

Running Robots: Modeling, Control and Energy Efficiency

Pouyan Taghipour Bibalan
(u4422921)

Supervisor : Dr. Roy Featherstone

School of Engineering
The Australian National University

A thesis submitted in partial fulfilment
of the degree of Master of Engineering at the Australian National University

June 2009

To my family

Acknowledgements

I would like to thank Dr. Roy Featherstone for his endless support during the course of this project. I would also like to thank anyone who directly or indirectly contributed to the successful completion of the project.

Abstract

Legged robots are types of mechanical machines that have a greater capability to move over difficult terrains than their wheeled counterparts. At the beginning of the development of legged robot research, the focus was mainly applied to increased versatility of these machines. Today, this field has reached a point where energy efficiency of locomotion has come to the forefront of research studies. Therefore, this thesis is dedicated to the energetic aspects of a specific branch of legged robots, i.e., the hopping and running ones. This has been achieved through detailed dynamics modeling and stable control of a monopod hopper as well as a bipedal runner. Detailed dynamics simulations have been made possible by accurately modeling the subsystems that comprise a hopping/running robot simulator. In particular, the focus has been given to the accurate modeling of the robot's actuators and the foot-ground interaction. The end result has been a hopping simulator which is sufficient for performing energy analysis, along with 3D visualization tools which greatly enhance fault detection and behavior analysis.

Contents

Acknowledgements	i
Abstract	ii
Table of Contents	iii
List of Tables	v
List of Figures	vi
1 Introduction	1
1.1 Background	1
1.2 Scope of the Project	2
1.3 Author's Contributions	3
1.4 Organization of the Thesis	4
2 Robot Model	7
2.1 Dynamics Modeling of the Robot Structures	7
2.1.1 Monopod Hopper	8
2.1.2 Bipedal Runner	11
2.2 Foot Model	13
3 Control System	14
3.1 Monopod Hopper	14
3.2 Bipedal Runner	18
4 Actuation System	19
4.1 DC Motors	20
4.1.1 Brushed DC Motor	20
4.1.2 Brushless DC Motor (BLDCM)	22
4.2 Transmission and Reduction System	24
	iii

4.3	Monopod Hopper Actuation System	26
5	Contact and Friction	32
5.1	Contact and Impact	33
5.1.1	Rigid Body Models	33
5.1.2	Compliant Models	33
5.2	Friction	37
5.2.1	Static Friction Models	39
5.2.2	Dynamic Friction Models	41
5.3	Final Choice of Contact and Friction Models	45
6	Simulation and Results	48
6.1	Contact and Friction Force	48
6.2	Control	54
6.3	Energetics	56
6.4	Suggested Improvements	63
7	Conclusion	67
	Appendix	69
	Bibliography	70

List of Tables

1.1	Some of the famous legged robots since 1970	2
2.1	Physical parameters of the planar one-legged hopper	11
3.1	State-Machine events and actions	16
4.1	DC motor specifications	30
5.1	Some important qualitative behaviors of friction	38
5.2	Parameter values for the compliant contact models	46
5.3	Parameter values for the friction models	47
6.1	Motor efficiency results	57

List of Figures

1.1	Pictures of some of the famous legged robots	5
1.2	Functional decomposition of a hopping/running robot	6
2.1	Graphical model of the 5 DOF monopod	10
2.2	Connectivity graph of the 5 DOF monopod	10
2.3	Monopod hopper with DC motors as actuators (Leg configuration 1)	11
2.4	Graphical model of the 7 DOF biped	12
2.5	Connectivity graph of the 7 DOF biped	12
2.6	Rolling foot example	13
3.1	Hopping states used in the State-Machine	15
3.2	Neutral position vs. Forward speed	18
4.1	Brushed DC motor equivalent circuit and free-body diagram	22
4.2	Leg actuator variables	31
5.1	Force diagram for a block on the ground	32
5.2	Linear spring-damper contact model	35
5.3	Double-state visco-elastic contact model	37
5.4	Friction behaviors	39
5.5	Classical friction models	42
5.6	The friction interface as a contact between bristles	43
6.1	Contact and friction force for two experimental runs of the hopper	50
6.2	Foot-tip horizontal and vertical positions	51
6.3	Phase diagram of the hopper	52
6.4	Impact response characteristics of the nonlinear spring-damper model	53
6.5	Structure of the controller	54
6.6	Height tracking of the monopod hopper	55
6.7	Velocity tracking of the monopod hopper	56

6.8	Motor torque/power/efficiency vs. speed	58
6.9	Input and output energy for the hip and leg motors	59
6.10	Cost of Transport graph	61
6.11	Specific Resistance graph	62
6.12	Artificial muscle technology	63
6.13	Leg configuration 2	64
6.14	Input and output energy for two leg configurations	65
6.15	Comparison of the hip motor input energy for two control techniques	66

Chapter 1

Introduction

1.1 Background

The design and study of mechanical machines has always been of interest to humans. Historically Leonardo Davinci is perhaps one of the first people who formally studied these machines. One of the most attractive branches of this field is legged robots, pioneered by Kato and Vukobratovic in the 1970's. The reason why legged locomotion came to the forefront of research study at this late time was because wheeled locomotion surpassed it in some important features such as ease of control and energy efficiency. Firstly, the stability and control of a legged robot was far more challenging than a wheeled one, and secondly its energy efficiency was much lower. However, the need for versatility, a characteristic of human beings which was missing in wheeled robots, contributed to the realization of the need for these machines. The perception that legged robots can move in rough terrains, or environments with discontinuous supports, as opposed to wheeled robots, motivated the advancement of these machines. Other than that, the attractive behaviors that could be achieved by machines that operate in anthropomorphic or animal-like ways had great implications both sociologically and commercially [69]. As a result, several prototypes of these machines have been built since the 1970's mainly to mimic human and animal behavior. Table 1.1 lists some of the most famous legged robots built in different parts of the world and some pictures of these robots are shown in Figure 1.1 (see p. 5). A literature review on legged robot research can be found in [38]. For a complete list of these types of robots, readers are encouraged to visit [7].

