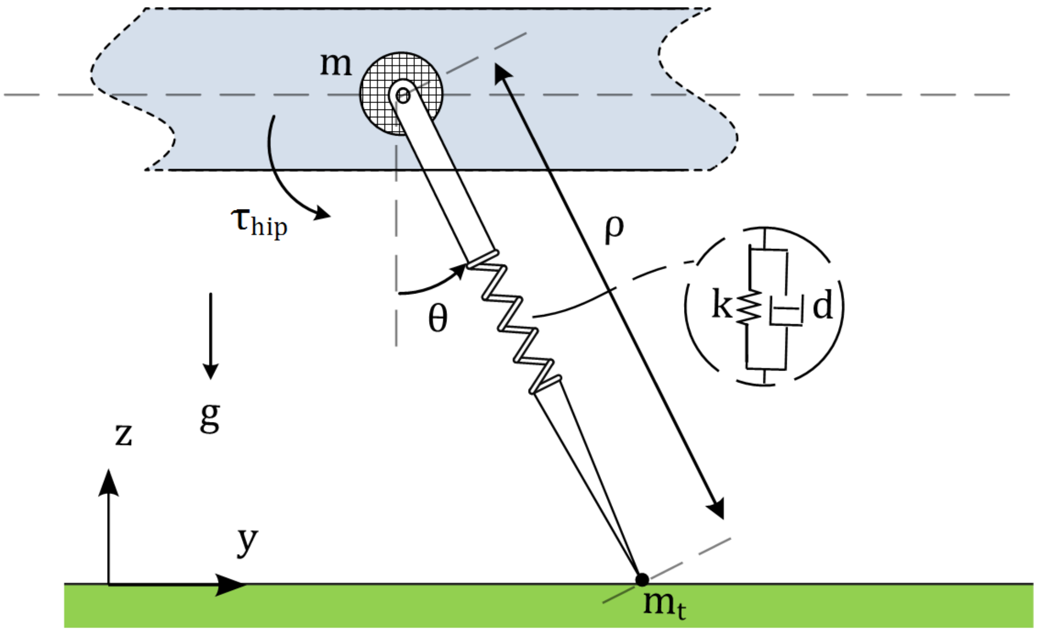

Figure 3.3: The planar model of the torque actuated spring mass hopper

simple, Ankaralı makes two assumptions (hence, the control become simpler) while providing necessary convergence to physical systems. First assumption constrains the orientation of the rigid body with a mass m_b to be horizontal. This establish the torque actuation without adding extra degree of freedom to the system. Second assumption introduces a relatively small point mass with respect to the body mass $(m_t << m_b)$ to handle the flight dynamics of the leg.

Since we require reaction forces between the joints, especially to detect transition events, our approach to derive the dynamical equations of systems both here and following chapters is Newton-Euler methodology as Ankaralı.

In the model, there are eight state variables which are used to define positions of body (y_b, z_b) and toe (y_t, z_t) in ground coordinate frame, and their first derivatives $\dot{y}_b, \dot{z}_b, \dot{y}_t, \dot{z}_t$ respectively. Mid-state variables are calculated based on states and system parameters which determine the physical properties of the system. Definitions of state, mid-state variables and parameters of the system model are explicitly listed in Table 3.1, 3.2 and 3.3 respectively. Here please note that, although our symbol notation for the variables and parameters are rather different from Ankaralı, the dynamical equations are exactly the same with respect to the model. For the sake of the completeness of our work, we use the same symbols for the same states throughout the thesis.

Table 3.1: State Variables of the SLIP-T model

Symbol	Definition
y_b	Horizontal body position with respect to the ground coordinate system
z_b	Vertical body position with respect to the ground coordinate system
y_t	Horizontal toe position with respect to the ground coordinate system
z_t	Vertical toe position with respect to the ground coordinate system
$\dot{y}_b$	Horizontal velocity of the body with respect to the ground coordinate system
$\dot{z}_b$	Vertical velocity of the body with respect to the ground coordinate system
$\dot{y}_t$	Horizontal velocity of the toe with respect to the ground coordinate system
$\dot{z}_t$	Vertical velocity of the toe with respect to the ground coordinate system

Table 3.2: Mid-State Variables of the SLIP-T model

Symbol	Definition
ρ	Length of the leg
$\dot{\rho}$	Radial velocity of the leg
θ	Angle of the leg with respect to the body vertical coordinate system in counter clockwise direction
$\dot{\theta}$	Angular velocity of the leg with respect to the body vertical coordinate system in counter clockwise direction
$F_y^{leg,body}$	Reaction force in horizontal direction between leg and body to the calculated value
$F_z^{leg,body}$	Reaction force in vertical direction between leg and body to the calculated value
F_y^{ground}	Ground reaction force in horizontal direction to the calculated value
F_z^{ground}	Ground reaction force in vertical direction to the calculated value
F_{spring}	Squeezing and stretching force on the spring
τ^{hip}	Torque value that is applied on the hip of the leg

Table 3.3: System Parameters of the SLIP-T model

Symbol	Definition
g	Gravity
k	Spring stiffness coefficient
d	Spring damping coefficient
m_b	Mass of the body
m_t	Point mass of the toe

In order to derive the dynamical equations using Newton-Euler force analysis, we divide the model into two parts: body and leg, and analyze all internal-external and inertial forces between them. Figure 3.4 shows the free body diagram (fbd) of the body which is same for both phases. Although a hip torque due to the leg is shown in the figure, since orientation of the body is assumed as fixed, the torque effect on the body is neglected while determining the dynamical equations. (3.1) shows equations based on the fbd of the body.

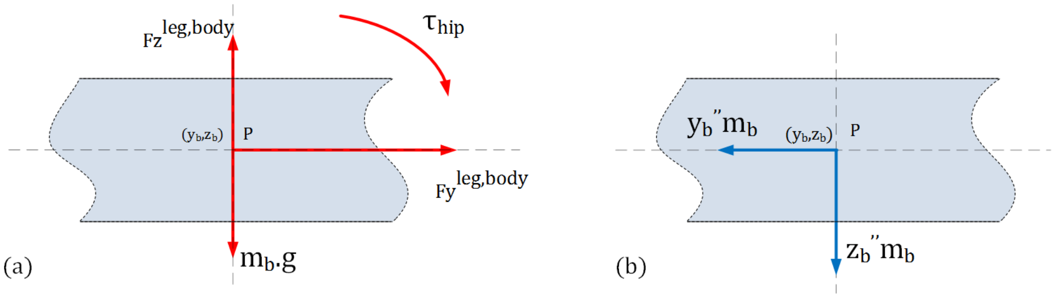

Figure 3.4: Free-body diagrams of (a) internal-external, and (b) inertial forces of the SLIP-T body in both flight and stance phases

$$
\begin{aligned}
m_b\ddot{y}_b - F_y^{leg,body} &= 0 \\
m_b\ddot{z}_b - F_z^{leg,body} &= -m_b g
\end{aligned} \tag{3.1}
$$

In flight phase, hip torque control tries to bring and keep the leg in the desired touch-down angle. Since we assume the leg as massless, only the toe mass is affected by gravity. We present equations based on the fbd (Figure 3.5) of the leg during the flight phase in (3.2).

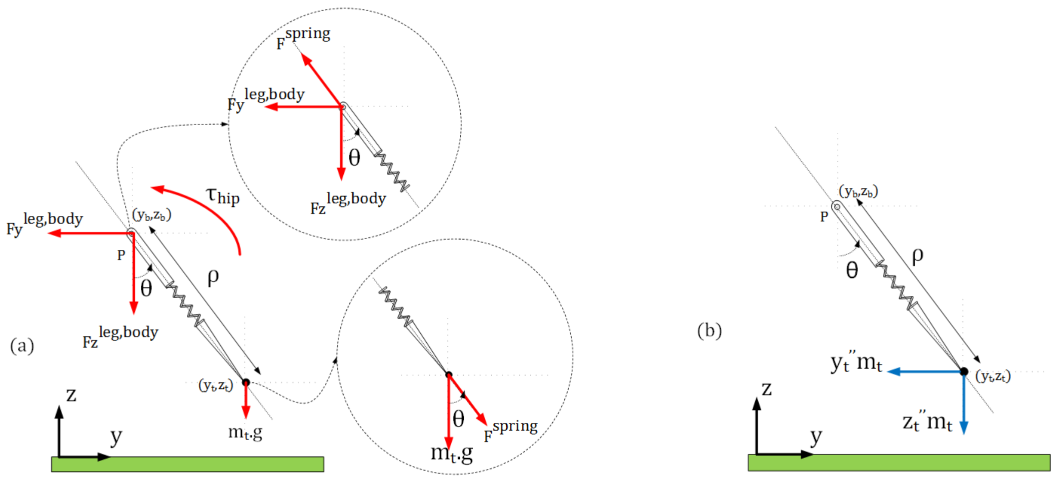

Figure 3.5: Free-body diagrams of (a) internal and external forces and (b) inertial forces of the leg in flight phase

$$
\begin{aligned}
m_t\ddot{y}_t + F_y^{leg,body} &= 0 \\
m_t\ddot{z}_t + F_z^{leg,body} &= -m_t g \\
F_z^{leg,body}cos(\theta) - F_y^{leg,body}sin(\theta) &= F^{spring} \\
F^{spring} &= -k(\rho - \rho_0) - d\dot{\rho} \\
m_t\rho cos(\theta)\ddot{y}_t + m_t\rho sin(\theta)\ddot{z}_t &= \tau^{hip} - m_t g\rho sin(\theta)
\end{aligned}
\tag{3.2}
$$

In stance phase, it is assumed that the toe is fixed on a point on the ground and the body follows a trajectory around this point due to the radial spring damper-force. The fbd and equations of the leg during stance is given in Figure 3.6 and (3.3) respectively.

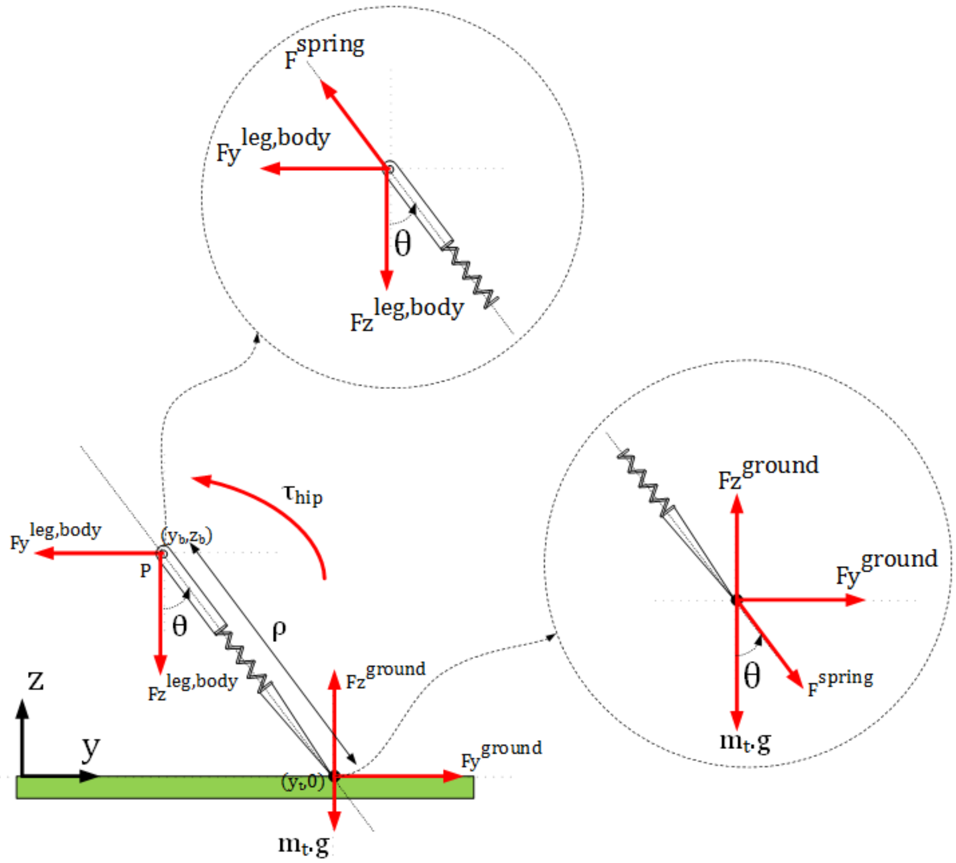

Figure 3.6: Free-body diagram of the internal forces of the leg in stance phase. There is no external force since leg is massless

$$
\begin{aligned}
F_y^{ground} - F_y^{leg,body} &= 0 \\
F_z^{ground} - F_z^{leg,body} &= m_t g \\
F_y^{leg,body} \rho cos(\theta) + F_z^{leg,body} \rho sin(\theta) &= -\tau^{hip} \\
F_z^{leg,body} cos(\theta) - F_y^{leg,body} sin(\theta) &= F^{spring} \\
F^{spring} &= -k(\rho - \rho_0) - d\dot{\rho}
\end{aligned}
\tag{3.3}
$$

To summarize; when we gather state variables into left hand side of the dynamical equations, the overall SLIP-T model neatly become:

During Flight:

$$
\begin{aligned}
\ddot{y}_b &= 0 \\
\ddot{z}_b &= -g \\
\ddot{y}_t &= \frac{\tau cos(\theta)}{m_t \rho} - \frac{sin(\theta)(k(\rho - \rho_0) + d\dot{\rho})}{m_t} \\
\ddot{z}_t &= \frac{\tau sin(\theta)}{m_t \rho} + \frac{cos(\theta)(k(\rho - \rho_0) + d\dot{\rho})}{m_t} - g
\end{aligned}
\tag{3.4}
$$

During Stance:

$$
\begin{aligned}
\ddot{y}_b &= \frac{-\tau cos(\theta)}{m_b \rho} + \frac{sin(\theta)(k(\rho - \rho_0) + d\dot{\rho})}{m_b} \\
\ddot{z}_b &= \frac{-\tau sin(\theta)}{m_b \rho} - \frac{cos(\theta)(k(\rho - \rho_0) + d\dot{\rho})}{m_b} - g \\
\ddot{y}_t &= 0 \\
\ddot{z}_t &= 0
\end{aligned}
\tag{3.5}
$$

In order to simplify dynamical equations further and make analysis more efficient, some body of work in the legged systems literature [3, 7], including Ankaralı, use nondimensionalization approach. The approach removes units in the equations by converting the physical variables into dimensionless versions. For the conversion, Ankaralı redefines the dimensionless time $(\bar{t})$ as [3]:

$$
\bar{t} = \frac{t}{\lambda} \tag{3.6}
$$

where;

$$
\lambda = \sqrt{\frac{\rho_0}{g}}) \tag{3.7}
$$

Table 3.4: Table of conversion from physical to dimensionless variables for the SLIP-T model

Dimensionless Variables	Conversion
$\ddot{\bar{y}}$	$:= \frac{\ddot{y}}{g}$
$\ddot{\bar{z}}$	$:= \frac{\ddot{z}}{g}$
$\bar{\rho}$	$:= \frac{\rho}{\rho_0}$
$\dot{\bar{\rho}}$	$:= \frac{\lambda\dot{\rho}}{g}$
$\bar{\theta}$	$:= \theta$
$\bar{F}^{spring}$	$:= \frac{F^{spring}}{mg}$
$\bar{\tau}$	$:= \frac{\tau}{mg\rho_0}$
n_t	$:= \frac{m_t}{m}$
κ	$:= \frac{k_s\rho_0}{mg}$
c	$:= \frac{d\rho_0}{\lambda mg}$, (where $\lambda = \sqrt{\frac{\rho_0}{g}}$)

and time derivative of state variables as [3]:

$$(\frac{d}{d\bar{t}})^n = \lambda^n(\frac{d}{dt})^n \tag{3.8}$$

Scaling all variables with conversions based on the Table 3.4, dynamical equations for the SLIP-T model in dimensionless form become as shown in (3.9) and (3.10).

During Flight (in dimensionless form):

$$\begin{aligned}
\ddot{\bar{y}}_b &= 0\\
\ddot{\bar{z}}_b &= -1\\
\ddot{\bar{y}}_t &= \frac{\bar{\tau}cos(\theta)}{n_t\bar{\rho}} - \frac{sin(\theta)(\kappa(\bar{\rho}-1)+c\dot{\bar{\rho}})}{n_t}\\
\ddot{\bar{z}}_t &= \frac{\bar{\tau}sin(\theta)}{n_T\bar{\rho}} + \frac{cos(\theta)(\kappa(\bar{\rho}-1)+c\dot{\bar{\rho}})}{n_t} - 1
\end{aligned} \tag{3.9}$$

During Stance (in dimensionless form):

$$
\begin{aligned}
\ddot{\bar{y}}_b &= \frac{-\bar{\tau} cos(\theta)}{\bar{\rho}} + sin(\theta)(\kappa(\bar{\rho}-1) + c\dot{\bar{\rho}}) \\
\ddot{\bar{z}}_b &= \frac{-\bar{\tau} sin(\theta)}{\bar{\rho}} - cos(\theta)(\kappa(\bar{\rho}-1) + c\dot{\bar{\rho}}) - 1 \\
\ddot{\bar{y}}_t &= 0 \\
\ddot{\bar{z}}_t &= 0
\end{aligned}
\tag{3.10}
$$

3.2 Control of SLIP-T for a Stable Locomotion

3.2.1 Control Dynamics

In the literature, "Deadbeat Gait Control" is one of the methods to perform stable locomotion for the SLIP model. However, the difference in dynamics of SLIP and SLIP-T makes it unavailable to use this control method presented in [3] detailed. Going around this problem, Ankaralı provided a novel approach to perform a gait control for the SLIP-T dynamics [3], by introducing a virtual leg (Figure 3.7) which kind of makes the system perform as the SLIP model.