At the beginning of the development of legged robots, the focus was mainly applied to increased versatility of these machines. Today the field of legged robotics has reached a point where energy efficiency is as important as, if not more important than, its versatility. Therefore, this thesis is organized around the study of the energy efficiency of a particular branch of these machines, that is, running robots and particular monopod hoppers. The es-

Robot Name	Year	Person	Institute
WBOT	1973	I.Kato	Waseda University Japan
ZMP baised robot	1972	M. Vukobratovic	Mihailo Puppín Institute Yugoslavia
Hopping robot	1980's	M.Raibert	Carnegie Mellon University
Passive Walker	1990's	McGeer	Simon Fraser University
ARL Monopod II	1994	-	McGill University
Rabbit[10]	2003	-	Several French Research Laboratories
Denise	2004	-	Delft University of Technology
New ASIMO	2005	-	Honda Motor Co
Big Dog	2005	-	Boston Dynamics
QRIO	2006	-	Sony

Table 1.1: Some of the famous legged robots since 1970

stantial difference between a walking and a running motion is the nature of the foot-ground interaction. Basically, running involves a flight phase where there is no contact between the leg and the ground, and a support phase where only one leg is on the ground, whereas walking involves a one-leg and a two-leg support phase, but no flight phase. Hopping, skipping and jogging would all be classified as kinds of running behavior. Although a running robot is not statically stable at any phase whereas a walking robot can be statically stabilized, it has been found in practice to be easier to analyze and control a running robot. Therefore, to be able to focus more on energy efficiency and bypass hard control challenges, the subject of this thesis is restricted to running robots. We investigate the modeling, control, and energy efficiency of these robots by means of detailed dynamics simulation. This is made possible by the wise selection and accurate implementation of the subsystems that constitute a walking or running machine simulator. These subsystems include the robot itself, the ground, and the actuators. A functional decomposition of a legged machine which shows the major subsystems is given in Figure 1.2 (on p.6), where the actuator and robot dynamics models are both a part of the robot block.

1.2 Scope of the Project

The objective of the project is not to develop a state-of-the-art humanoid simulator, but the focus will mainly be on understanding the dynamics governing a legged robot, actively controlling its motion, measuring its energy efficiency through accurate modeling of the major subsystems involved, and finally finding ways to maximize its energy economy. That is why we have adopted a number of simplifications as our assumptions to allow us to fulfill these tasks in the short time period of this project. These assumptions are:

1. Running, rather than walking, is the main focus of this thesis and most of the discussions are focused on a monopod hopper.
2. The analysis, design, and modeling are restricted to the planar case as opposed to the spatial case where extra degrees of freedom are present.
3. Simple controllers are considered to achieve the control objectives. These controllers are based on heuristics and are generally referred to as intuitive controllers due to their dependance on a person's intuition of a system.
4. Interactions between the foot and the ground are modeled as point contacts.
5. Ideal actuators were considered initially, and controller design was achieved using this assumption.
6. Later on in the project, these ideal actuators are substituted with models of electric DC motors.

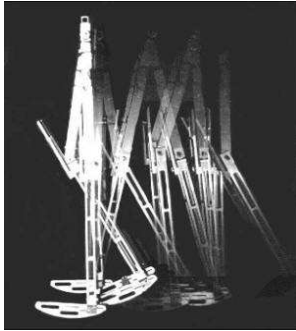
1.3 Author's Contributions

This project was initially proposed by my supervisor Dr. Roy Featherstone. As stated earlier, the goal of this project is to measure and optimize the energy efficiency of a special type of legged machines, i.e., running robots. Having a background in electrical engineering, I had to first educate myself on kinematics and dynamics of rigid body systems. Afterwards, I started developing simple articulated body structures and ground models. In the meantime, I was provided with a tentative Simulink model of a monopod hopper as shown in Figure 1.2, where the dynamics calculation routines in the robot block were supplied by Dr. Featherstone and are available as a MATLAB library [24]. My main challenges were to design the controller block, embed electrical actuators in the dynamics model, and to deploy alternative ground models that would be a closer implementation of the reality. Thus, the major contribution of this thesis has been the choice of accurate contact, friction, and actuator models, and the practical implementation of a hopping/running controller. Through these means, I was able to accurately measure the energy dissipation on different surfaces. All the simulations are done in MATLAB and Simulink, where the end result is a complete simulation package which is sufficient for calculating energy efficiency of a running machine. Because of the modular design of the blocks, the simulator can be modified with the least effort to migrate to other types of legged robots. In fact, we were able to easily migrate from a monopod hopper to a biped runner by making slight changes to the system. We predict that a migration from 2 dimensions to 3 dimensions is also straight forward by

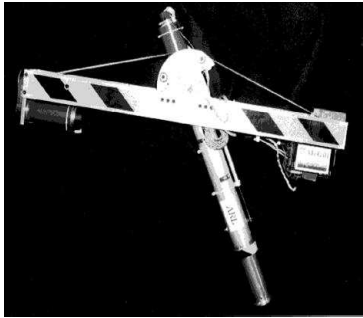
using the spatial version of the already available dynamics calculation routines and modifying the ground model to encompass three-dimensional forces. Apart from that, the use of masked Simulink blocks makes tweaking the parameters easy, because the user needs only to enter the parameter values for each subsystem, without needing to struggle with understanding the underlying structure. For control purposes, a modified version of Raibert's hopping algorithm was successfully implemented, through which the robot can achieve a desired hopping height and horizontal velocity. In addition, different types of contact and friction models are provided, so users can select different models, and then observe the resulting behaviors. Visualization of the dynamic behavior of the whole system is also done through two methods. A 2D visualization software for the planar hopper was developed by Dr. Featherstone which works by passing the joint positions and velocities of the robot to an m-file. Alongside that, I developed a 3D virtual reality model of the robot in VRML format (Virtual Reality Modeling Language). VRML is a standard file format for representing 3-dimensional vector graphics. By linking the VRML file to the robot model through Virtual Reality Toolbox in Simulink, I was able to produce a 3D visualization of the running robot and observe its behavior from different vantage points.

1.4 Organization of the Thesis

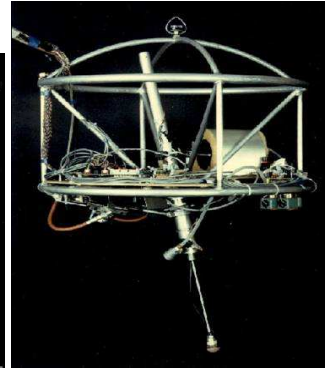
This thesis is organized into 6 chapters. The chapters follow the functional decomposition given in Figure 1.2. Each chapter, describes in more detail, one of the major subsystems that comprises a legged robot as a whole. Chapter 2 includes descriptions of the robot model itself including the monopod hopper and the bipedal runner. Chapter 3 gives a detailed explanation of the control algorithm used for stabilizing the motion of the robot. Chapter 4 talks about the type of the actuation including motors and transmission systems. Chapter 5 gives an extensive literature review of contact and friction models, and finally parameter values for the selected models are also provided. The results obtained from simulations along with an energetic analysis of the robot constitute the core part of Chapter 6. In the same chapter, some possible modifications and improvements are discussed and their validity is demonstrated through simulations. Finally, the last chapter (Chapter 7) gives conclusions and future work.



(a) McGeer Passive Walker



(b) ARL Monopod



(c) Raibert's Hopper



(d) Denise



(e) ASIMO



(f) Rabbit



(g) QRIO



(h) Big Dog

Figure 1.1: Pictures of some of the famous legged robots

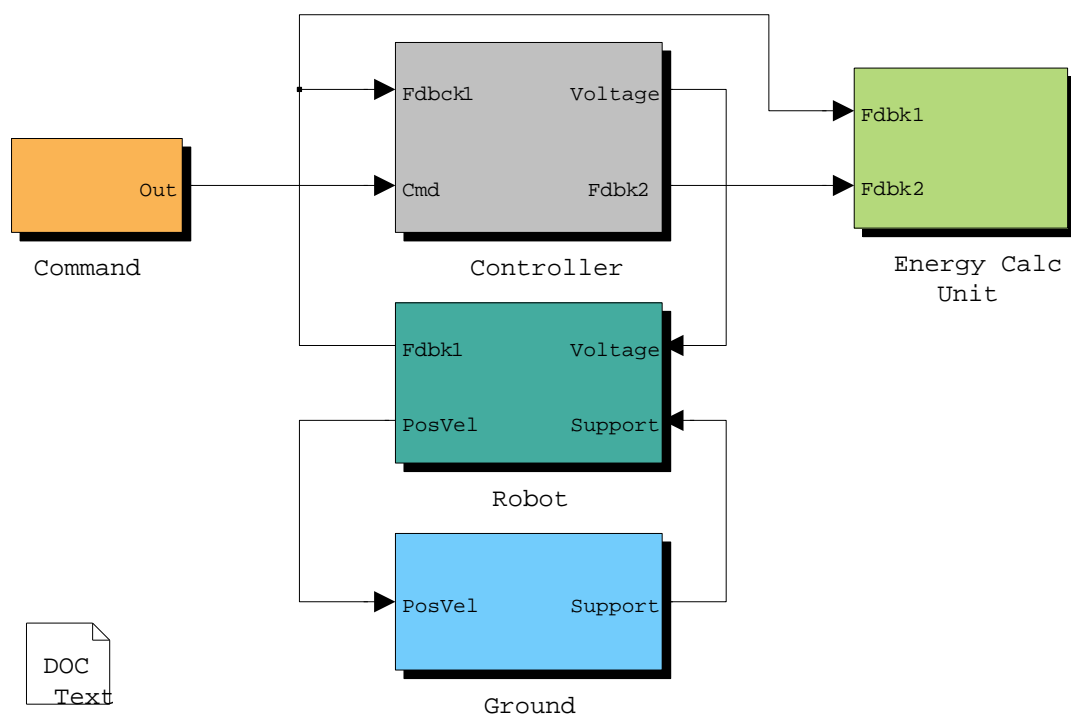


Figure 1.2: Functional decomposition of a hopping/running robot

Chapter 2

Robot Model

In this chapter, we present the structure and specifications of the robots that were simulated in the project. We also explain the scheme used for modeling these robots as articulated rigid-body systems in the plane. A detailed description of the theory of rigid-body kinematics and dynamics is beyond the scope of this thesis, and the dynamics of the robot mechanism is only touched upon briefly here. For a better understanding robot dynamics, the interested reader is encouraged to read the dynamics book of Featherstone [23].

2.1 Dynamics Modeling of the Robot Structures

A legged robot is usually modeled as a kinematic tree (i.e., a rigid-body system having the connectivity of a tree). Generally, there are two ways of modeling rigid-body systems in a computer. In the first approach, the equations of motion for a given system are manually derived and are later translated into computer code. The second approach, the one used in this thesis, is to construct a system model describing a given configuration and feed it as an argument to a model-based dynamics calculation routine. This approach is more desirable since it is easier to construct, modify, and use a large number of times. Indeed, using this approach made it possible for us to easily migrate from a monopod hopper to a bipedal runner with only slight modifications.

From the degrees of motion freedom point of view, analysis of rigid-body systems can be done in two ways: planar (i.e., assuming the motion of the system is restricted within the plane) or spatial (i.e., assuming the motion is in the whole space). Several representations for either of these cases exist. Craig [15] mentions and describes representations for spatial dynamics using 3D vectors and quaternions (4D vectors), while Featherstone [23] uses a more efficient vector representation referred to as spatial vectors. Using spatial vector representation, dynamics equations can be succinctly expressed in symbolic form. Moreover, a quicker derivation of the dynamics algorithms is made possible, and conversion to com-

puter code is also easier to achieve. Planar analysis is a special case of spatial analysis where additional constraints are present. In this thesis, the dynamics analysis is restricted to the plane, where the planar modeling scheme and calculation routines from [23] are specifically used.

The dynamics calculation software used was previously developed by Featherstone in MATLAB [24]. After developing a system model for the robot, the model is fed to the forward dynamics algorithm as an argument along with the external forces, joint positions and velocities, all described in generalized coordinates. Perhaps the most important equation in deriving the equations of motion and modeling the dynamic behavior of a robot is:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}), \quad (2.1)$$

where \mathbf{q} , $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ are vectors of joint position, velocity, and acceleration variables respectively. $\boldsymbol{\tau}$ is a vector of applied forces, $\mathbf{H}(\mathbf{q})$ is a matrix of inertia terms, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is a vector of force terms that account for the coriolis and centrifugal forces, gravity, and any other forces acting on the system other than those in $\boldsymbol{\tau}$. Most of the algorithms used for calculating the forward and inverse dynamics are a result of this equation. The forward dynamics algorithm which was specifically used, is based on the articulated-body algorithm in [23]. The order of complexity of this algorithm, which calculates the forward dynamics of a kinematic tree, is $O(N_B)$, where N_B is the number of the bodies in the tree.