Here, in order to define virtual toe position we introduce two more state variables (Table 3.5) and respectively mid-state variables (Table 3.6) into the SLIP-T model. Please note that, in this section we continue our dimensionless approach started at the end of previous section, so that we also define these new variables in dimensionless form. As you may notice we use the bar notation over the variables and parameters to indicate that they are dimensionless.

Table 3.5: Dimensionless Virtual State Variables

Symbol	Definition
$\bar{y}_v$	Virtual toe position in horizontal direction with respect to the ground coordinate system
$\bar{z}_v$	Virtual toe position in vertical direction with respect to the ground coordinate system

Now, let us achieve a desired touchdown angle ($\bar{\psi}^*_{td}$) and leg length ($\bar{\xi}^*_{td}$) for the virtual

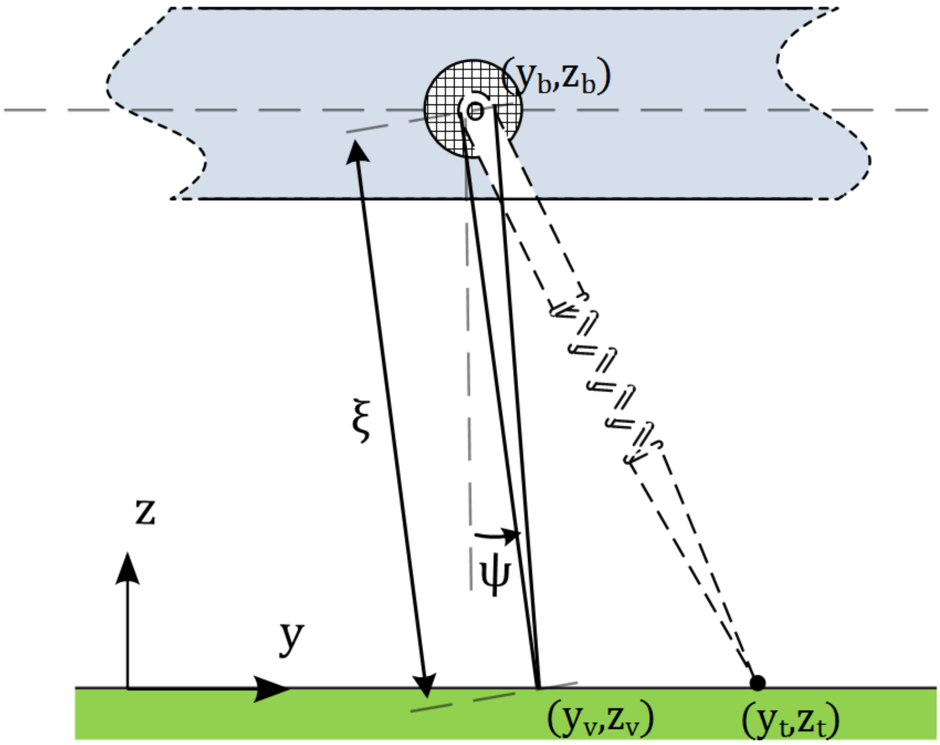

Figure 3.7: The planar SLIP-T model including virtual leg

Table 3.6: Dimensionless Virtual Mid-State Variables

Symbol	Definition
$\bar{\xi}$	length of the virtual leg
$\bar{\psi}$	Angle of the virtual leg with respect to the body vertical coordinate system in counter clockwise direction

leg during flight. Then for this case, by using simple trigonometry we can calculate the touchdown angle $(\bar{\theta}_{td})$ for the physical leg as [3]:

$$\bar{\theta}_{td} = arccos(\bar{\xi}^*_{td} cos(\bar{\psi}^*_{td})) \tag{3.11}$$

and the virtual toe position can be easily found as [3]:

$$\begin{bmatrix} \bar{y}_v \\ \bar{z}_v \end{bmatrix} = \begin{bmatrix} \bar{z}_b + \bar{\xi}^*_{td} cos(\bar{\psi}^*_{td}) \\ 0 \end{bmatrix} \tag{3.12}$$

For simplicity, from now on let us address both virtual leg length and virtual leg angle as our virtual leg variables. Hence, during flight we can obtain desired virtual leg variables by bringing the physical leg into the angle calculated by equation (3.11) with a simple PD controller for the hip torque.

During stance phase, in order to control the hip torque, first we need to find the dynamic equations with respect to virtual leg variables. Using the same trigonometric facts above and in terms of dimensionless virtual leg variables, dynamical equations of SLIP-T model given in equations (3.10) become:

$$\begin{aligned} \ddot{\bar{\xi}} &= \bar{\xi}\dot{\bar{\psi}}^2 - cos(\bar{\psi}) + K_{\bar{\xi}} \\ \ddot{\bar{\psi}} &= \frac{-2\dot{\bar{\xi}} + sin(\bar{\psi})}{\bar{\xi}} + \frac{K_{\bar{\psi}}}{\bar{\xi}^2} \\ \ddot{\bar{y}}_t &= 0 \\ \ddot{\bar{z}}_t &= 0 \end{aligned} \tag{3.13}$$

where K is the forcing vector which captures the effect of spring force and hip torque [3]:

$$K = [K_{\bar{\xi}}, K_{\bar{\psi}}]^T = (D_c\bar{\theta})\bar{\tau} + (D_c\bar{\theta})\bar{F}_{spring} \tag{3.14}$$

where $D_c\bar{\theta}$ and $D_c\bar{\theta}$ are Jacobian matrices of leg angle and physical leg length with respect to virtual leg variables [3]:

$$D_c\bar{\theta} = \begin{bmatrix} \frac{\partial\bar{\theta}}{\partial\bar{\xi}} \\ \frac{\partial\bar{\theta}}{\partial\bar{\psi}} \end{bmatrix} = \begin{bmatrix} \frac{sin(\bar{\psi}-\bar{\theta})}{\bar{\rho}} \\ \frac{\bar{\xi}}{\bar{\rho}} cos(\bar{\psi} - \bar{\theta}) \end{bmatrix} \tag{3.15}$$

$$D_c\bar{\rho} = \begin{bmatrix} \frac{\partial\bar{\rho}}{\partial\bar{\xi}} \\ \frac{\partial\bar{\rho}}{\partial\bar{\psi}} \end{bmatrix} = \begin{bmatrix} cos(\bar{\psi} - \bar{\theta}) \\ \bar{\xi} sin(\bar{\psi} - \bar{\theta}) \end{bmatrix} \tag{3.16}$$

In order to make dynamical equations (3.13) match with basic SLIP dynamics as close as possible, Ankaralı suggests to choose K as [3]:

$$K = [K_{\bar{\xi}}, K_{\bar{\psi}}]^T = [DU(\bar{\xi}), 0]^T \tag{3.17}$$

where $U(\bar{\xi})$ is radial potential and $DU(\bar{\xi})$ is corresponding radial force. Since the model has only one controllable actuator, $K_{\bar{\xi}}$ and $K_{\bar{\psi}}$ cannot be regulated together. Hence, they choose to preserve the angular momentum around virtual toe and ignore the radial potential component. Using $K_{\bar{\psi}} = 0$ in the equation (3.13) we find the hip torque as [3]:

$$\bar{\tau} = -\bar{\rho} tan(\bar{\psi} - \bar{\theta}) \bar{F}_{spring} \tag{3.18}$$

3.2.2 General Control Objective of SLIP-T

Having determined the control dynamics of the SLIP-T model, in this section we discuss our control objective by answering "how" and "which variables" should we control for a stable locomotion.

Generally, apex states such as horizontal velocity ($\dot{\bar{y}}_a$) and vertical height ($\bar{z}_a$) of the body are adjusted as a control objective for SLIP and SLIP based system models. Hence, for the SLIP-T let us define the set of apex states as [3]:

$$\chi_a = \{X_a | X_a = [\dot{\bar{y}}_a, \bar{z}_a]^T\} \tag{3.19}$$

and successive apex states regulated as [3]:

$$X_a[n+1] = f_a(X_a[n], U[n]) \tag{3.20}$$

where $f_a(.,.)$ is the apex return map and $U[n]_{i=1,...,n}$ are the control inputs. For SLIP-T model, we give definitions for the apex return map and control input parameters in the following section, but first let us introduce them and state our control aim to converge desired apex states asymptotically by regulating control inputs.

3.2.2.1 Apex Return Maps

Basically, apex return maps calculate the successive apex states as in equation (3.20) given initial apex states. Due to the hybrid characteristic of our model, in order to get from one apex to another, we should know all "apex-to-touchdown" (f_{a2t}), "touchdown-to-liftoff" (f_{t2l}) and "liftoff-to-apex" (f_{l2a}) sub-return maps which together generate our apex return map, f_a such as:

$$f_a = (f_{a2t})o(f_{t2l})o(f_{l2a}) \tag{3.21}$$

To have the optimal return map, we need an analytical expression which can be obtained by solving both flight and stance dynamics. The analytical solutions for our system during flight, which include solutions for (f_{a2t}) and (f_{l2a}), are trivial; because our system just follows a ballistic trajectory starting from some initial conditions [7,34]. On the other hand, the stance dynamics that we have obtained previously are not integrable as the SLIP model [20]. Hence, there is no exact analytical solution for our system during stance phase and consequently for the (f_{t2l}).

To overcome this problem in SLIP dynamics, several controllers based on numerical solutions for the f_a were presented by some researchers [10]. However, these solutions are not accurate enough and have burden on their computation. Other than numerical approach, some researchers came up with approximate analytical solutions [6, 18, 34, 36], which were proved as more efficient than the numerical methods for the stance mapping. Hence, Ankaralı uses one of these approximate return maps in their gait controller which we give details in the following section.

3.2.2.2 Control Input Variables

Commonly in the literature, the main locomotion control input for the SLIP and related models is composed by two variables which are:

1. $\bar{\psi}$: Touchdown angle of the leg
2. ΔE: Net change in the total mechanical energy of the system

Although $\bar{\psi}$ is obviously available to us, ΔE cannot be obtained from the model directly and should be calculated using the variables within our system. In the related literature, one can find different control variables which were used to calculate ΔE [34, 36, 46]. To obtain the gait level control, Ankaralı uses leg lengths at touchdown

$\bar{\xi}_{td}$ and lift-off $\bar{\xi}_{lo}$ in order to calculate the net change in total mechanical energy during stance. As a result, our control input defined as [3]:

$$\mathcal{U} = \{U | U = [\bar{\psi}_{td}, \bar{\xi}_{td}, \bar{\xi}_{lo}]^T\} \tag{3.22}$$

3.2.3 Gait Level Control of SLIP-T

To perform the gait level controller for the SLIP-T model, Ankaralı adjusts the deadbeat control strategy for the SLIP [3], in which Geyer's analytical approximate stance map [18] shown in equation (3.23) is used to form the apex return map.

$$\begin{aligned} \bar{\xi}(\bar{t}) &= 1 + a + bsin(\hat{\omega}_0 \bar{t}) \\ \bar{\psi}(\bar{t}) &= \bar{\psi}_t d + p_{\bar{\psi}}(1-2a)(\bar{t} - \bar{t}_d) + \frac{2bp_{\bar{\psi}}}{\hat{\omega}}[cos(\hat{\omega}_0 \bar{t}) - cos(\hat{\omega}_0 \bar{t}_{td})] \end{aligned} \tag{3.23}$$

where $\hat{\omega}_0$, a and b defined as [3]:

$$\begin{aligned} \hat{\omega}_0 &:= \sqrt{\kappa_s + 3p_{\bar{\psi}}{}^2} \\ a &:= \frac{p_{\bar{\psi}}{}^2 - 1}{\hat{\omega}_0^2} \\ b &:= \sqrt{a^2 + \frac{2E - p_{\bar{\psi}}{}^2 - 2}{\hat{\omega}_0^2}} \end{aligned} \tag{3.24}$$

Here E and $p_{\bar{\psi}}$ are the total mechanical energy and angular momentum respectively. Furthermore, in the derivation of his stance map Geyer assumes that touchdown and liftoff lengths are equal to the rest length of the spring. Using this assumption and the analytical expressions for the virtual leg variables, critical time events (touchdown, bottom and lift-off) are given as [3]:

$$\begin{aligned} \bar{t}_{td} &= \frac{\pi - arcsin(\frac{\bar{\xi}_{td} - 1 - a}{b})}{\hat{\omega}_0} \\ \bar{t}_{lo} &= \frac{2\pi + arcsin(\frac{\bar{\xi}_{lo} - 1 - a}{b})}{\hat{\omega}_0} \\ \bar{t}_b &= \frac{3\pi}{\hat{\omega}_0} \end{aligned} \tag{3.25}$$

In the deadbeat control, in order to achieve the desired apex states [3]:

$$X_a^* = f_a(X_a, U^*) \tag{3.26}$$

Table 3.7: Computation of leg length control variables $\bar{\xi}_{td}$ and $\bar{\xi}_{lo}$ [3]

	$\Delta E > 0$	$\Delta E < 0$
$\bar{\xi}_{td}$	$1 - \sqrt{\frac{2\Delta E}{\kappa_s}}$	1
$\bar{\xi}_{lo}$	1	$1 - \sqrt{\frac{2\Delta E}{\kappa_s}}$

We need to determine the control input variables as [3]:

$$U^* = [\bar{\psi}^*_{td}, \bar{\xi}^*_{td}, \bar{\xi}^*_{lo}]^T\} \tag{3.27}$$

In order to find leg length at touchdown, $\bar{\xi}^*_{td}$ and lift-off, $\bar{\xi}^*_{lo}$ we use the single stride mechanical energy difference between current, X_a and the desired, X^*_a apex states which is calculated as [3]:

$$\Delta E_{\bar{\tau}} = (\bar{z_a}^* - \bar{z_a}) + \frac{1}{2}(\dot{\bar{y}}^*_a)^2 - (\dot{\bar{z}}_a)^2 + \Delta E_{loss} \tag{3.28}$$

Here ΔE_{loss} is the energy loss due to the damping and accurate estimation of this loss is a hard problem. Hence, Ankaralı uses sinusoidal fit to calculate $\Delta E_{\bar{\tau}}$ inspired from their measured data, touchdown and lift-off compression rates and stance duration.

The sign of the energy difference determine whether we inject energy into the system by compressing the spring before the stance phase, or take the energy out from the system. Based on the sign, Table 3.7 gives touchdown and lift-off leg length commands for the SLIP model.

According to the Table 3.7, when $\Delta E < 0$, touchdown leg length should be kept equal to the rest length and lift-off length of the leg should be shorten. Since there is no such mechanism in SLIP-T model, we cannot adjust the lift-off leg length as in the table. Fortunately, because of the damping factor on the spring model, the sign of ΔE rarely becomes negative. As a result, Ankaralı assumes $\bar{\xi}_{td} = \bar{\xi}_{lo} = 1$ which means that $\bar{\xi}_{lo}$ is no longer control input variable.

On the other hand, when $\Delta E > 0$, the touchdown leg length should be shorten like compressing the spring and the lift-off length is kept equal to rest length. However, due to the difference on torque controlled dynamics of our model and fully passive stance dynamics of SLIP model, the formulation for the touchdown leg length is not accurate.

Hence, Ankaralı makes a deeper analysis and calculate the energy supplied by the hip torque as [3]:

$$\Delta E_{\bar{\tau}} = \int_{\bar{t}_{td}}^{\bar{t}_{lo}} \bar{\tau}\dot{\bar{\theta}}(\bar{t})d\bar{t} = \int_{\bar{t}_{td}}^{\bar{t}_{lo}} -\bar{\rho}(\bar{t})tan(\bar{\psi}(\bar{t}) - \bar{\theta}(\bar{t}))\bar{F}_{spring}(\bar{t})\dot{\bar{\theta}}(\bar{t})d\bar{t} \tag{3.29}$$

Having already compensated for damping, they assume that [3]:

$$\bar{F}_{spring}(\bar{t}) = -\kappa(\bar{\rho}(\bar{t}) - 1) \tag{3.30}$$

Hence the energy difference formula become [3]:

$$\Delta E_{\bar{\tau}} \approx \int_{\bar{t}_{td}}^{\bar{t}_{lo}} \bar{\rho}(\bar{t})tan(\bar{\psi}(t) - \bar{\theta}(\bar{t}))\kappa(\bar{\rho}(\bar{t}) - 1)\dot{\bar{\theta}}(\bar{t})d\bar{t} \tag{3.31}$$

Still it does not provide an analytical solutions and they further assume that $(1 - \bar{\rho}) \approx (1 - \bar{\xi})$), which is plausible if there is not much difference between the actual and desired gait parameters. Moreover, they approximate the virtual leg angle to its bottom value since the angle difference between physical and virtual leg remains relatively constant during stance phase. Thus, equation (3.31) becomes [3]:

$$\Delta E_{\bar{\tau}} \approx \int_{\bar{t}_{td}}^{\bar{t}_{lo}} \kappa(\bar{\xi}(\bar{t}) - 1)tan(\bar{\psi}_b - \bar{\theta}_b)\bar{\rho}\dot{\bar{\theta}}(\bar{t})d\bar{t} \tag{3.32}$$

Inserting the radial solution in equation (3.23) [3]:

$$\Delta E_{\bar{\tau}} \approx \kappa tan(\bar{\psi}_b - \bar{\theta}_b)\bar{\rho}_b\dot{\bar{\theta}}_b(a(\bar{t}_{td} - \bar{t}_{lo}) - \frac{b(cos(\hat{\omega}_0\bar{t}_{td}))}{\hat{\omega}_0}) \tag{3.33}$$

where $\hat{\omega}_0$, a, b and time events defined in equations (3.24) and (3.25) [3].

They want to avoid from solving this equation numerically, so that they modify equation (3.33) in order to use the neutral touchdown angle. They do this modification using their observation that angular dynamics do not substantially effect the radial, energetic behavior of the system.

$$\bar{\psi}_n := \{\bar{\psi}_{td} \mid [\dot{\bar{y}}_a, \bar{z}_a]^T = \hat{f}_a(\bar{\psi}_t, [\dot{\bar{y}}_a, \bar{z}_a])\} \tag{3.34}$$

Using the neutral touchdown angle that defined in equation (3.34) [3] and taking it as an input to the controller, they solve for $\bar{\xi}_{td}$ to achieve the desired pumping energy.

Up to now, we have determined $\bar{\xi}_{lo} = 1$ and calculate $\bar{\xi}_t d$ using the desired energy that should be given into the system in order to reach the successive apex state. Now we can find the remaining control input $\bar{\psi}_t d$ using the approximate apex return map f_a which is formed by Geyer's approximate stance map together with apex-to-touchdown and liftoff-to-touchdown maps. In order to reduce the control problem into one dimensional equation, Ankaralı defines an operator that retrieves the desired forward velocity from the apex return map such that [3]:

$$\dot{y}^* = (\pi_{\bar{y}^*} \ o \ \hat{f}_a)(\bar{\psi}_{td}) \tag{3.35}$$

However, neither one of the approximate return maps is not invertible in closed form. Hence we solve this problem numerically by defining a minimization [3]:

$$\bar{\psi}_t = \underset{\frac{-\pi}{2}<\bar{\psi}<\frac{\pi}{2}}{\mathrm{argmin}} \ (\dot{\bar{\mathrm{y}}}^* \text{-} (\pi_{\bar{\mathrm{y}}^*} \ \mathrm{o} \ \hat{\mathrm{f}}_{\mathrm{a}}))^2 \tag{3.36}$$

since the equation has simple one dimensional form and ψ_{td} has a monotonic behavior.

CHAPTER 4

PLANAR ONE-LEGGED HOPPING ROBOT WITH REACTION WHEEL

4.1 Motivation and Introduction

In the previous chapter we have introduced the SLIP-T model and reviewed its gait controller to generate stable locomotion. However in the model an infinite body inertia is assumed such that the applied hip torque does not affect the attitude of the robot body. In practice, applying time varying leg torque profiles would result in a net change in the body attitude at the end of each stride. To compensate for this angular disturbance, a compensating torque is required to be applied directly to the robot body. In this chapter, we discuss the feasibility of a reaction wheel based approach to this problem by placing the reaction wheel at the center-of-mass (CoM) of the robot body. Hence, we can use the gait level controller for the locomotion of our model without any modification. For each hybrid phase of the system (flight and stance) during the locomotion of the monopod, the dynamical equations of the planar SLIP-T model with reaction wheel is derived using Newton-Euler formulation. Furthermore, we implement a physics based hybrid simulation of this model. We present results of these simulations in order to discuss the feasibility of the reaction wheel approach to compensate for the angular disturbances in body attitude.

4.2 System Model of the Planar Monopod Hopper Robot with Reaction Wheel

Our system model is structurally very similar to the SLIP-T model which is introduced in the previous chapter. In our model, we take away the assumption of stationary body attitude by adding a rotational degree of freedom. In order to compensate disturbances on the body attitude, the body consists of a reaction wheel which is placed at CoM of the body (Figure 4.1). To simplify the dynamical equations, leg of the robot is assumed as massless; but in order to make the hip torque control more realistically, leg is modeled as a single dissipative spring with a toe mass m_t which is relatively very small compared to the mass of the robot body m_b. $(m_t << m_b)$

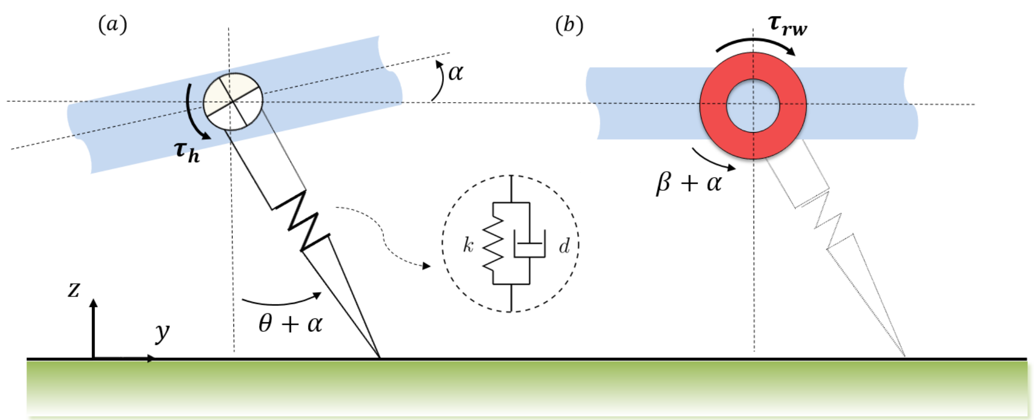

Figure 4.1: Planar monopod hopper robot model which has (a) angular disturbance α due to the hip torque τ^{hip} and (b) the reaction wheel (red disc) based approach to stabilize the robot body using a compensation torque τ^{rw}

In the model, there are twelve state variables. Eight of them are same as SLIP-T model which are defined as positions of body y_b, z_b and toe y_t, z_t in ground coordinate frame and their first order derivatives with respect to time: $\dot{y}_b$, $\dot{z}_b$, $\dot{y}_t$, $\dot{z}_t$. The remaining four variables are reaction wheel angle β, the angle of the body α, and first time order derivatives $\dot{\beta}$, $\dot{\alpha}$, respectively. Mid-state variables such as leg length ρ and leg angle θ are calculated based on states and system parameters, which determine the physical properties of the system. Also the ground reaction forces between reaction wheel, body, leg and ground in both horizontal and vertical directions is shown by $F_y^{rw,body}$, $F_z^{rw,body}$, $F_y^{leg,body}$, $F_z^{leg,body}$, F_y^{ground}, F_z^{ground}. The explicit definitions of state, mid-state variables and parameters of the system model can be referred in Table 4.1, 4.2 and 4.3 respectively.

Table 4.1: State Variables of the monopod hopper with reaction wheel model

Symbol	Definition
y_b	Horizontal body position with respect to the ground coordinate system
z_b	Vertical body position with respect to the ground coordinate system
y_t	Horizontal toe position with respect to the ground coordinate system
z_t	Vertical toe position with respect to the ground coordinate system
β	Angle of the reaction wheel with respect to the body coordinate system in counter clockwise direction
α	Angle of the body with respect to the ground coordinate system in counter clockwise direction
$\dot{y}_b$	Horizontal body velocity with respect to the ground coordinate system
$\dot{z}_b$	Vertical body velocity with respect to the ground coordinate system
$\dot{y}_t$	Horizontal toe velocity with respect to the ground coordinate system
$\dot{z}_t$	Vertical toe velocity with respect to the ground coordinate system
$\dot{\beta}$	Angular velocity of the reaction wheel with respect to the body coordinate system in counter clockwise direction
$\dot{\alpha}$	Angular velocity of the body with respect to the ground coordinate system in counter clockwise direction

Table 4.2: Mid-State Variables of the monopod hopper with reaction wheel model

Symbol	Definition
ρ	length of the leg
θ	Angle of the leg with respect to the body coordinate system in counter clockwise direction
$F_y^{rw,body}$	Reaction force in horizontal direction between reaction wheel and body
$F_z^{rw,body}$	Reaction force in vertical direction between reaction wheel and body
$F_y^{leg,body}$	Reaction force in horizontal direction between leg and body
$F_z^{leg,body}$	Reaction force in vertical direction between leg and body
F_y^{ground}	Ground reaction force in horizontal direction
F_z^{ground}	Ground reaction force in vertical direction
F^{spring}	Radial force on the spring by compression and decompression
τ^{rw}	Torque value that is applied on the reaction wheel
τ^{hip}	Torque value that is applied on the hip of the leg

Table 4.3: System Parameters of th monopod hopper with reaction wheel model

Symbol	Definition
g	Gravity
k	Spring stiffness coefficient
d	Spring damping coefficient
m_{rw}	Mass of the reaction wheel
m_b	Mass of the body
m_t	Point mass of the toe
I_{rw}	Moment of inertia of the reaction wheel
I_b	Moment of inertia of the body without reaction wheel

4.3 Dynamic Equations of the Planar Monopod Hopper Robot with Reaction Wheel

Since we require the reaction forces between reaction wheel, body and leg, especially to detect transition events, we continue our approach from the previous chapter and derive the two different set of dynamic equations with respect to the successive phases (flight and stance) of the planar system using Newton-Euler methodology.

In order to derive the dynamical equations using Newton-Euler force analysis, we divide the model into three parts: reaction wheel, body and leg, and analyze all internal-external and inertial forces between them. In free body diagrams, here there are only two main differences apart from the SLIP-T model. First one is obviously the fbd (Figure 4.2) of the reaction wheel which is same for both phases, and respective equations are shown in (4.1).

$$\begin{aligned} m_{rw}\ddot{y}_b + F_y^{rw,body} &= 0 \\ m_{rw}\ddot{z}_b + F_z^{rw,body} &= -m_{rw}g \\ I_{rw}(\ddot{\beta} + \ddot{\alpha}) &= \tau^{rw} \end{aligned} \tag{4.1}$$

Secondly there can be an angular change α in body due to the τ^{hip}, which is compensated by the τ^{rw}. Hence, we should include them into the respective fbd diagram (Figure 4.3) that we neglect in SLIP-T model. Thus, equations for the body modified as in (4.2).

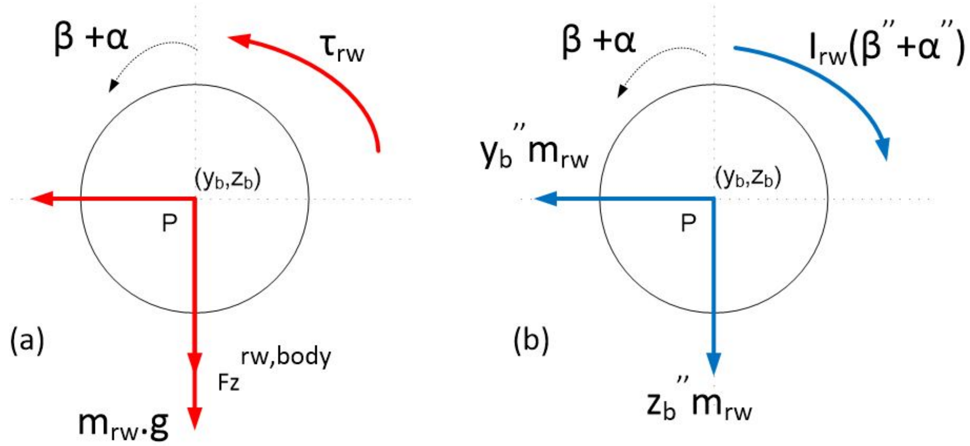

Figure 4.2: Free-body diagrams of (a) internal-external, and (b) inertial forces of the reaction wheel in both flight and stance phases

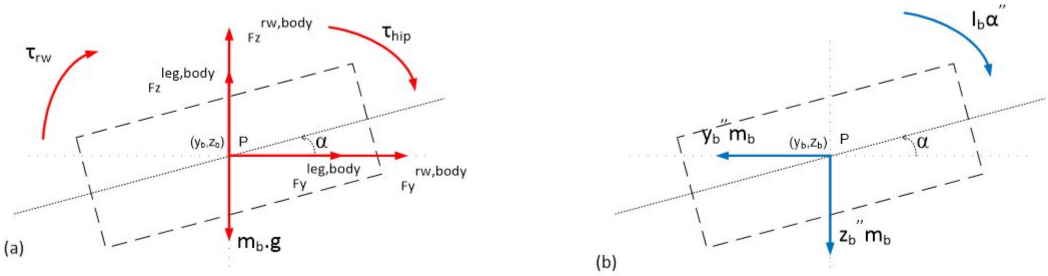

Figure 4.3: Free-body diagrams of (a) internal-external, and (b) inertial forces of the planar monopod hopper robot body in both flight and stance phases

$$
\begin{aligned}
m_b\ddot{y}_b - F_y^{rw,body} - F_y^{leg,body} &= 0 \\
m_b\ddot{z}_b - F_z^{rw,body} - F_z^{leg,body} &= -m_b g \\
I_b\ddot{\alpha} &= -\tau^{rw} - \tau^{hip}
\end{aligned}
\tag{4.2}
$$

The fbd (Figures 4.4, 4.5) and equations (4.3, 4.4) for the leg in both phases remains same as we derived in Chapter 3.

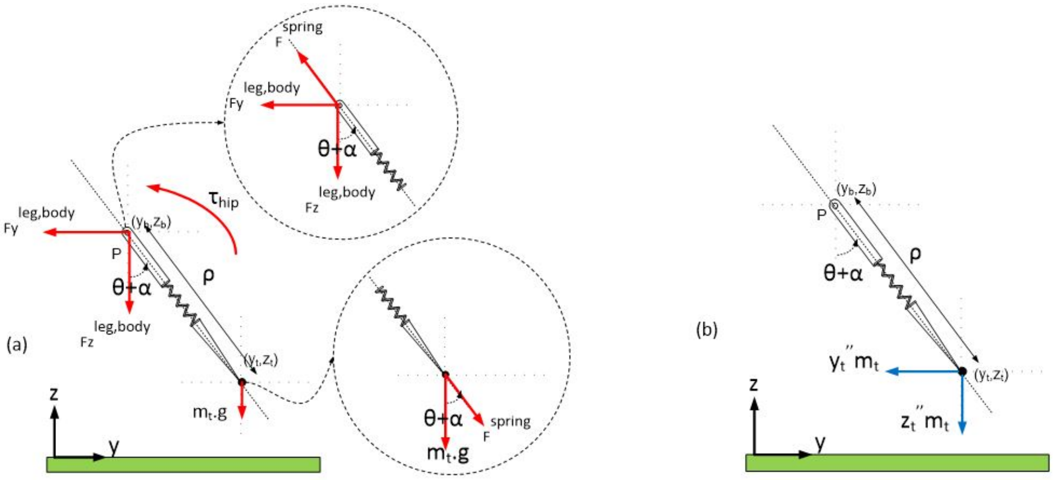

Figure 4.4: Free-body diagrams of (a) internal and external forces and (b) inertial forces of the leg in flight phase

$$
\begin{aligned}
m_t\ddot{y}_t + F_y^{leg,body} &= 0 \\
m_t\ddot{z}_t + F_z^{leg,body} &= -m_t g \\
F_z^{leg,body}cos(\theta+\alpha) - F_y^{leg,body}sin(\theta+\alpha) &= F^{spring} \\
F^{spring} &= -k(\rho-\rho_0) - d\dot{\rho} \\
m_t\rho cos(\theta+\alpha)\ddot{y}_t + m_t\rho sin(\theta+\alpha)\ddot{z}_t &= \tau^{hip} - m_t g\rho sin(\theta+\alpha)
\end{aligned}
\tag{4.3}
$$

$$
\begin{aligned}
F_y^{ground} - F_y^{leg,body} &= 0 \\
F_z^{ground} - F_z^{leg,body} &= m_t g \\
F_y^{leg,body}\rho cos(\theta+\alpha) + F_z^{leg,body}\rho sin(\theta+\alpha) &= -\tau^{hip} \\
F_z^{leg,body}cos(\theta+\alpha) - F_y^{leg,body}sin(\theta+\alpha) &= F^{spring} \\
F^{spring} &= -k(\rho-\rho_0) - d\dot{\rho}
\end{aligned}
\tag{4.4}
$$

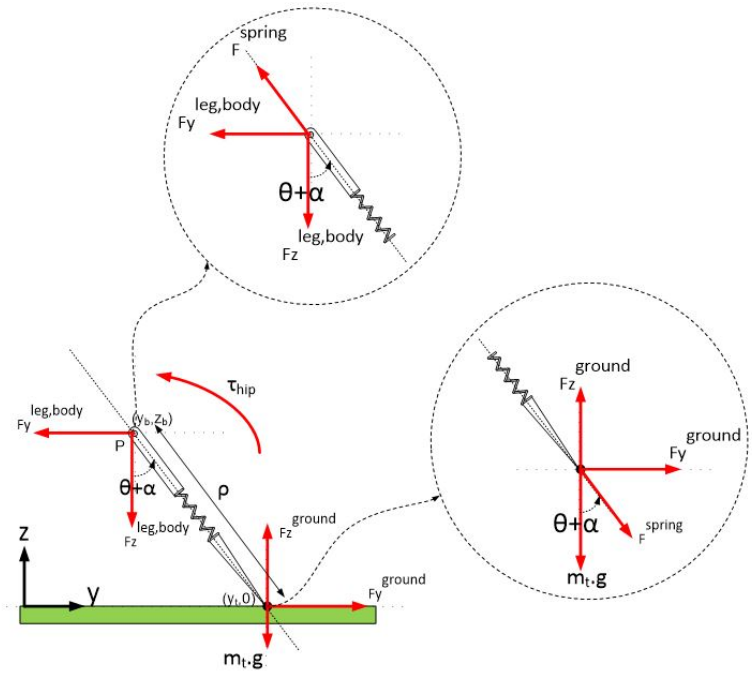

Figure 4.5: Free-body diagram of the internal forces of the leg in stance phase. There is no external force since the leg is massless.