2.1.1 Monopod Hopper

Figure 2.1 shows the physical structure of a 5 DOF monopod hopper simulated in this project. It is comprised of three rigid bodies, which we call the body, the hip and the leg. The body is nominally a beam with weights at each end. The hip and the leg are nominally cylinders. The body is connected to the hip through a revolute joint whereas the hip is connected to the leg through a prismatic joint. This configuration is very similar to Raibert's monopod hopper [56], in which he uses pneumatic actuators for actuating the hip and leg joints, and where the springiness of the leg is achieved through a control mechanism which regulates the air pressure using solenoid valves in the leg cylinder. Table 2.1 shows the physical parameter values representing this structure. The given values do not correspond to a real robot but have been chosen to resemble a lighter version of Raibert's hopper. From Table 2.1, it can be seen that the body has been assigned a high moment of inertia compared to the leg. As will be seen, this allows the robot to servo the hip in the flight phase for foot placement without much change in the body attitude. The same parameter values and body mass assumptions can be used to represent the structure of a hopper with electric motors as actuators. Figure 2.3 shows such a hopper structure, which we call *configuration 1* throughout the thesis. This structure is similar to the ARL monopod [1] in which DC servo motors,

reduction drives, and a springy leg are used. The leg can be considered as two parts: the sprung part, which is the upper part of the leg, and the unsprung part, which is the lower part of the leg (i.e., the part below the spring). In our simulations, the whole leg constitutes the unsprung section. Many other hopping robot structures exist in the literature. Sayyad et al. [61] give a comprehensive review of developments in the field of single-legged hopping robots.

Monopod Hopper System Model

The first step in deriving a system model for an articulated body is to draw its connectivity graph. The procedure of drawing the connectivity graph and obtaining a system model is from [23]. The monopod hopper discussed in the previous section can be modeled as a 5 DOF robot comprising of $n = 6$ rigid bodies, with its connectivity graph having 6 links (B_0 to B_5) and 5 joints (J_1 to J_5) as shown in Figure 2.2. In this figure, the first link serves as the fixed base, the second and third links are imaginary links, and the fourth, the fifth, and the sixth links represent the body, the hip, and the leg of the robot respectively. As for the joints, the first two account for the translation of the body in the x and y directions, while the third stands for the rotation of the body around the z axis. The fourth and the fifth links represent the actuated joints of the hopper, namely, the hip and the leg joints. The hip joint is a revolute joint around the z axis, while the leg joint is a prismatic joint in the y direction.

Operation of the Monopod Hopper

To start a hopping simulation, the hopper is dropped with an initial height and an initial horizontal velocity with the leg spring at its normal length. We define *robot height* as the y coordinate of the body's center of mass with respect to the reference frame (as shown in Figure 2.1). Due to gravity, the hopper accelerates towards the ground. The moment the foot makes contact with the ground is referred to as *touch down*. The leg spring compresses until the body is completely stopped where the leg spring starts its decompression phase. This moment is referred to as *bottom*. Then the body starts to accelerate in the opposite direction until the moment the foot loses contact with the ground, which is referred to as *lift-off*. The body continues its upward movement before its velocity reaches zero and one cycle of hopping completes. This moment of maximum altitude is referred to as *top*. In each hopping cycle, a fraction of energy is dissipated for accelerating the unsprung part of the leg which loses most of its energy when hitting the ground. Hence the mass of the unsprung part of the leg should be low compared to the total mass of the system to reduce energy losses. Other losses are due to friction in the moving parts of the system. To maintain the hopping height, the leg motor starts compressing the leg spring during stance to inject extra

energy into the system. To continue its forward motion, the hip motor servos the hip for foot placement during the flight phase.