For both phases (flight and stance), we convert the dynamic equations (from 4.1 to 4.4) into the state matrix form [5]. We invert the mass-inertia matrix (M_{flight}, M_{stance}) (4.6, 4.9) and then multiply it with the known forces vector (f_{flight}, f_{stance}) (4.8, 4.11). Hence, the second derivatives of the states and the unknown mid-state variables (x_{flight}, x_{stance}) (4.7, 4.10) are obtained as in equation (4.5). At the end, we have the vector field of the system for both phases. Solving these vector fields using an ordinary differential equations solver, we obtain the successive state values.

$$\begin{aligned} M.x &= f \\ x &= M^{-1}.f \end{aligned} \tag{4.5}$$

$$M_{flight} = \begin{pmatrix} m_{rw} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & m_{rw} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{rw} & I_{rw} & 0 & 0 & 0 & 0 \\ m_b & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & m_b & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & I_b & 0 & 0 & 0 & 0 \\ 0 & 0 & m_t & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & m_t & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -sin(\theta+\alpha) & cos(\theta+\alpha) \\ 0 & 0 & m_t\rho cos(\theta+\alpha) & m_t\rho cos(\theta+\alpha) & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.6}$$

$$x_{flight} = \begin{pmatrix} \ddot{y}_b & \ddot{z}_b & \ddot{y}_t & \ddot{z}_t & \ddot{\beta} & \ddot{\alpha} & F_y^{rw,body} & F_z^{rw,body} & F_y^{leg,body} & F_z^{leg,body} \end{pmatrix} \tag{4.7}$$

$$f_{flight} = \begin{pmatrix} 0 & -m_{rw}g & \tau_{rw} & 0 & -m_b g & -\tau_{rw}-\tau_{hip} & 0 & -m_t g & -k(\rho-\rho_0)-d\dot{\rho} & \tau_{hip}-m_t g\rho sin(\theta+\alpha) \end{pmatrix} \tag{4.8}$$

$$M_{stance} = \begin{pmatrix} m_{rw} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{rw} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_{rw} & I_{rw} & 0 & 0 & 0 & 0 & 0 & 0 \\ m_b & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & m_b & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & I_b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \rho cos(\theta+\alpha) & \rho sin(\theta+\alpha) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -sin(\theta+\alpha) & cos(\theta+\alpha) & 0 & 0 \end{pmatrix} \tag{4.9}$$

$$x_{stance} = \begin{pmatrix} \ddot{y}_b & \ddot{z}_b & \ddot{\beta} & \ddot{\alpha} & F_y^{rw,body} & F_z^{rw,body} & F_y^{leg,body} & F_z^{leg,body} & F_y^{ground} & F_z^{ground} \end{pmatrix} \tag{4.10}$$

$$f_{stance} = \begin{pmatrix} 0 & -m_{rw}g & \tau_{rw} & 0 & -m_b g & -\tau_{rw}-\tau_{hip} & 0 & m_t g & -\tau_{hip} & -k(\rho-\rho_0)-d\dot{\rho} \end{pmatrix} \tag{4.11}$$

4.4 Physics Based Hybrid System Simulation

Hybrid system simulation (with ground contact) of the planar monopod robot which has a reaction wheel located in CoM is performed by using MATLAB environment

[27]. The overall simulation flowchart is given in Figure 4.6. According to the chart, first we specify initial values of state variables and system parameters, which determine the phase of the system. Then controllers (hip and reaction wheel) make the states converge to desired values. The hybrid dynamical equations are solved at each iteration by an ordinary differential equation solver which we use "ode45" function in our simulations. The transition between phases triggered by states or mid-state variables using the event function for the ode45 solver. Specifically, we define the transition from flight to stance by detecting the time when vertical toe position decay to zero. The transition from stance to flight is determined when the vertical ground force (sensed by the leg) becomes zero. The flow continues until a criteria based on the states reached or a user defined time duration passed.

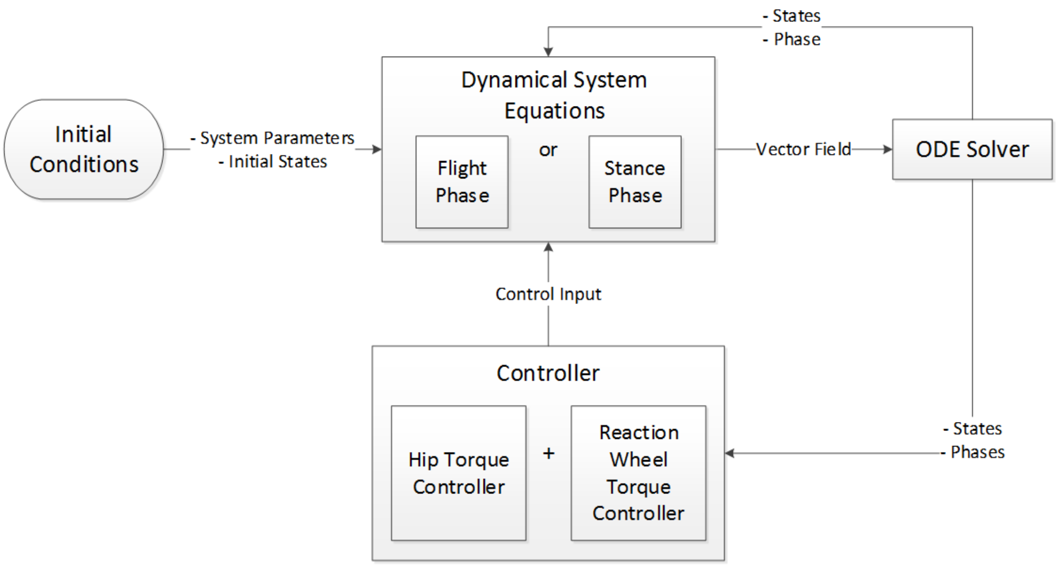

Figure 4.6: General flowchart of the physics based hybrid system simulation

Two hip torque controllers are used in simulations. These are:

1) Controller that keeps the robot in standing position

2) Controller for one-direction stable locomotion

For the reaction wheel controller, in order to compensate the disturbed body angle we use a simple PD controller in form of:

$$\tau^{rw} = -k_{prw}\alpha - k_{drw}\dot{\alpha}. \tag{4.12}$$

4.4.1 Using the hip torque controller that keeps the robot in standing position

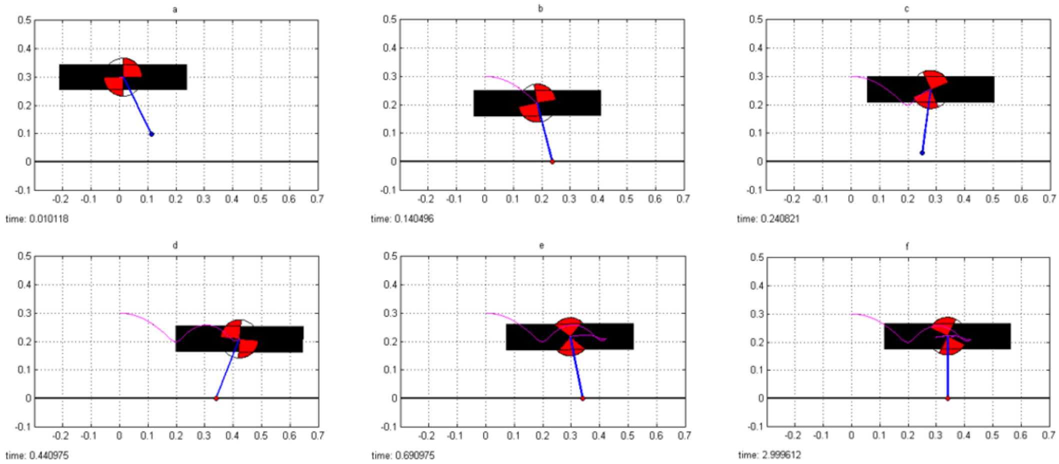

Figure 4.7: Snapshots from the animation of the planar monopod hopper with reaction wheel under the hip torque controller which tries to keep the robot in standing position

In this section, sample simulation results that are related with the reaction wheel angular velocity $\dot{\beta}$, body attitude α, generated torques for reaction wheel τ^{rw} and hip τ^{hip} are presented. Here, the hip torque which keeps the robot in standing position is controlled with a basic PD controller such as:

$$\tau^{hip} = -k_p(\theta + \alpha) - k_d(\dot{\theta} + \dot{\alpha}). \tag{4.13}$$

The proportional k_{prw}, k_p and derivative k_{drw}, k_d control parameters for both reaction wheel and hip torque is determined by empirically and further analysis are needed for the optimization of these parameters. In the sample simulation the control parameters adjusted as: $k_{prw} = 40, k_{drw} = 20, k_p = 40, k_d = 1$.

In the hybrid system simulations, our system model is tested starting from various initial conditions. Table 4.4 shows (a)initial state values and also (b)system parameters for the sample simulation, where we initially start our monopod system in flight phase with an initial horizontal velocity same for both body and leg. Simulations are performed sufficient durations until the stability of body attitude is obtained.

Starting from the initial conditions stated above, Figure 4.8 shows the leg angle values with respect to time under the hip torque controller which tries to bring and keep the leg in vertical position.

Table 4.4: Initial Conditions and System Parameters (Standing controller scenario)

(a)

Symbol	Initial Value and Unit
y_b	0 m
z_b	0.3 m
y_t	0.1 m
z_t	0.1 m
β	0 rad
α	0 rad
$\dot{y}_b$	1.3 $\frac{m}{s}$
$\dot{z}_b$	0 $\frac{m}{s}$
$\dot{y}_t$	1.3 $\frac{m}{s}$
$\dot{z}_t$	0 $\frac{m}{s}$
$\dot{\beta}$	0 $\frac{rad}{s}$
$\dot{\alpha}$	0 $\frac{rad}{s}$

(b)

Symbol	Value and Unit
g	9.81 $\frac{m}{s^2}$
k	15000 $\frac{N}{m}$
d	100 $\frac{Ns}{m}$
m_{rw}	1 kg
m_b	3 kg
m_t	0.2 kg
r_{rw}	0.067 m
b_w	0.045 m
b_l	0.22 m
I_{rw}	0.002 $kg.m^2$
I_b	0.013 $kg.m^2$

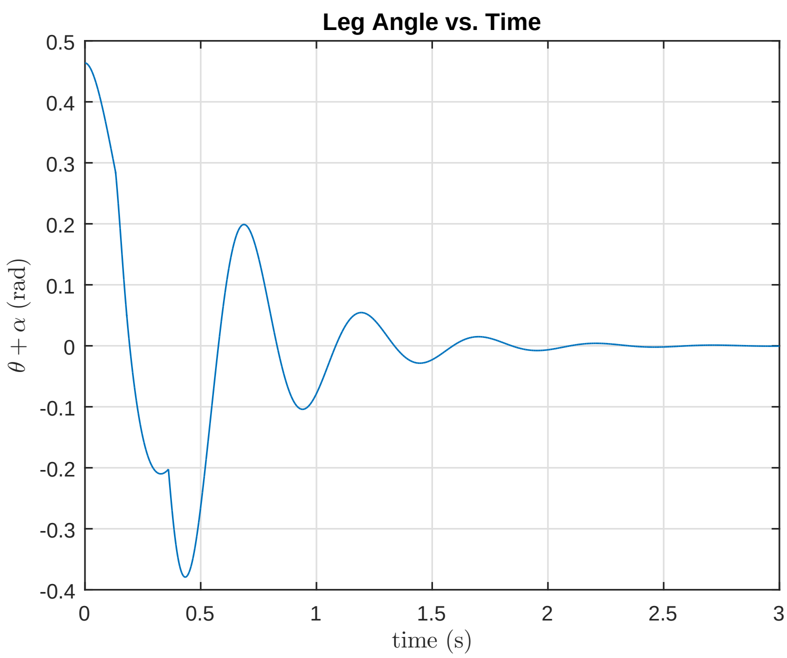

Figure 4.8: The sample simulation graph that shows leg angle values with respect to time under the hip torque controller which tries to keep the robot in standing position

In the absence of the reaction wheel control, the deviation on body angle due to the hip torque is shown in Figure 4.9. Since toe mass is much lighter than body mass, the required torque which brings the leg to the desired angle in flight phase is much less than the required toque that moves the body in stance phase. Hence, the primary angular deviation occurs during the stance phase.

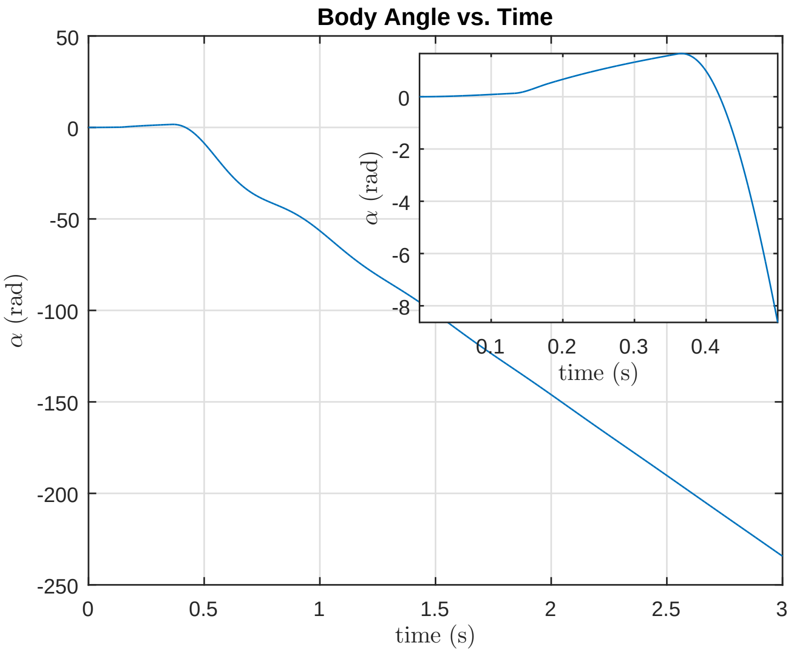

Figure 4.9: The sample simulation graph that shows body angle values with respect to time under the hip torque controller which tries to keep the robot in standing position but no reaction wheel control

In the presence of reaction wheel, the body attitude is kept at the horizontal position as shown in the top of the Figure 4.10. The bottom of the same figure shows changes in angular velocity of the reaction wheel with respect to time. It is clear that reaction wheel speed converges to a certain value which depends on the initial conditions of the system. If the required compensation torque is higher, the converged velocity of the reaction wheel would be also higher since torque and acceleration are linearly related.

The phase portrait of the system (Figure 4.11) reveals that both body angular velocity $\dot{\alpha}$ and body angle α converge to zero in steady state.

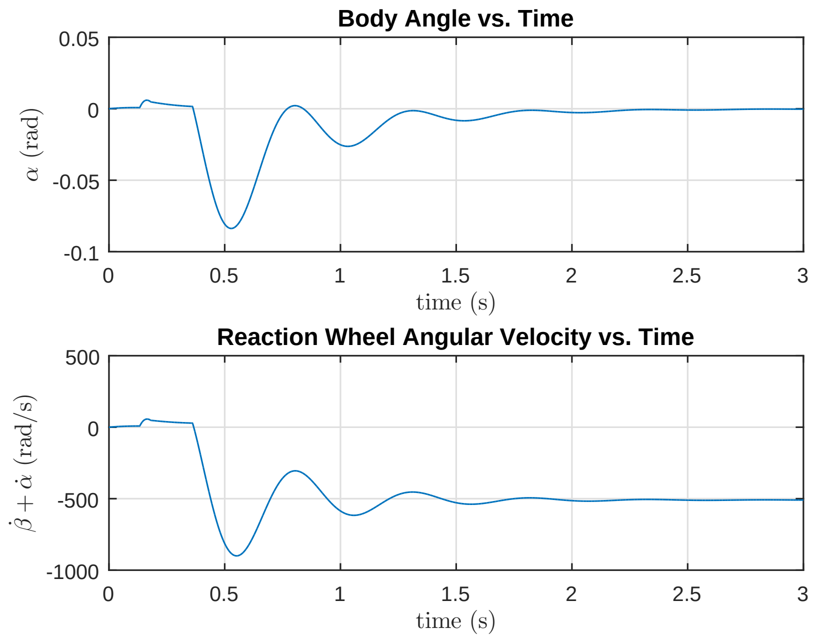

Figure 4.10: The sample simulation graph that shows (top) body angle and (bottom) angular velocity of the reaction wheel with respect to time under the hip torque controller which tries to keep the robot in standing position

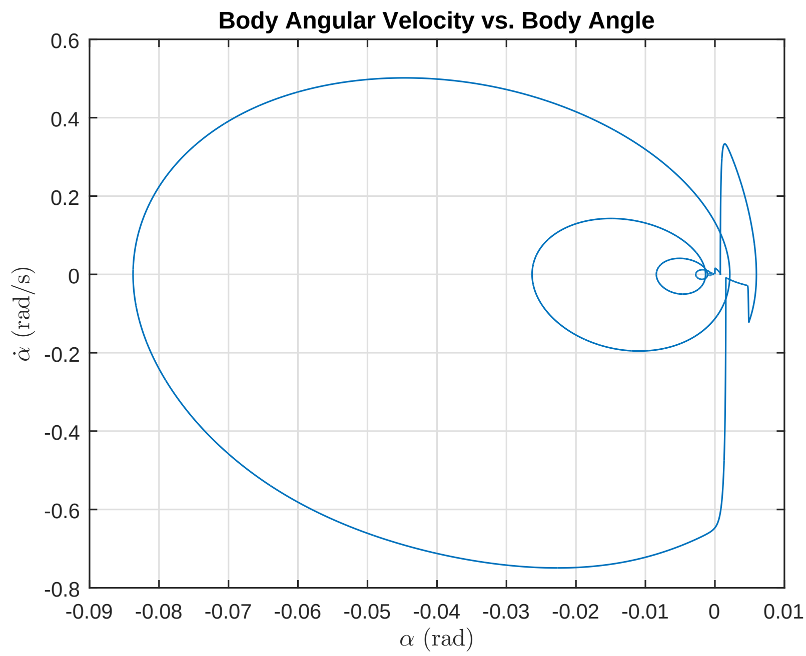

Figure 4.11: The sample simulation graph that shows body angular velocity with respect to body angle under the hip torque controller which tries to keep the robot in standing position

When we draw the reaction wheel torques with respect to its angular velocity values, we obtain a spiral form (Figure 4.12) which converges to a zero torque with a constant velocity point.

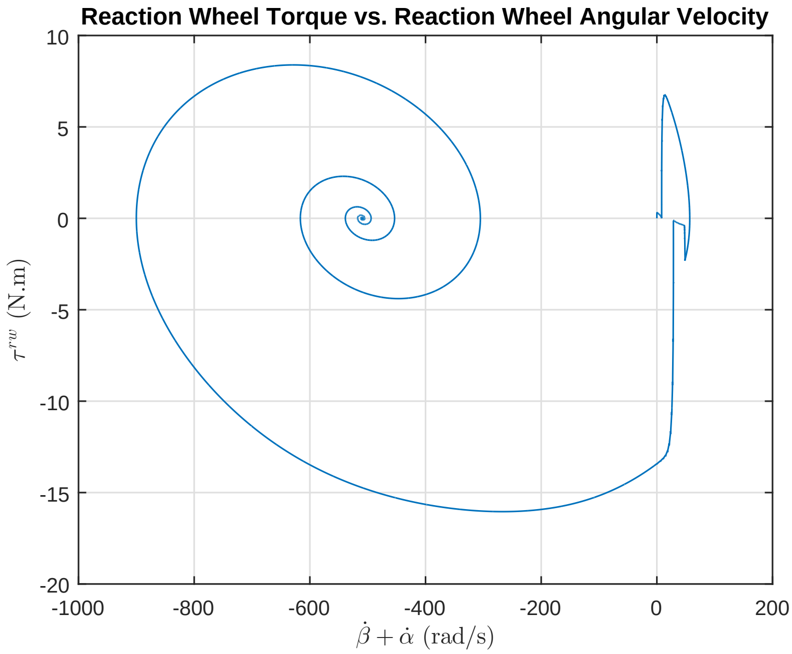

Figure 4.12: The sample simulation graph that shows torque τ_{rw} with respect to in angular velocity $\dot{\beta}$ for the reaction wheel under the hip torque controller which tries to keep the robot in standing position

4.4.2 Using the hip torque controller for one-direction stable locomotion

In order to achieve a stable locomotion of the SLIP-T model, a template based hip torque controller is presented in previous chapter and further analysis on the controller can be found in [3, 4]. Note that, although our robot model and SLIP-T are slightly different, these differences do not have an effect on the control dynamics since we assume that reaction wheel placed at the center of mass of the body and also the axis of the body angle coincide with this center.

Again, the reaction wheel is controlled with a basic PD controller and its control parameters $k_{prw} = 40, k_{drw} = 20$ are adjusted experimentally as in the previous section. Table 4.5 shows (a)initial state values and also (b)system parameters for a

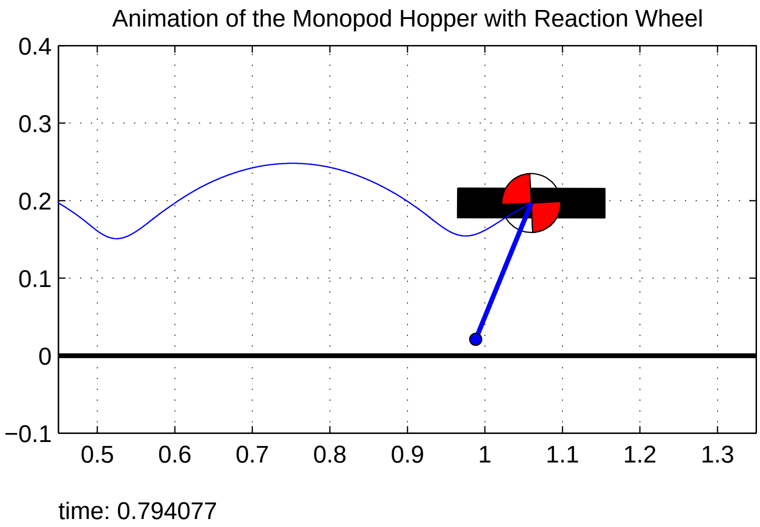

Figure 4.13: Snapshots from the animation of the planar monopod hopper with reaction wheel under the hip torque controller for one-direction stable locomotion

different sample simulation than the previous section. Here we also start our monopod simulation in flight phase with an initial velocity same for body and leg. Since the controller performs a cyclic behavior, simulation is performed 8 seconds which is sufficient enough to make clear observations. For the hip torque controller, the desired height and speed of the robot body are taken as 0.23 m and 1.5 m/s, respectively.

Using the hip torque controller for stable locomotion and the initial system parameters, leg angle variation in time is presented in Figure 4.14.

While there is no reaction wheel control, the body attitude instantly changes in one angular direction under the high hip torque profile in stance (Figure 4.15). Actually in flight phase, hip torque is also applied in the other direction to bring the leg to a desired angle. Hence, there should be a slow down on body angular velocity during the flight. However, hip torque due to the toe mass is too small to see the changes clearly in the figure.

Controlling the reaction wheel torque, the phase portrait of the system converges to a point (Figure 4.16) and the attitude of the robot is successfully kept at an almost

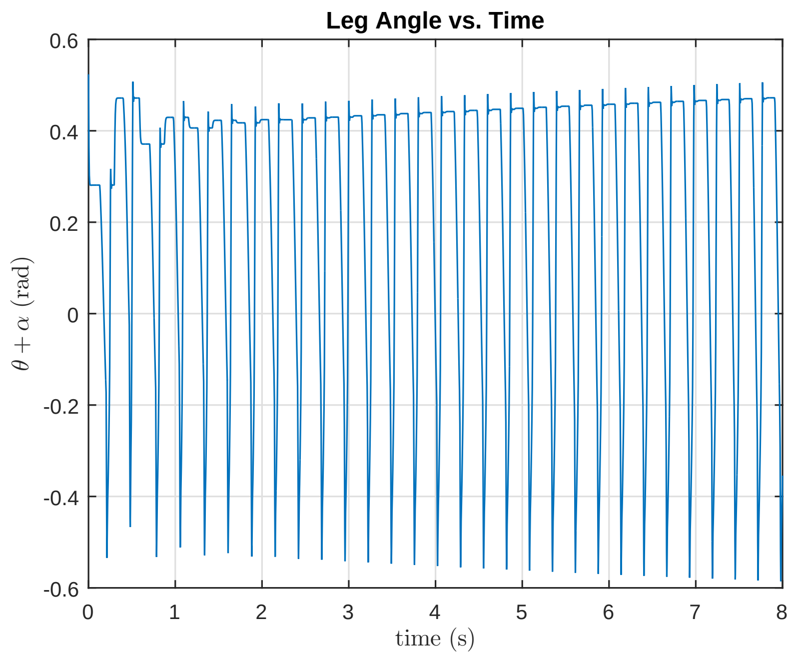

Figure 4.14: The sample simulation graph that shows leg angle values with respect to time under the hip torque controller for one-direction stable locomotion

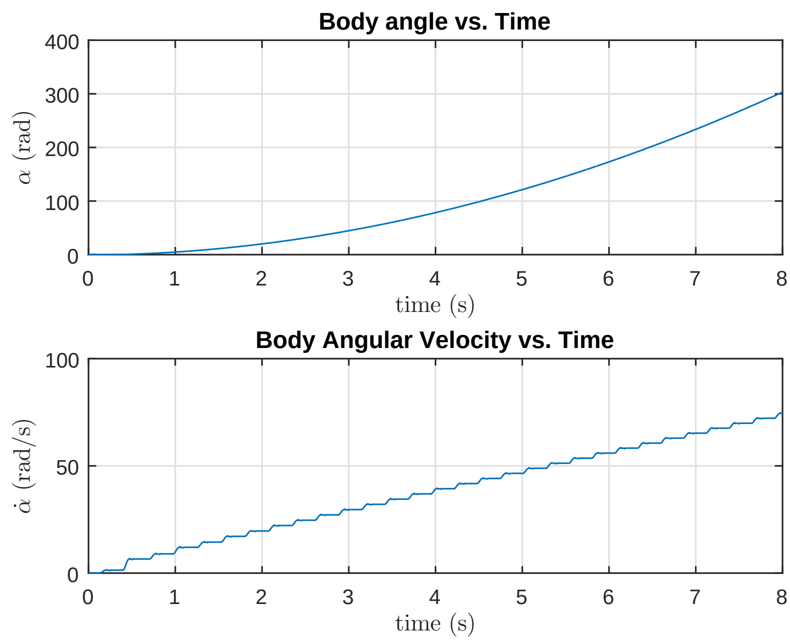

Figure 4.15: The sample simulation graph that shows body angle with respect to time under the hip torque controller for one-direction stable locomotion bu no reaction wheel control

Table 4.5: Initial Conditions and System Parameters (One-direction stable locomotion controller scenario)

(a)

Symbol	Initial Value and Unit
y_b	0 m
z_b	0.266 m
y_t	0.095 m
z_t	0.10146 m
β	0 rad
α	0 rad
$\dot{y}_b$	1.2281 $\frac{m}{s}$
$\dot{z}_b$	0.0001 $\frac{m}{s}$
$\dot{y}_t$	1.2281 $\frac{m}{s}$
$\dot{z}_t$	0.0001 $\frac{m}{s}$
$\dot{\beta}$	0 $\frac{rad}{s}$
$\dot{\alpha}$	0 $\frac{rad}{s}$

(b)

Symbol	Value and Unit
g	9.81 $\frac{m}{s^2}$
k	12000 $\frac{N}{m}$
d	72 $\frac{Ns}{m}$
m_{rw}	1 kg
m_b	6 kg
m_t	0.2 kg
r_{rw}	0.038 m
b_w	0.045 m
b_l	0.168 m
I_{rw}	0.000722 $kg.m^2$
I_b	0.23465 $kg.m^2$

constant angle (top Figure 4.17). However, the angular speed of the reaction wheel unrealistically increases to keep up with the desired input torque level (Figure 4.17).

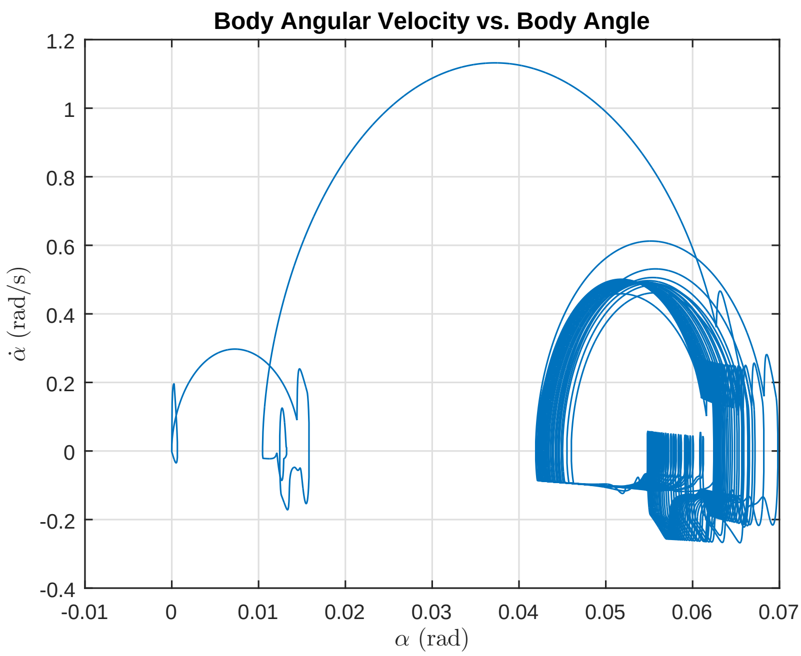

Figure 4.16: The sample simulation graph that shows body angular velocity with respect to body angle under the hip torque controller for one-direction stable locomotion

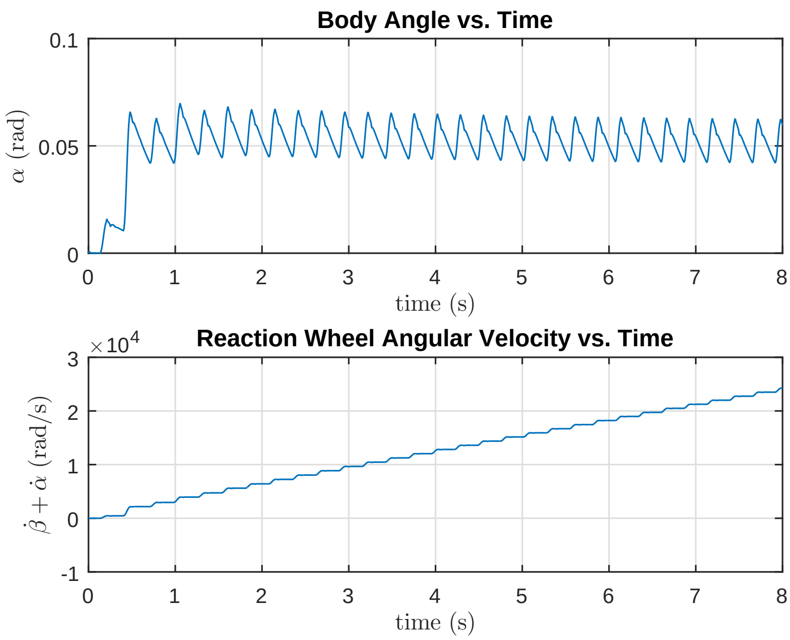

Figure 4.17: The sample simulation graph that shows (top) body angle (bottom) angular velocity of the reaction wheel (PD controlled) with respect to time under the hip torque controller for one-direction stable locomotion

Since the robot runs in one direction, the angular change due to the periodic hip torque affects on a single direction. This fact makes the reaction wheel approach infeasible for this particular situation. Since reaction wheel speed keeps increasing, the torque-velocity graph (Figure 4.18) does not converge to a point as in previous section, instead it shows a periodic characteristic.

Furthermore, we observe that our simple PD controller cannot diminish the angular offset completely. Hence, we add an integral term to obtain a PID controller such that:

$$\tau^{rw} = -k_{prw}\alpha - k_{drw}\dot{\alpha} - k_{irw}\int_0^t \alpha(t)dt. \tag{4.14}$$

Using this PID controller to determine the reaction wheel torque input, we are able to remove the angular offset (Figure 4.19).

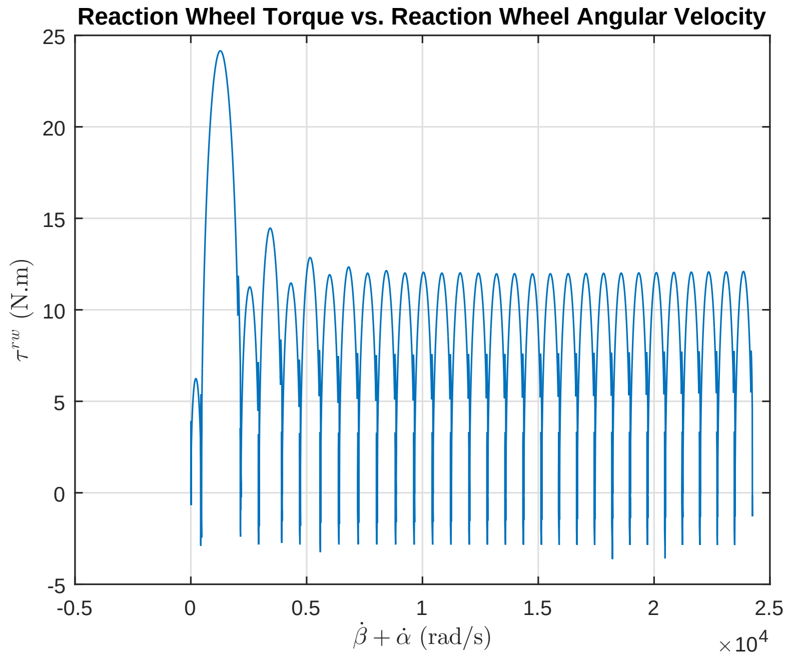

Figure 4.18: The sample simulation graph that shows torque with respect to in angular velocity for the reaction wheel under the hip torque controller for one-direction stable locomotion

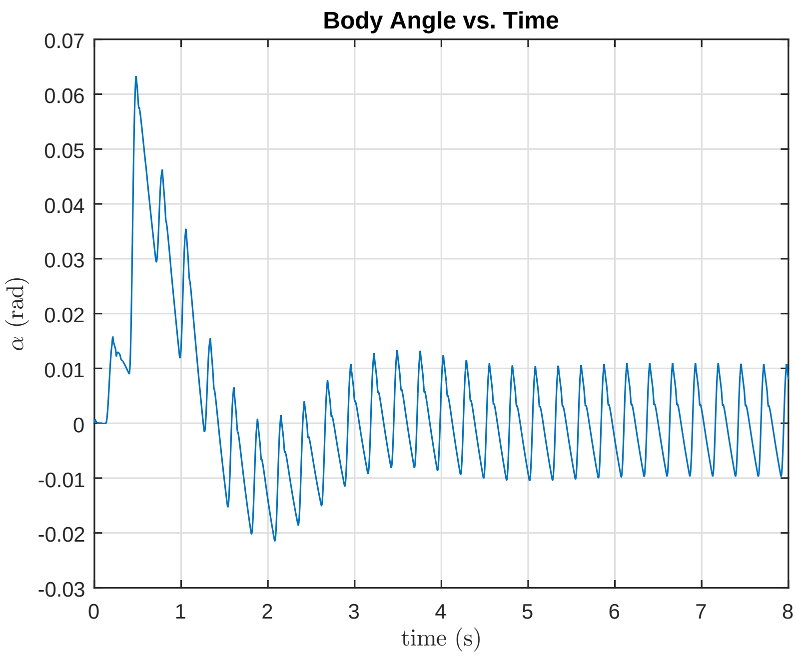

Figure 4.19: The sample simulation graph that shows body angle (PID controlled) with respect to time under the hip torque controller for one-direction stable locomotion

4.5 Discussion

In the present chapter, we modify the SLIP-T model by adding the angular degree of freedom for the body which has a finite inertia. Moreover, in order to compensate the angular disturbances on body attitude caused by the hip torque, we placed a reaction wheel at the CoM of the body. We derive the mathematical model using Newton-Euler methodology and implement the physics based hybrid simulation using Matlab environment. To discuss the feasibility of the reaction wheel approach, two different simulation scenarios are performed. In the first one, monopod's leg is balanced in standing vertical position using a PD based hip torque controller. In the second one, gait level controller from the literature is used for a stable locomotion. The simulation results show that the reaction wheel approach, even with a simple PD controller is highly feasible to compensate for the disturbances on body attitude under the hip torque controller which keeps the robot at standing position. However, we also observe that under the stable running controller, the approach becomes infeasible. The reason behind this infeasibility is the net required hip torque profiles at each stride. In flight, the hip torque brings the leg (which is modeled as point toe mass) into a certain angle. Since the toe mass is much smaller than the body mass, during flight the required hip torque is fractional comparing with the hip torque that moves the body in stance phase. Hence, the net change in body attitude occurs in the same angular direction. To generate the compensating torque which stabilizes the body attitude, reaction wheel velocity increases at each stride. This phenomena makes the approach impractical for a physical monopod platform.