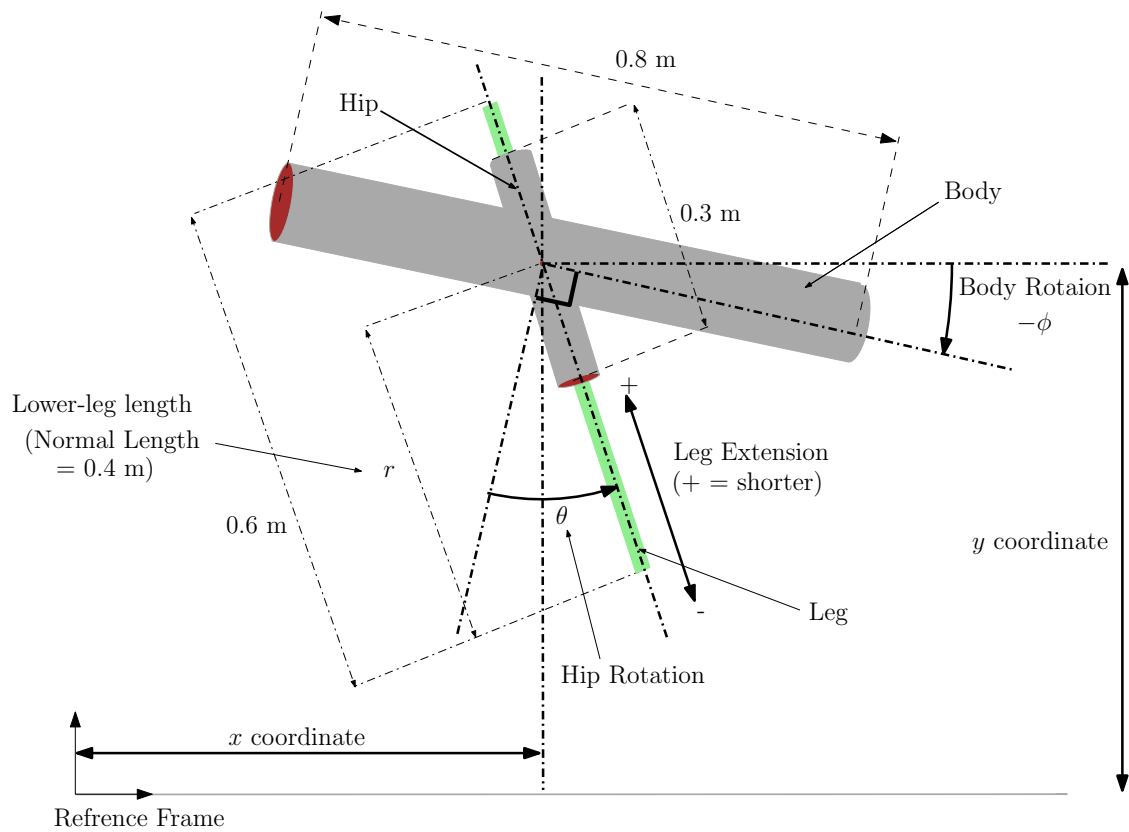


Figure 2.1: Graphical model of the 5 DOF monopod

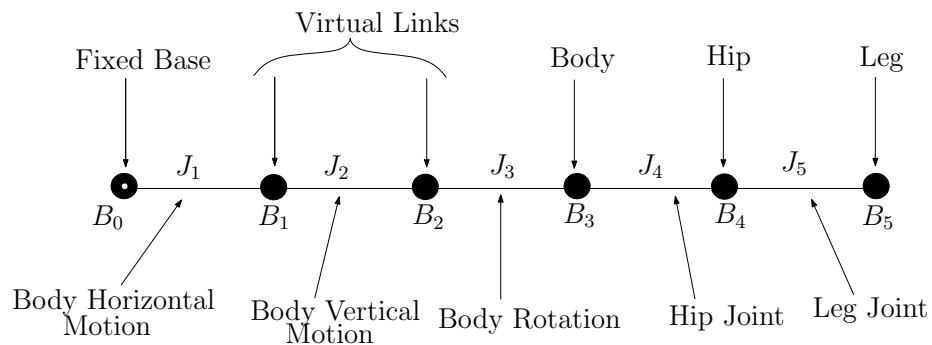


Figure 2.2: Connectivity graph of the 5 DOF monopod

Parameter	Values (SI Units)
Body width	0.8 m
Hip length	0.3 m
Leg length	0.6 m
Body mass	5 .0Kg
Hip mass	0.5 Kg
Leg mass	0.5 Kg
Body moment of inertia	0.6 Kg.m ²
Hip moment of inertia	0.004 Kg.m ²
Leg moment of inertia	0.05 Kg.m ²
Body, leg, and hip center of mass	Center Points

Table 2.1: Physical parameters of the planar one-legged hopper of Figure 2.1

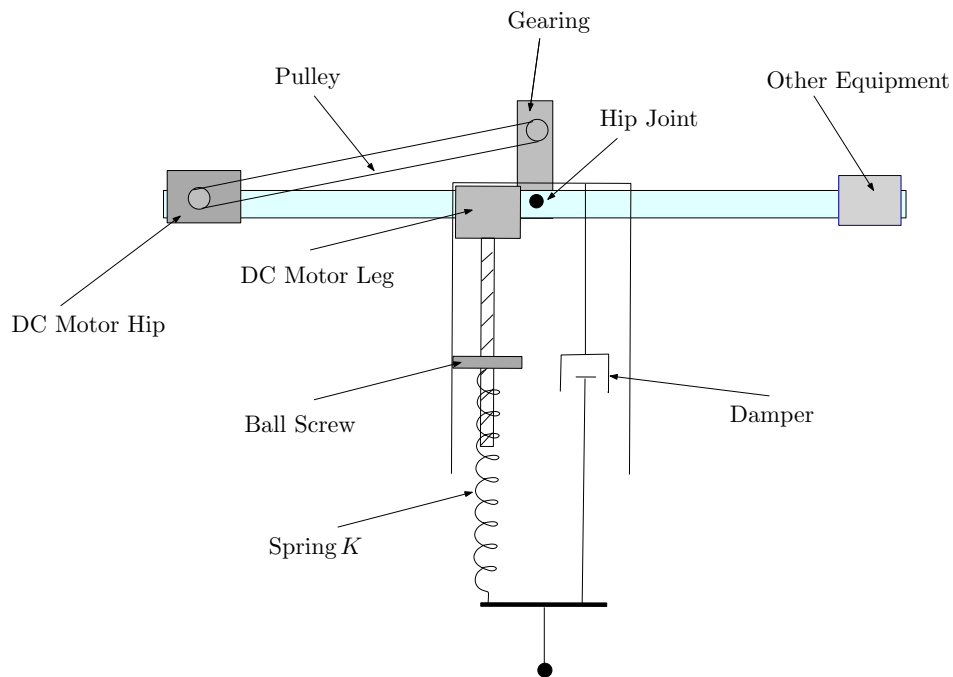


Figure 2.3: Monopod hopper with DC motors as actuators (Leg configuration 1)

2.1.2 Bipedal Runner

Earlier in this chapter, it was mentioned that since we are using a model-based dynamics calculation routine, migration to a bipedal runner can be achieved simply by developing

a system model for the bipedal robot structure. A simple graphical model for the bipedal runner structure used in our project is given in Figure 2.4. As can be seen, it is a 7 DOF rigid-body system comprising of 5 bodies which we call the main body, the two hips, which are each connected to the body by a revolute joint, and the two legs, one connected to each hip. This bipedal runner structure is identical to the monopod hopper structure, except that an extra hip-leg pair has been added. The connectivity graph of this system is given in Figure 2.5.