CHAPTER 5

PLANAR ONE-LEGGED HOPPER WITH A NOVEL APPROACH: DRAG WHEEL

5.1 Motivation and Introduction

In the previous chapter, simulation results under the hip torque controller for one-direction stable locomotion show that the body attitude is cyclically disturbed in a single angular direction. This causes the reaction wheel to speed up in order to achieve the required torque levels for compensating angular disturbances. To prevent such an infeasible result, in this chapter we present our novel air drag assisted reaction wheel approach. Giving mathematical model of the drag wheel we update our dynamical equations that we derive in the previous chapter. Under two hip torque controllers that reaction wheel fails in practice to control the body attitude, we analyze the drag wheel concept. At the end of this chapter, in order to determine the practical feasibility of our approach, we present a physical experiment.

5.2 System Model of the Drag Wheel

This concept basically uses the aerodynamic drag created by the resisting air flow of the radial blades (Figure 5.1) which we add into our reaction wheel design. Hence, when reaction wheel speeds up, it encounters air resistance torque due to the blades. In the literature, this drag torque is calculated using the equation (5.1). From the equation, we can observe that the resistive torque is related with the square of the angular velocity. Hence, any increase in angular velocity would cause a particular drag torque. Here, in order not to affect the system dynamics of our model, we assume that the drag wheel exerts same force at every direction while it rotates. In other words, the net air drag force (not torque!) created by the wheel is assumed to be zero.

$$\tau_{drag} = \frac{1}{2}\rho C_d A \omega^2 \tag{5.1}$$

where;

τ_{drag}: Drag torque

ρ: Density of the fluid

ω: Angular velocity

C_d: Drag coefficient

A: Cross sectional area

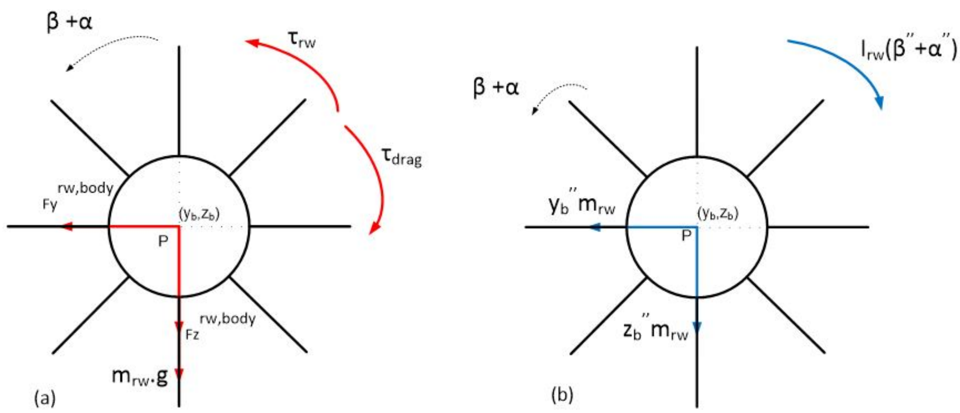

Figure 5.1: Free-body diagrams of (a) internal-external, and (b) inertial forces of the drag wheel in both flight and stance phases. β and θ are determined with respect to the body

5.3 Dynamical Equations of the Drag Wheel

Having the same free-body-diagrams for the leg and body, we only modify the reaction wheel equations in the previous chapter. The aerodynamic drag is added as a sink term for the reaction wheel torque in the right hand side and the drag wheel equations become as in (5.2). The associated free-body-diagram for the drag wheel is given in

Figure 5.1.

$$
\begin{aligned}
m_{rw}\ddot{y}_b + F_y^{rw,body} &= 0 \\
m_{rw}\ddot{z}_b + F_z^{rw,body} &= -m_{rw}g \\
I_{rw}(\ddot{\beta} + \ddot{\alpha}) &= \tau^{rw} - sgn(\dot{\beta})\tau^{drag} \\
\tau^{drag} &= K_{drag}\dot{\beta}^2
\end{aligned}
\tag{5.2}
$$

where we define our drag torque constant K_{drag} as the multiplication of all constant parameters that are specific for a particular wheel design and the environment such that:

$$K_{drag} = \frac{1}{2}\rho C_d A. \tag{5.3}$$

The only difference in state matrix of the drag wheel system from the reaction wheeled version is the known forces vectors: $f_{flight}^{dw}, f_{stance}^{dw}$. Adding the drag term, these updated force vectors can be written as:

$$f_{flight}^{dw} = \begin{pmatrix} 0 & -m_{rw}g & \tau^{rw} - K_{drag}sgn(\dot{\beta})\dot{\beta}^2 & 0 & -m_b g & -\tau^{rw} - \tau^{hip} & 0 & -m_t g & -k(\rho - \rho_0) - d\dot{\rho} & \tau^{hip} - m_t g \rho sin(\theta + \alpha) \end{pmatrix} \tag{5.4}$$

$$f_{stance}^{dw} = \begin{pmatrix} 0 & -m_{rw}g & \tau^{rw} - K_{drag}sgn(\dot{\beta})\dot{\beta}^2 & 0 & -m_b g & -\tau^{rw} - \tau^{hip} & 0 & m_t g & -\tau^{hip} & -k(\rho - \rho_0) - d\dot{\rho} \end{pmatrix} \tag{5.5}$$

In the equations (5.4, 5.5), the sgn term is added since the drag force affects the wheel in the opposite direction of its angular velocity.

5.4 Physics Based Hybrid System Simulation

Having the updated vector field for the drag wheeled system, we generate the physics based hybrid system simulation as in the previous chapter. In order to compare the drag wheel with the reaction wheel approach, we present two scenarios. This allows us to make one-to-one comparison between the reaction wheel and the novel drag wheel design. Both simulation scenarios are revealed that when K_{drag} is in the range of $[0.0001, 0.001]$, the concept can be used bounding the reaction wheel speed as well as providing required compensation torque values. Although the required drag torque constant K_{drag} is found empirically in the simulations, in the Section 5.5 we discuss the feasibility of this term further by giving some physical experiment results. In two scenarios, we adjusted the drag torque constant as $K_{drag} = 4.10^{-4}$.

5.4.1 Using the hip torque controller that keeps the robot's leg at a constant angle

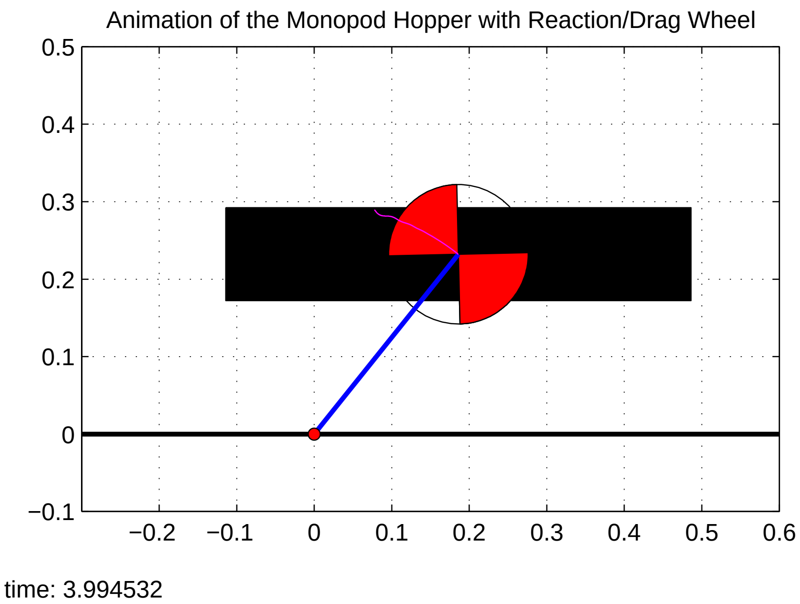

Figure 5.2: Snapshot from the animation of the planar monopod hopper with reaction/drag wheel

In the first scenario, we set up the hip torque controller to keep the robot's leg at a constant angle (Figure 5.2). We use a PD controller in the form of:

$$\tau^{hip} = -k_p((\theta+\alpha)^{des} + \theta + \alpha) - k_d(\dot{\theta} + \dot{\alpha}). \tag{5.6}$$

where the control parameters and the desired leg angle adjusted as $k_p = 2000$, $k_d = 500, \ (\theta+\alpha)^{des} = \dfrac{-30 * 180}{pi} rad = -30^o$

Initially the system is started from stance phase with an initial leg angle $((\theta+\alpha)_0 = -15^o)$.

From the simulations, we observe that some residual angle is left when we use PD controller for the reaction/drag wheel. Hence, in this chapter we use PID method to control the reaction/drag wheel. The PID controller equation and control parameters are given as:

$$\tau^{rw} = -k_{prw}\alpha - k_{drw}\dot{\alpha} - k_{irw}\int_0^t \alpha(t)dt$$
$$where \qquad k_{prw} = 40, k_{drw} = 20, k_{irw} = 100. \tag{5.7}$$

Table 5.1: Initial Conditions and System Parameters (Standing at a constant angle controller scenario)

(a)

Symbol	Initial Value and Unit
y_b	0.0776 m
z_b	0.2898 m
y_t	0 m
z_t	0 m
β	0 rad
α	0 rad
$\dot{y}_b$	0 $\frac{m}{s}$
$\dot{z}_b$	0 $\frac{m}{s}$
$\dot{y}_t$	0 $\frac{m}{s}$
$\dot{z}_t$	0 $\frac{m}{s}$
$\dot{\beta}$	0 $\frac{rad}{s}$
$\dot{\alpha}$	0 $\frac{rad}{s}$

(b)

Symbol	Value and Unit
g	9.81 $\frac{m}{s^2}$
k	15000 $\frac{N}{m}$
d	100 $\frac{Ns}{m}$
m_{rw}	1 kg
m_b	2 kg
m_t	0.2 kg
r_{rw}	0.09 m
b_w	0.06 m
b_l	0.3 m
I_{rw}	0.004 $kg.m^2$
I_b	0.0156 $kg.m^2$

Starting the system from the initial conditions (Table 5.1) and the hip torque controller (equation 5.6) brings the leg to the desired angle as shown in Figure 5.3.

In the system simulation, the hip torque controller applies the necessary torque values to keep the leg at a constant standing angle. At the same time, these torque profiles continuously disturb the body angle. Using the PID controller in both reaction and drag wheeled systems, the body attitude of the robot is stabilized successfully in the horizontal position(Figure 5.4). Moreover, the phase portrait of the system shows that both body angular velocity and body angle converges to zero (Figure 5.5). However, to compensate the angular disturbances, the reaction wheel speeds up impractically (Figure 5.6). On the other hand, drag torque slows down the wheel when it speeds up. Hence, drag wheel velocity keeps bounded while the necessary torque levels are achieved.

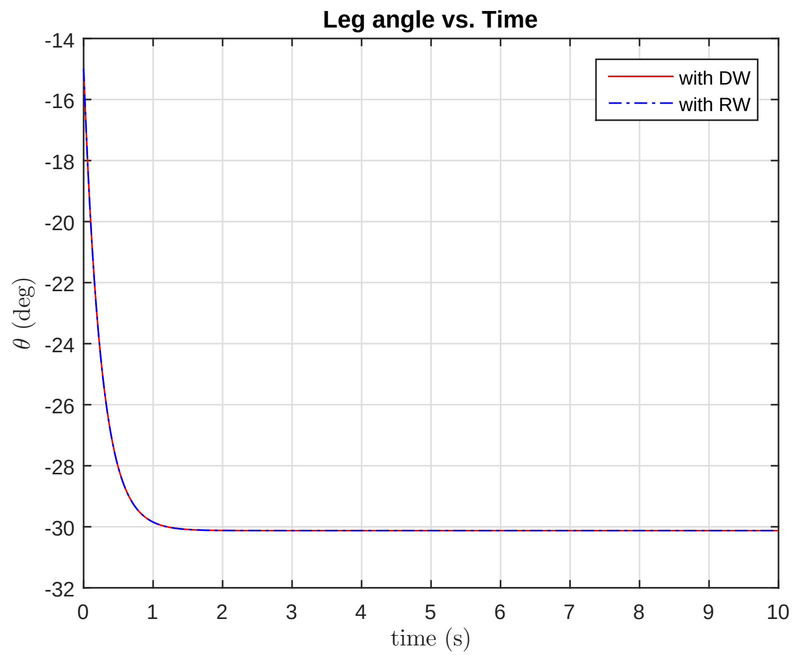

Figure 5.3: Leg Angle with respect to time under the hip torque controller which tries to keep the robot's leg at a constant angle

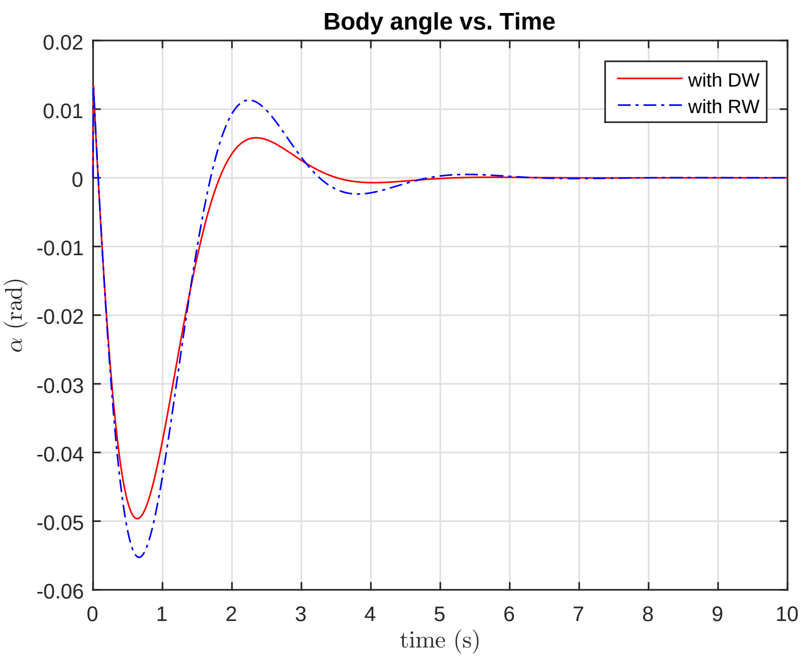

Figure 5.4: Body angle with respect to time under the hip torque controller which tries to keep the robot's leg at a constant angle

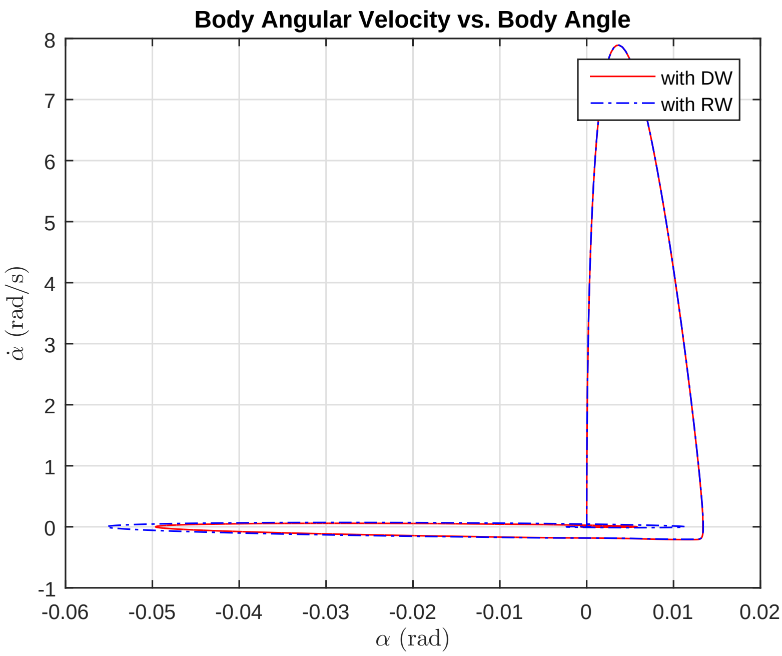

Figure 5.5: Angular velocity of the body with respect to body angle under the hip torque controller which tries to keep the robot's leg at a constant angle