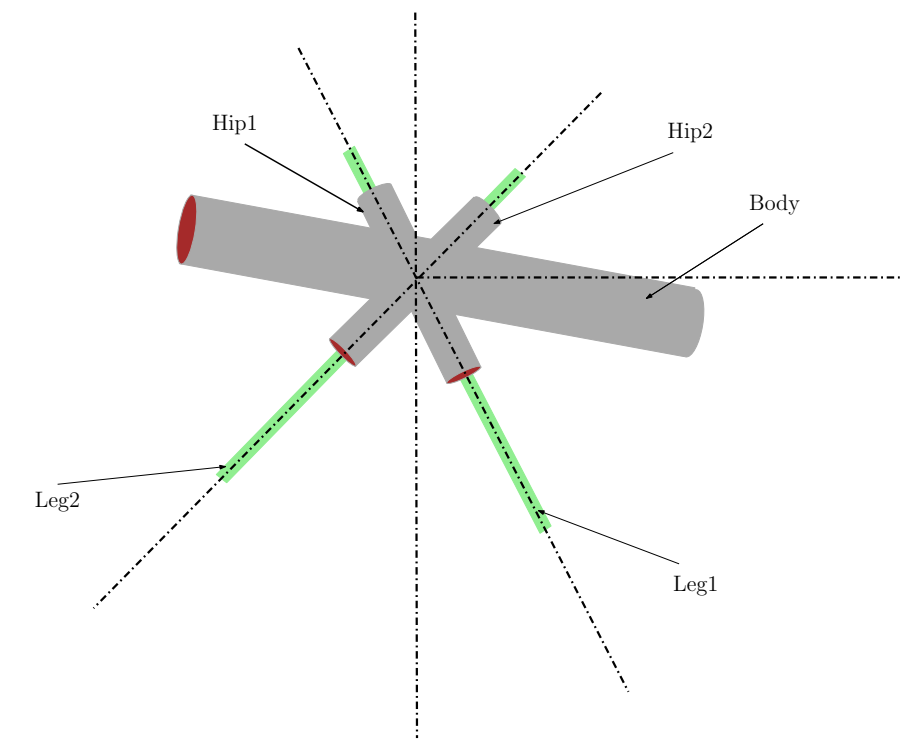


Figure 2.4: Graphical model of the 7 DOF biped

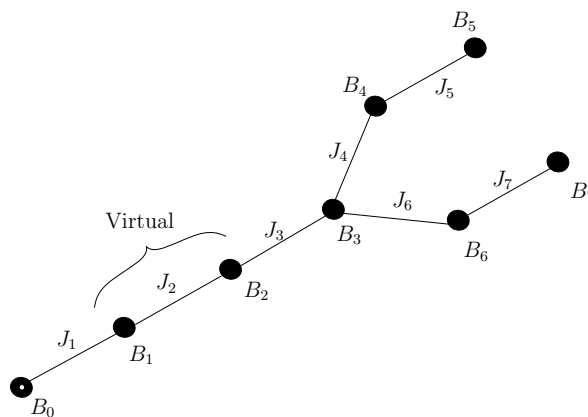


Figure 2.5: Connectivity graph of the 7 DOF biped

2.2 Foot Model

Point feet are used instead of rolling feet, mainly due to the inherent complexity in modeling rolling feet. It is also assumed that while in contact, the foot is supported by a single point on the ground. An example of the complexities in modeling of rolling feet is shown in the two pictures of Figure 2.6. The picture on the top shows an undistorted surface, where the points P_1 and P_2 are located close to each other. The bottom picture shows the same surface after a rolling foot has rolled over it. As a result of rolling, points P_1 and P_2 have changed position and are now marked as P'_1 and P'_2 . By comparing these two figures, it is obvious that the distance between the two points has changed and is not the same, i.e.,

$$|P'_2 - P'_1| \neq |P_2 - P_1|.$$

In fact, parts of the surface which are closer to the center of contact are more severely compressed than the parts which are further away from it. In addition to this cause, in a rolling contact model, the foot is supported by multiple contact points and the forces due to all of these contact points should be taken into account. These two reasons clearly show two of the complexities that exist in modeling rolling feet. That is why, although point feet are not real representations of human or animal feet, to avoid the complexity in modeling rolling feet and to focus on control and energetics of hopping and running, point feet are used in our simulations.

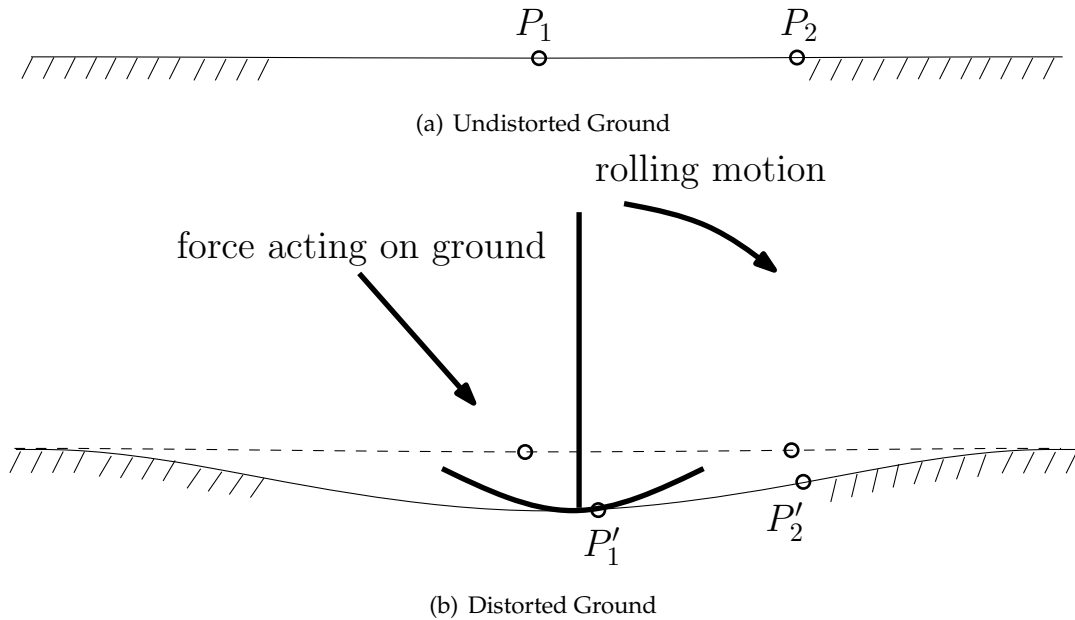


Figure 2.6: Rolling foot example

Chapter 3

Control System

The control system which is discussed here and was used in our project is a modified version of the monopod hopper controller proposed by Raibert [54, 55]. In what follows, a detailed description of this controller is given. Because of the high amount of similarity to the monopod hopper, only a very short discussion on the control of the bipedal runner is given towards the end of the chapter.

3.1 Monopod Hopper

Running resembles the bouncing of a ball in the sense that it consists of two phases: ballistic flight and elastic collision. In contrast with a bouncing ball, where energy is dissipated in each collision phase, a running robot can maintain its energy using actuators. Moreover, a running robot can tip like an inverted pendulum, hence it needs to actively balance by moving its feet in response to tipping motions, consequently highlighting the need for control.

Balancing a hopper through mathematical analysis is perceived to be a difficult task, however, simple control strategies based on intuition can easily be utilized to achieve the control objectives required for hopping and running. Well-known examples of intuitive controllers are the so called PD controllers, which are extensively used in robotics applications. Intuitive PD controllers were also used by Raibert [54] for balancing his monopod hopper, where he decomposes the control task into three parts:

- Control of hopping height
- Control of forward speed
- Control of body attitude

Control of hopping height is achieved by specifying the thrust to be delivered by the leg on each hop, whereas control of forward speed is achieved by extending the foot forward

to a position that will provide the needed acceleration during stance, and finally control of body attitude is achieved by actuating the hip during the stance phase. To synchronize these three parts, a state-machine is used. Indeed, this synchronization is possible since these three control tasks are known to be largely decoupled. Figure 3.1 shows a diagram of the different states which occur during a hopping cycle. Table 3.1 lists the actions that need to be taken in each state, along with a description of these states. In a real robot, these states are obtained from the sensor data on the robot structure, whereas in our simulation model, they are detected from feedback signals from the robot and ground models. In the next section, a detailed analysis of Raibert's three part controller adapted for our monopod is presented.

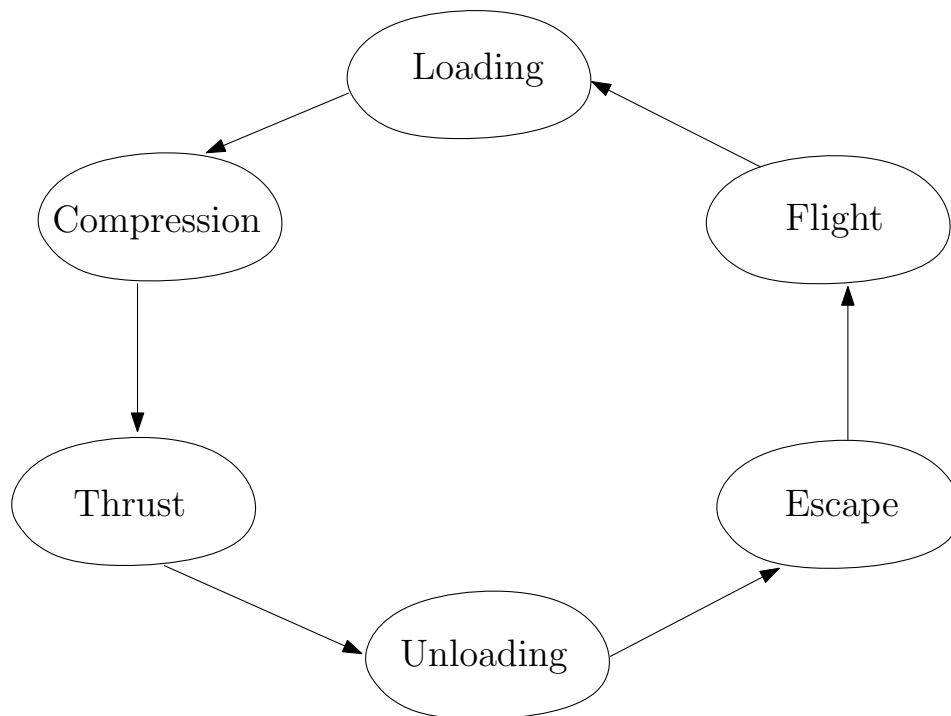


Figure 3.1: Hopping states used in the State-Machine

Control of Hopping Height

The basic hopping motion of the monopod shown in Figure 2.3 is forced passively by the robot dynamics, whereas the thrust delivered to the body during each hop determines the hopping height. This thrust is provided from the potential energy that is stored in the leg spring when the robot reaches its bottom. However, due to energy dissipation, extra energy needs to be injected into the system if the robot is to maintain its hopping height or if it is to achieve a greater height. This extra energy is provided by compressing the spring which is

State	Description	Action
Loading	the few mili-secs after foot makes contact	servo leg motor to resist back-driving
Compression	leg spring compresses	servo hip motor for body attitude servo leg motor to resist back-driving
Thrust	the period from maximum leg compression until the required energy is injected	servo leg motor to to compress spring servo hip motor for body attitude
Unloading	leg spring reaches its natural length	stop servoing the hip motor
Escape	no contact tip of foot height $< d_{\text{escape}}$ ¹	servo leg motor to zero position
Flight	no contact tip of foot height $> d_{\text{escape}}$	servo hip motor for foot placement servo leg motor to act as a damper
¹ d_{escape} :	is the minimum required distance between the tip of the leg and the ground before which the leg will collide with the ground if swung forward for foot placement	

Table 3.1: State-Machine events and actions

connected between the leg and the actuator. The length of this compression can be found by comparing the energy needed to reach a desired height with the actual energy of the robot. Another approach for finding the compression length is to use a large look-up table which stores the required spring compression length from one hopping height to another [56]. Yet, a simpler method for controlling the hopping height is to compare the maximum hopping height achieved in the previous hopping cycle with the desired hopping height at the current cycle and set the compression length proportional to their difference. This is a proportional control law and is given as:

$$\Delta l = \Delta l_{\text{old}} + k_p (h_d - h_{\text{max}}), \quad (3.1)$$

where Δl is the spring compression length, k_p is the proportional gain, h_{max} and Δl_{old} are the maximum height of the robot and the compression length of the spring at the previous cycle, and h_d is the desired height for the robot.

Other methods for minimizing the hopping height error exist in the literature, e.g., Helferty et al. [32] present an adaptive control regime based on artificial neural networks for minimizing hopping height error.

Control of Forward Speed

The forward speed can be regulated by determining the position of the foot before each landing. This is done by actuating the hip joint during the flight phase. Raibert observed that for each forward speed, there is a unique foot position that results in zero net forward acceleration, called the neutral point x_{f0} (Figure 3.2), and that the net forward acceleration varies with forward foot position. Placing the foot forward of the neutral point, the machine decelerates, and placing it behind the neutral point results in a forward acceleration. To achieve a reference speed, the control system first approximates the location of the neutral point based on the current forward speed and then places the foot behind or in front of the neutral point depending on whether it wants to accelerate or decelerate. An important parameter for calculating the neutral point is the stance period. For a given robot mass and leg spring stiffness, the stance period is constant regardless of the forward speed, considering that a spring mass system oscillates with a period that is independent of amplitude. The desired foot angle before touch down is calculated from the following formula:

$$\theta_d = \phi - \sin^{-1}\left(\frac{\dot{x}T_s}{2r} + \frac{k_{\dot{x}}(\dot{x} - \dot{x}_d)}{r}\right), \quad (3.2)$$

where $k_{\dot{x}}$ is the velocity gain, \dot{x} is the velocity of the robot in the horizontal direction, r is the lower leg length (as shown in Figure 2.1, p. 10), T_s is the stance time, and ϕ is the orientation of the body of the robot with respect to the reference frame. After finding θ_d , a PD controller can be used to servo the hip to this desired angle based on the following control law:

$$\tau = -k_p(\theta - \theta_d) - k_v(\dot{\theta}), \quad (3.3)$$

where τ is the calculated torque, k_p is the proportional gain, k_v is the derivative gain, and θ is the hip angle. The effectiveness of this control algorithm is guaranteed so long as the ratio of the body moment of inertia to the leg moment of inertia is high, ensuring that the orientation of the leg can change during flight without severely disturbing the attitude of the body.

This control algorithm results in a steady-state velocity error that depends on the forward velocity and the parameters of the model [55]. To remedy this problem, Sznaiier and Damborge [65] proposed two control laws, one for the height controller which reduces the coupling between the horizontal and vertical motions, and one for the horizontal controller which completely removes the forward velocity error adaptively even in situations where changes in nominal robot conditions are present (e.g., changes in the robot mass due to different payloads).