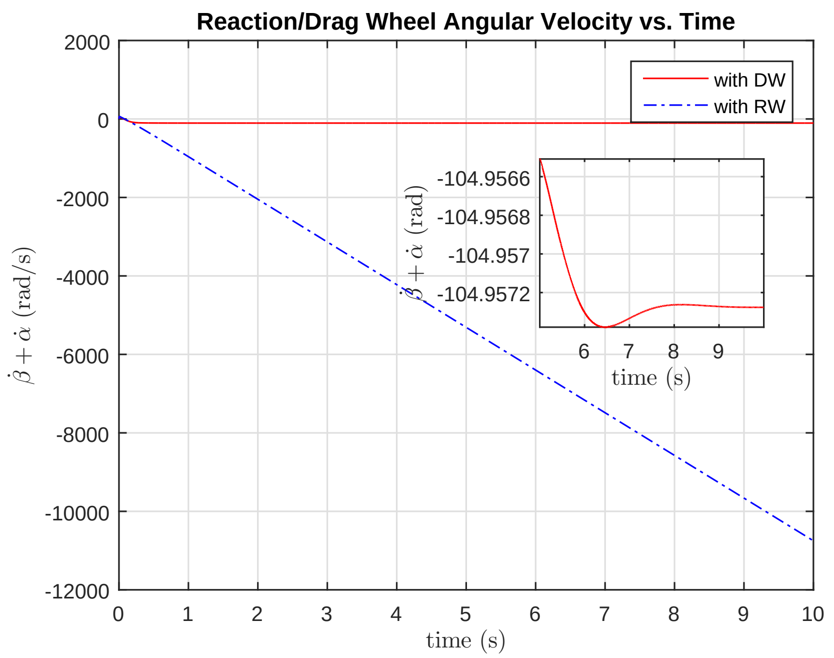

Figure 5.6: Angular velocity of the reaction/drag wheel with respect to time under the hip torque controller which tries to keep the robot's leg at a constant angle

5.4.2 Using the hip torque controller for one-direction stable locomotion

In the second scenario, the same initial conditions and system parameters (Table 4.5) and the same hip torque controller used in Chapter 4 which enables stable running for the monopod in one direction.

Simulations show that, the gradually increasing angular speed of the reaction wheel is now successfully kept bounded around realistic speed levels (Figure 5.7) and the body attitude preserved in horizontal direction using PID controller. Moreover, the unstable periodic characteristic torque-velocity graph using reaction wheel control in previous chapter, now using the drag wheel approach converges to a point where both angular velocity and torque value become zero as shown in Figure 5.8.

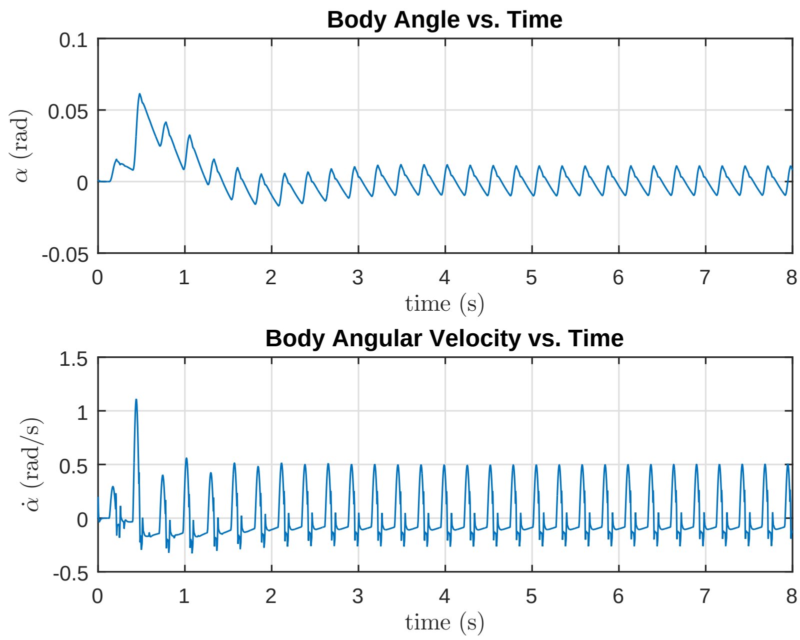

Figure 5.7: The sample simulation graph that shows (top) body angle (bottom) angular velocity of the drag wheel with respect to time under the hip torque controller for one-direction stable locomotion

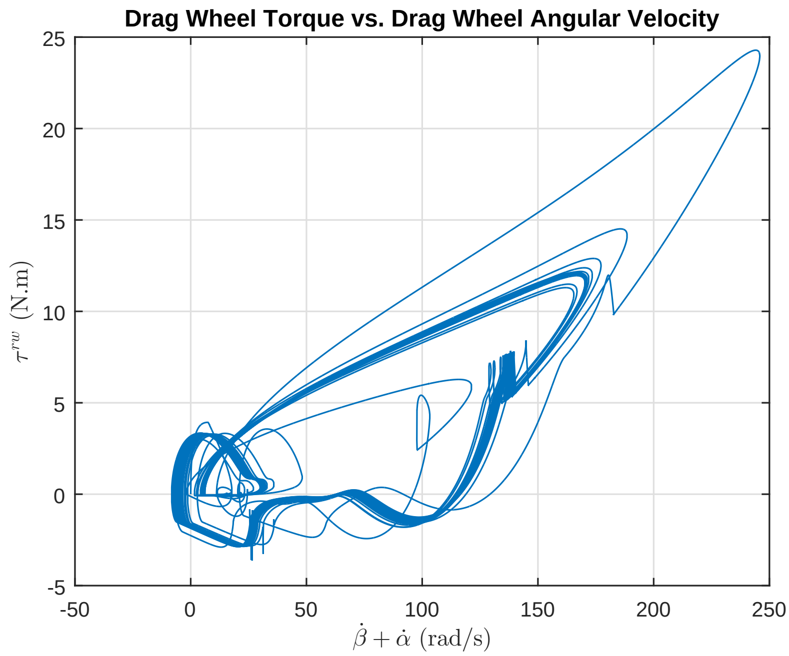

Figure 5.8: The sample simulation graph that shows torque values with respect to the angular velocity for the drag wheel under the hip torque controller for one-direction stable locomotion

5.5 Feasibility of the Drag Wheel

In the previous section through mathematical model based simulations, we show that the drag wheel concept can be used to bound the reaction wheel velocity while obtaining required torque values. In simulations, we adjust the drag torque constant in order to reach the desired torque profiles for compensating angular disturbances caused by the hip torque. In this context, one can ask whether our drag coefficients can be practically achievable. Hence in this section we try to answer this question setting an experiment with a simple drag wheel design.

5.5.1 Experimental Setup and Drag Wheel Design

Our experiment setup consists of (Figure 5.9):

(1) Maxon EC-45 Flat Brushless DC Motor

(2) Maxon EPOS-2 70/10 Digital Positioning Motor Controller

(3) Drag Wheel

(4) Power Supply

(5) Computer

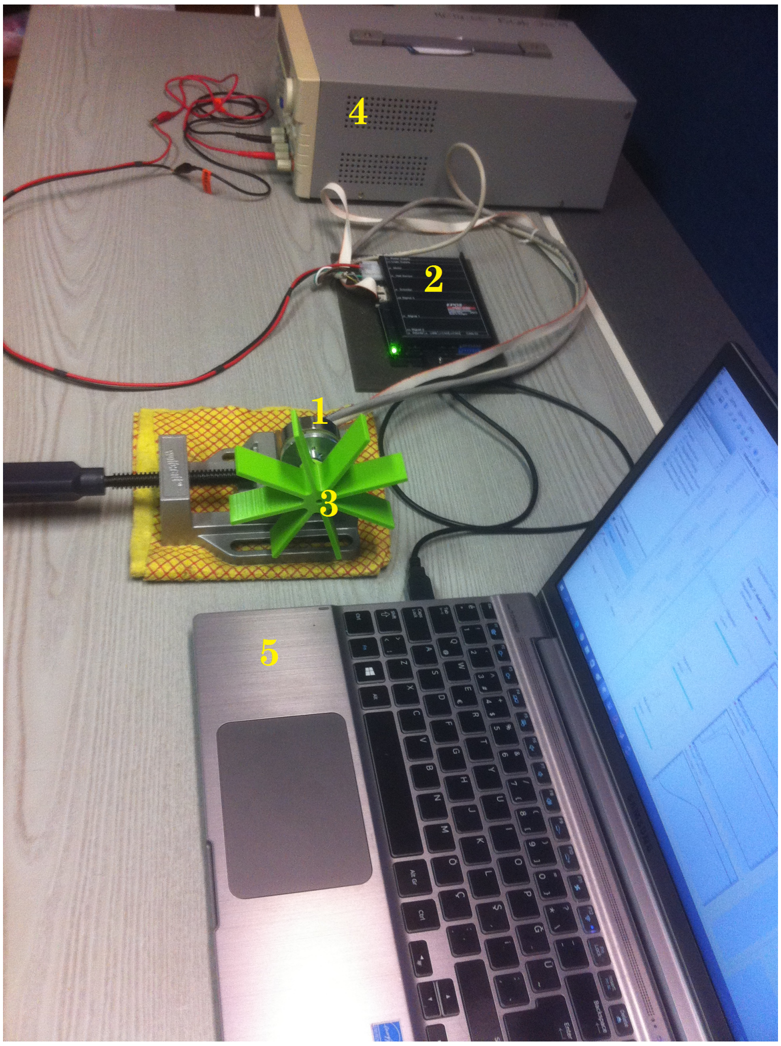

Figure 5.9: Experimental setup to calculate drag torque constant. (1) Motor, (2) Motor controller, (3) Drag wheel, (4) Power supply, (5) Computer

We design a symmetric drag wheel which has an eight propeller. To obtain high air resistance, each propeller consists of wide surface area in the rotational direction. We make the design by Solidworks [37] and built it using a 3D printer. The design details can be found in figures 5.10, 5.11 and 5.12.

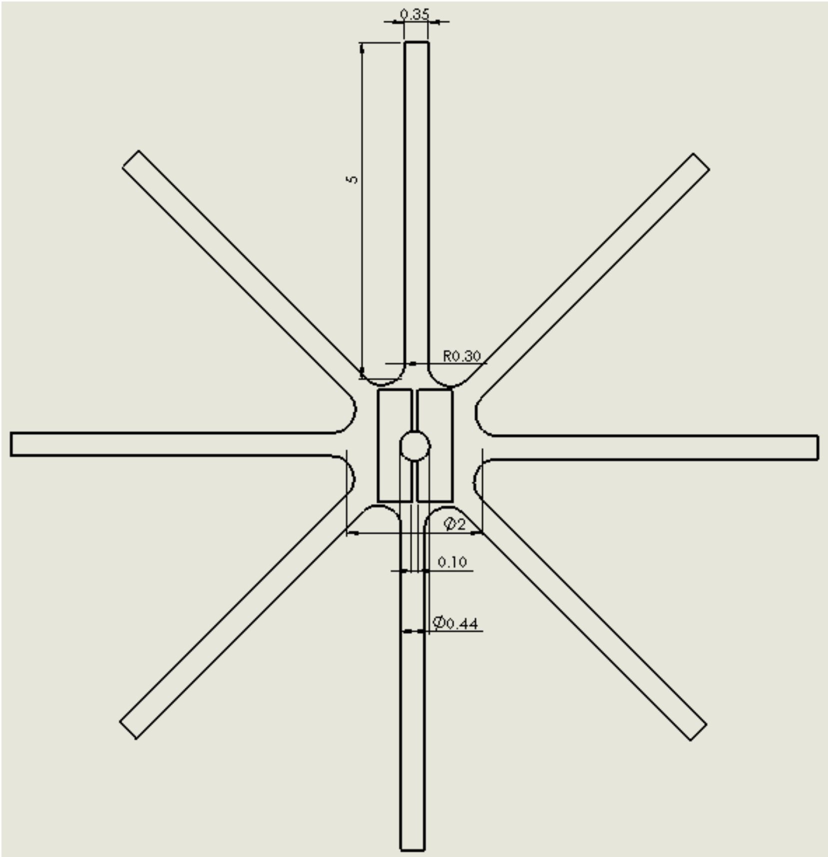

Figure 5.10: Front view of the drag wheel design

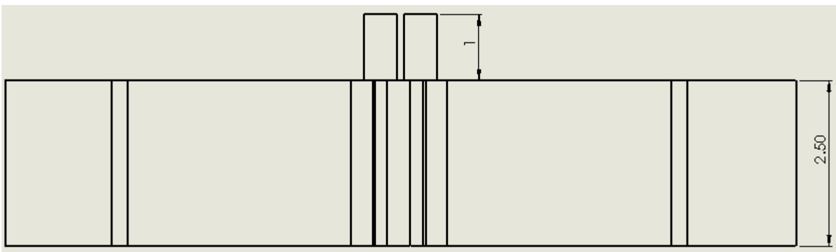

Figure 5.11: Rear view of the drag wheel design

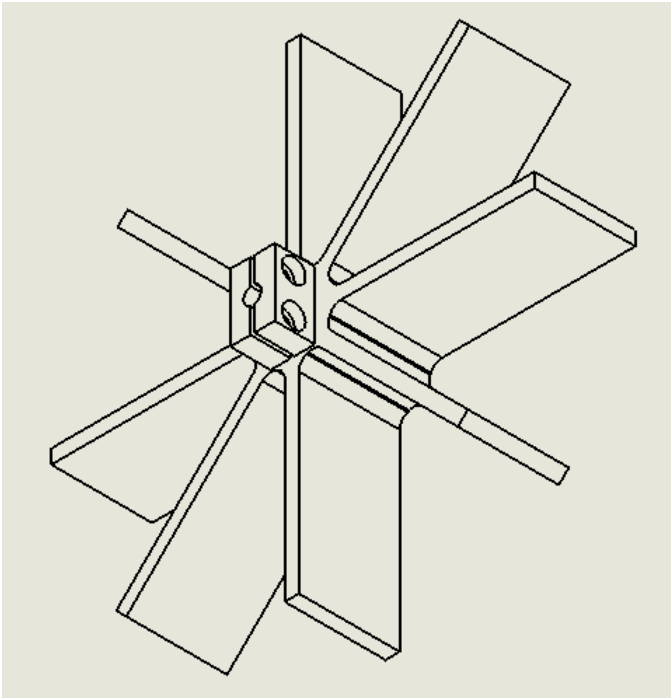

Figure 5.12: Isometric view of the drag wheel design

5.5.2 Methodology to Calculate the Drag Torque Constant

In order to evaluate the drag torque constant which is adjusted at the start of the simulations, we set up an experiment. Our methodology to calculate the drag torque constant K_{drag} is:

1) Find the current that motor draws without drag wheel with respect to different velocity values (Figure 5.13)
2) Find the current that motor draws with drag wheel with respect to different velocity values (Figure 5.13)
3) Find the difference in currents and multiply with the torque coefficient to find the drag torque generated by the wheel (Figure 5.14 and 5.15)
4) Find the drag torque constant using the equation (5.1) (Figure 5.16)

It is clear from the Figure 5.16 that the drag torque constant ($K_{drag} = 1.10^{-4}$) used in the simulations and the constant that we have obtained from the experiment matches. Hence, we have obtained the required drag torque constant using a simple impeller design in order to bound the angular velocity of the reaction wheel.

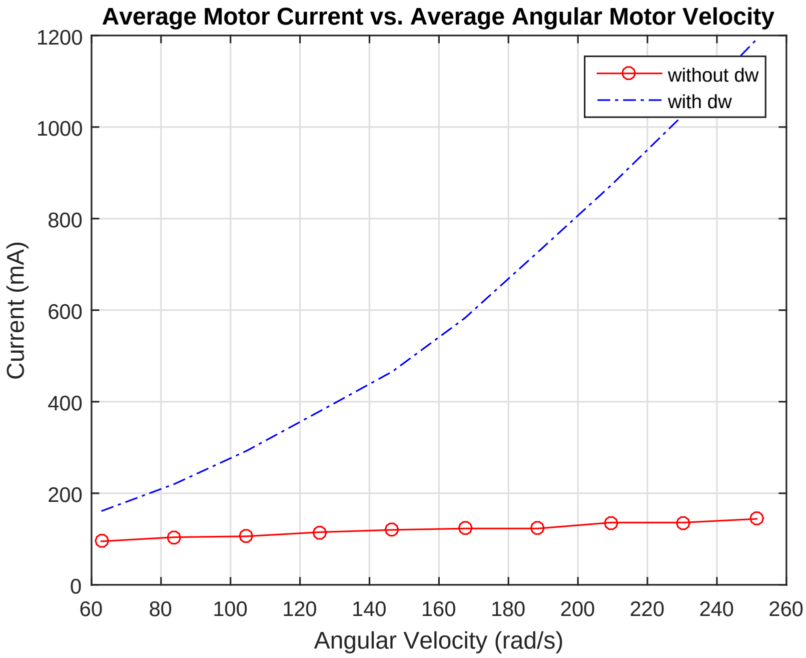

Figure 5.13: The graph that shows the average current drawn by the motor with respect to average angular velocity with (blue) or without (red) the drag wheel

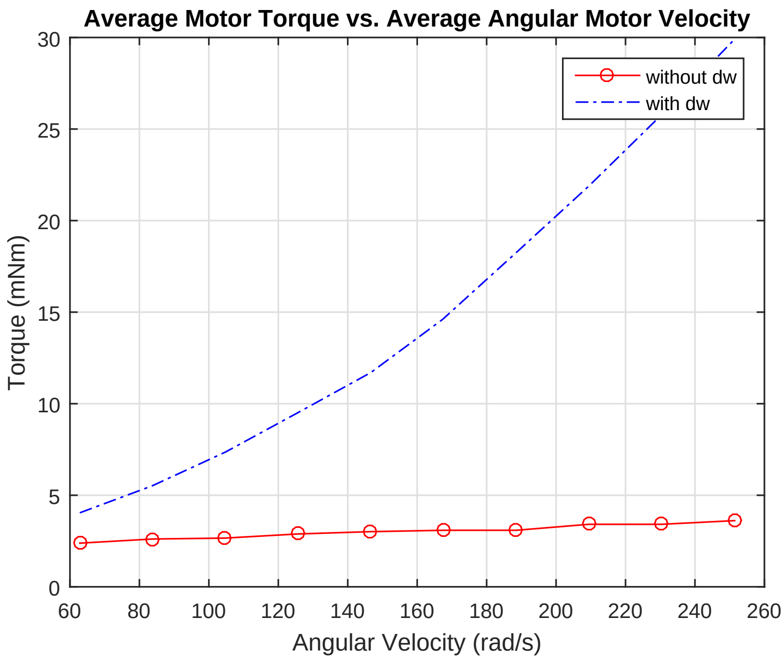

Figure 5.14: The graph that shows the average torque generated by the motor with respect to average angular velocity with (blue) or without (red) the drag wheel

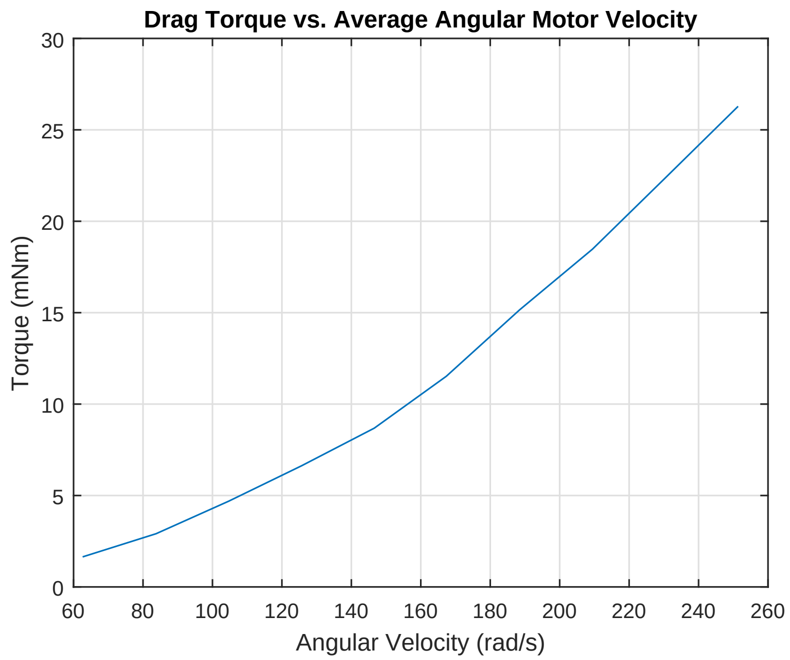

Figure 5.15: The graph that shows the drag torque generated using the drag wheel with respect to average angular velocity

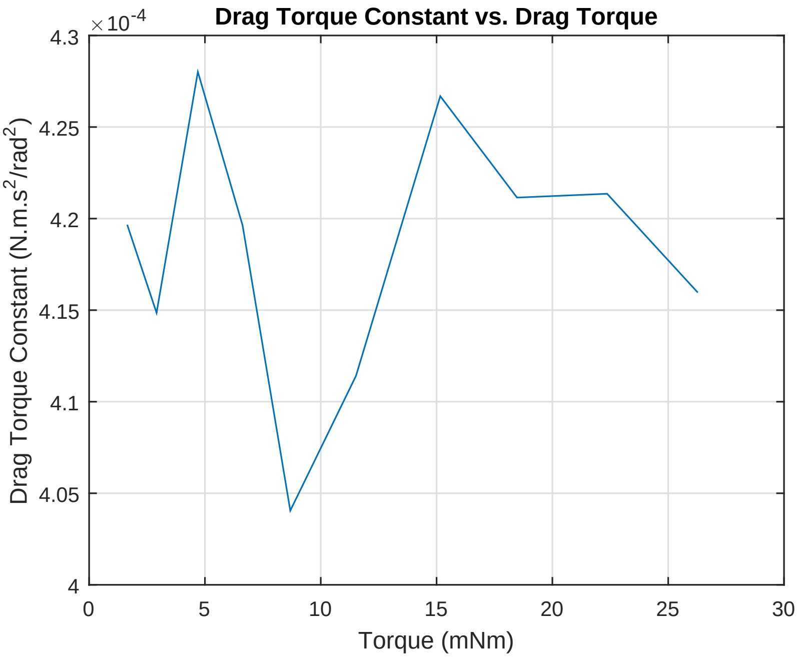

Figure 5.16: The graph that shows the calculated drag torque constant with respect to the drag torque generated using the drag wheel

5.6 Discussion

In order to supply the required torque, reaction wheel needs to be accelerated. Practically there is a limit velocity for every reaction wheel system. Hence, at every instance if the required torque is just in the same angular direction, the reaction wheel approach become infeasible due to the unbounded angular velocity. The present chapter introduces a novel reaction wheel design which aims to limit the wheel speed using the drag torque. Having the radial blades in our wheel design, we obtain a drag torque which is proportional to the square of the angular velocity of the wheel. Hence, we able to bound the reaction wheel velocity. We update the dynamical system equations for the monopod hopper defined in Chapter 4 by adding the drag term. To compare both approaches (reaction and drag wheel), we implement physics based system simulations for the monopod hopper as in the previous chapter. In the simulations, we choose two scenarios in which the body attitude of the robot is disturbed in the same angular direction. Both approaches compensate the angular disturbances of the body successfully. However, to reach the torque profiles, reaction wheel velocity increases in an unbounded manner as expected. On the other hand, drag wheel velocity is bounded due to the drag torque. In the dynamical equations, the drag torque constant term determines the magnitude of the drag torque. Thus, it is reasonable to ask whether this constant term that we adjusted in the simulations is feasible. Having a simple 3D printed drag wheel, we have done a physical experiment in order to find its drag torque constant. Since the experimentally found one and the adjusted one in the simulations matches, we think that the drag wheel concept is not only solve our problem related with the unbounded angular velocity levels that we encounter in simulations but also practically feasible.

CHAPTER 6

CONCLUSION

6.1 Summary

In this thesis, the body attitude stabilization of a hip-torque actuated planar hopping monopod robot is considered. To represent and analyze this system, we made use of the SLIP-T model and its associated gait level locomotion controller from the literature. SLIP-T is a hip-torque based extension of the well-known planar SLIP model. Since the existing locomotion controller assumes an infinite body inertia, the time varying hip-torque applied by the controller does not result in a body attitude change. This is not a realistic assumption for a physical robot since for any finite inertia, the torque applied to the leg will result in a net change in the robot body attitude at the end of each stride. We attempt to overcome this assumption in our study by considering a reaction wheel approach. Here, the torque generated by the reaction wheel mounted on the center-of-mass (CoM) of the body is used to correct the attitude disturbance caused by the leg torque. To develop this solution, we firstly consider a reaction wheel pendulum model from the literature and its Lagrangian formulation. This is useful because the hopping robot, during the stance phase of its locomotion behaves in accordance with an extension of this model by the use of a leg spring and leg torque. We develop this extension by analyzing the augmented system (robot and reaction wheel) using Newton-Euler formulation and derive its hybrid dynamic equations in its flight and stance phases. Furthermore, we implement a physics based hybrid simulation of this model and perform its simulation based experimental analysis in two different motion scenarios. In one scenario, the robot is balanced on its leg to stand upright while in the second scenario, the robot moves in a regulated (cyclic) forward velocity and hopping height. For the hip torque control, we use a simple PD controller for the first scenario while using a proposed locomotion controller from the literature for the second scenario. For both scenarios, experimentally tuned PD and PID control alternatives are investigated for the control of the reaction wheel motor torque and our results are presented for each case. Our results demonstrate that even a PD controller for the reaction wheel is sufficient to keep the robot attitude horizontal for the first scenario. However, for the stable locomotion scenario, a PD controller results in a steady-state error in body attitude and a PID controller is needed to bring the attitude to horizontal. However, the reaction wheel velocities required to achieve this are unbounded (hence not feasible) in both cases. To solve this problem, we propose a novel "air drag assisted reaction wheel", implemented as an impeller design which can generate torque that is also a function of the wheel rotational velocity. With this, we demonstrate that attitude can be regulated together with a bounded wheel velocity. We demonstrate using simulation results that the robot model can maintain stable locomotion as well as horizontal body attitude while the reaction wheel velocity is within achievable limits. We also present using physical experiments that this can be done by a reasonably sized impeller mounted on the reaction wheel.

6.2 Future Work

Throughout the thesis, we use PD and PID based control methods for the reaction/-drag wheel. The control coefficients of these methods adjusted empirically in the simulations. For the first future extension of the present work, these coefficients would be obtained through some optimization. But before the optimization, we need to define a performance criteria for our system. As a second extension, we would like to verify the simulation results for the monopod hopper with the physical experiments. To show the significance of the drag wheel concept against the reaction wheel, we would set up an experiment on an inverted pendulum. Using the drag wheel, we would try to control the pendulum at a constant angle (other than vertical position) which cannot be achieved using a regular reaction wheel. After that we would like to test the drag wheel on a planar hopper robot. Moreover in the long term, the planar dynamical equations and also the hybrid system simulation would be extended to 3D.

REFERENCES

[1] SimMechanics model and simulate multibody mechanical systems. http://www.mathworks.com/products/simmechanics/. Accessed: 2016-11-02.

[2] Arvind Ananthanarayanan, Mojtaba Azadi, and Sangbae Kim. Towards a bio-inspired leg design for high-speed running. *Bioinspiration & biomimetics*, 7(4):046005, 2012.

[3] Mert Ankaralı. Control of Hexapedal Pronking Through a Dynamically Embedded Spring Loaded Inverted Pendulum Template. Master's thesis, Middle East Technical University, 2010.

[4] M Mert Ankaralı and Uluc Saranlı. Control of underactuated planar pronking through an embedded spring-mass hopper template. *Autonomous Robots*, 30(2):217–231, 2011.

[5] Mark D Ardema. *Newton-Euler Dynamics*. Springer Science & Business Media, 2006.

[6] O. Arslan, U. Saranli, and O. Morgul. An approximate stance map of the spring mass hopper with gravity correction for nonsymmetric locomotions. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 2388–2393, May 2009.

[7] Omür Arslan. Model based methods for the control and planning of running robots. Master's thesis, Bilkent University, 2009.

[8] ZY Bayraktaroglu. Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot. *Mechanism and Machine Theory*, 44(3):591–602, 2009.

[9] Daniel J Block, Karl J Åström, and Mark W Spong. The reaction wheel pendulum. *Synthesis Lectures on Control and mechatronics*, 1(1):1–105, 2007.

[10] Sean Goodwin Carver. *Control of a spring-mass hopper*. Cornell University, 2003.

[11] Nicholas Cherouvim and Evangelos Papadopoulos. Speed control of quadrupedal bounding using a reaction wheel. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006*

IEEE International Symposium on Intelligent Control, 2006 IEEE, pages 2160–2165. IEEE, 2006.

[12] Avik De and Daniel E Koditschek. Parallel composition of templates for tail-energized planar hopping. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4562–4569. IEEE, 2015.

[13] Jared Diamond. Why animals run on legs, not on wheels. *Discover*, 4(9):64–67, 1983.

[14] Stephen J Dodds and Jan Vittek. Spacecraft attitude control using an induction motor actuated reaction wheel with sensorless forced dynamic drive. 1998.

[15] Robert J Full and Daniel E Koditschek. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202(23):3325–3332, 1999.

[16] Mohanarajah Gajamohan, Michael Merz, Igor Thommen, and Raffaello D'Andrea. The cubli: A cube that can jump up and balance. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3722–3727. IEEE, 2012.

[17] Shengmin Ge and Hao Cheng. A comparative design of satellite attitude control system with reaction wheel. pages 359–364, June 2006.

[18] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Spring-mass running: simple approximate solution and application to gait stability. *Journal of theoretical biology*, 232(3):315–328, 2005.

[19] Reinhard Grundmann, Friedrich Schönholtz, Dipl-Ing FH Herbert Eidam, and B Rahn. Grundlagen der ventilatorentechnik. *TLT-Turbo GmbH, Bad Hersfeld*, 2005.

[20] Philip Holmes. Poincaré, celestial mechanics, dynamical-systems theory and chaos. *Physics Reports*, 193(3):137–163, 1990.

[21] Philip Holmes, Robert J Full, Dan Koditschek, and John Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *Siam Review*, 48(2):207–304, 2006.

[22] Michael LaBarbera. Why the wheels won't go. *American Naturalist*, pages 395–408, 1983.

[23] VJ Lappas, WH Steyn, and CI Underwood. Attitude control for small satellites using control moment gyros. *Acta Astronautica*, 51(1):101–111, 2002.

[24] Sung-Hee Lee and Ambarish Goswami. Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots. In *Robotics*

and Automation, 2007 IEEE International Conference on, pages 4667–4672. IEEE, 2007.

[25] Fei Li, Weiting Liu, Xin Fu, Gabriella Bonsignori, Umberto Scarfogliero, Cesare Stefanini, and Paolo Dario. Jumping like an insect: Design and dynamic optimization of a jumping mini robot based on bio-mimetic inspiration. *Mechatronics*, 22(2):167–176, 2012.

[26] Jinsu Liu, Xiaoping Chen, and Manuela Veloso. Simplified walking: A new way to generate flexible biped patterns. In *Proc. of 12th Int. Conf on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. World Scientific, 2009.

[27] MATLAB. *version 7.12.0 (R2011a)*. The MathWorks Inc., Natick, Massachusetts, 2011.

[28] MATLAB. *version 8.5.0.197613 (R2015a)*. The MathWorks Inc., Natick, Massachusetts, 2015.

[29] Philipp Michel, Joel Chestnutt, Satoshi Kagami, Koichi Nishiwaki, James Kuffner, and Takeo Kanade. Gpu-accelerated real-time 3d tracking for humanoid locomotion and stair climbing. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 463–469. IEEE, 2007.

[30] Kazuhisa Mitobe, Genci Capi, and Yasuo Nasu. Control of walking robots based on manipulation of the zero moment point. *Robotica*, 18(06):651–657, 2000.

[31] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture point: A step toward humanoid push recovery. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 200–207. IEEE, 2006.

[32] Jerry Pratt and Gill Pratt. Intuitive control of a planar bipedal walking robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 2014–2021. IEEE, 1998.

[33] Marc H Raibert. Legged robots that balance. 1985.

[34] Uluc Saranlı. *Dynamic locomotion with a hexapod robot*. PhD thesis, The University of Michigan, 2002.

[35] Umberto Scarfogliero, Cesare Stefanini, and Paolo Dario. The use of compliant joints and elastic energy storage in bio-inspired legged robots. *Mechanism and Machine Theory*, 44(3):580–590, 2009.

[36] William John Schwind. *Spring loaded inverted pendulum running: a plant model*. University of Michigan, 1998.

[37] SolidWorks. *Premium 2013 x64 Edition*. SolidWorks Corporation, Waltham, Massachusetts, 2013.

[38] Mark W Spong. The swing up control problem for the acrobot. *Control Systems, IEEE*, 15(1):49–55, 1995.

[39] Mark W Spong and Daniel J Block. The pendubot: A mechatronic system for control research and education. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, volume 1, pages 555–556. IEEE, 1995.

[40] Mark W Spong, Peter Corke, and Rogelio Lozano. Nonlinear control of the reaction wheel pendulum. *Automatica*, 37(11):1845–1851, 2001.

[41] Mark W Spong, Seth Hutchinson, and M Vidyasagar. Robot modeling and control. 2006.

[42] Barkan Ugurlu and Atsuo Kawamura. Bipedal trajectory generation based on combining inertial forces and intrinsic angular momentum rate changes: Eulerian zmp resolution. *Robotics, IEEE Transactions on*, 28(6):1406–1415, 2012.

[43] Ronny Votel and Doug Sinclair. Comparison of control moment gyros and reaction wheels for small earth-observing satellites. 2012.

[44] Meng Wang, Xi-zhe Zang, Ji-zhuang Fan, and Jie Zhao. Biological jumping mechanism analysis and modeling for frog robot. *Journal of Bionic Engineering*, 5(3):181–188, 2008.

[45] Xin Wang, Mantian Li, Pengfei Wang, Wei Guo, and Lining Sun. Bio-inspired controller for a robot cheetah with a neural mechanism controlling leg muscles. *Journal of Bionic Engineering*, 9(3):282–293, 2012.

[46] Garth Zeglin. *The bow leg hopping robot*. PhD thesis, Carnegie Mellon University Pittsburgh, 1999.