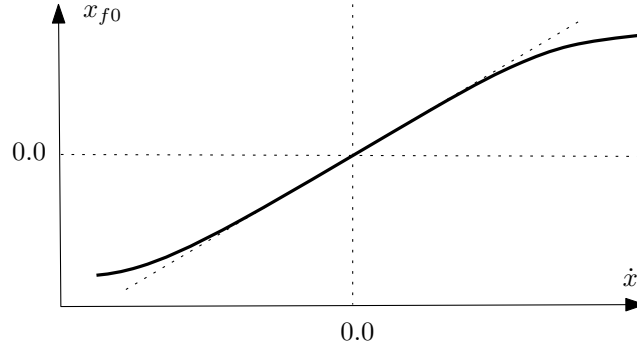


Figure 3.2: Neutral position vs. Forward speed

Control of Body Attitude

The task of the third part of the control system is to maintain an upright body attitude by exerting torques about the hip during stance. In fact, servoing the hip during the stance is possible because of the existing friction between the foot and the ground which prevents the foot from slipping. However, care should be taken in the timing in which torques are applied because the robot can always slip if the ground normal force, and as a consequence the friction on the leg tip is not high enough. The states *Loading* and *Unloading* in Figure 3.1 are present for this reason. After taking these considerations into account the control law is:

$$\tau = -k_p(\phi - \phi_d) - k_v(\dot{\phi}), \quad (3.4)$$

where k_p is the proportional gain, k_v is the derivative gain, ϕ is the body angle, and ϕ_d is the desired body angle. It is worth mentioning that the stance phase is the only time that the robot can change its angular momentum and this is basically achieved through changing the body attitude by actuating the hip. Acrobatic behavior from the hopper can be achieved through controlling the body attitude during the stance phase.

3.2 Bipedal Runner

The control system designed for the bipedal runner is very similar to the one used for the monopod hopper. The main different in controlling the bipedal runner is in the coordination of the legs. This is achieved by dividing the entire running cycle into two phases where each phase corresponds to the one leg that is acting as the stance support. This technique is referred to as leg sequencing. It is also possible to add a concept called virtual leg and apply the same control technique to a quadruped [53].

Chapter 4

Actuation System

Actuators are the motors responsible for motion in a robot. The type of actuators used in robotic applications range from the more powerful hydraulic and pneumatic, to the less powerful electric, piezoelectric, ultrasonic, and so on. In general, the criteria for selecting a suitable actuator for a specific robotics application are the required power-to-weight ratio, cost, efficiency, cleanliness, controllability and availability in the market. According to these criteria, hydraulic, pneumatic, and electric actuators are the most widely used in legged robot applications.

Hydraulic and pneumatic actuators were historically used in Raibert's hoppers [54]. They have the advantage of having high power-to-weight ratios compared with those electric motors that are suitable for mobile robot applications. Using these powerful actuators particularly helped Raibert to focus on design and control problems without having to worry about actuation constraints. However, these actuators have some disadvantages which have made electric motors the attractive choice in the more recent mobile robot applications. As a comparison, electric motors are better off in terms of being a clean, safe and cheap technology, suitable for indoor use and autonomous robots. Moreover they can be easily integrated in a robot structure and are better steerable.

Biological robotic joints require high torques at low velocities whereas small electric motors that are suitable to be used in legged robot applications (e.g., in the joints of a monopod hopper) are basically torque limited and have a very low efficiency at low rotor speeds (refer to Figure 6.8, p. 58). To match the high-torque low-velocity requirements of biological robotic joints, a speed reduction system is usually used along with the electric motors. The benefit of a speed reduction system is two folds. Firstly, it increases the output torque of the motor, and secondly, it allows the motor to reach higher speeds where maximum power can be delivered. Nevertheless, one must take into account that adding a transmission system to a motor has some drawbacks, including the introduction of additional dynamics to the system, waste of energy for merely accelerating gear inertia, and high driving torque losses

attributed to high gear ratio mechanisms. Hence, care should be taken in the design and selection of these parts.

For the hopping machine of Figure 2.3, the hardest challenge is to maximize energy injection at maximum leg compression. To maximize energy efficiency and make use of electric motor speed-torque characteristic, Papantoniou [50] deploys a clutch which engages the spinning motor at maximum leg compression. However, this technique requires a considerably large stance time which is not always practical. Hence, the best way to maximize energy injection at the short stance period is to find the best motor and transmission combination suitable for this task.

Towards this goal, in this chapter, a detailed mathematical modeling for two types of electric motors, i.e., the brushed DC motor and the three phase brushless DC motor is presented. Thereafter, a brief description of some transmission systems is given. In the end of the chapter, the procedure of selecting a motor and transmission combination suitable for the leg of our monopod hopper is described.

4.1 DC Motors

DC motors are types of electric motors which are powered by direct current electricity which is available in small packages, as opposed to AC motors which are powered by alternating current electricity which is not easily at hand. This makes DC motors suitable for mobile robot applications. Similar to AC motors, DC motors have AC voltages and currents flowing in them, however, they can work with a DC power source because of a mechanism called commutation. According to the type of commutation used, DC motors fall into two main categories: the mechanically commutated ones generally referred to as brushed DC motors, and the electronically commutated ones known as brushless DC motors. In this section, a detailed mathematical modeling of these two types of DC motors is presented. Precise mathematical modeling is necessary as a starting point for accurate dynamics simulations, control, and energy calculation purposes.

4.1.1 Brushed DC Motor

In a brushed DC motor the commutation is done mechanically through brushes. This type of motor in turn comes in different types: the separately excited, the shunt, the series, and the permanent-magnet DC motor. A permanent-magnet DC motor (PMDC) is a type of DC motor whose poles are made of permanent magnets as opposed to the other types in which field windings are deployed. Because no field windings are used, a PMDC has a smaller volume and horsepower size compared to the other types and is the most widely used in mobile robot applications [12].

Control of the brushed DC motor

Depending on the application, different variables can be controlled in a DC motor, ranging from velocity and angle to the generated torque. In many robotics applications, it is often desirable to control the torque generated by the motor. This is done by controlling the armature current through voltage source v_a (as will be seen in Equation (4.1)). However, there are generally two restrictions in this process. Firstly, the inductance of the motor limits the rate of the change in the armature current and secondly there is an upper bound on the voltage capability of the voltage source v_a . The net effect is that of a low-pass filter between the requested current and the output torque. Nevertheless, in most applications we can ignore the armature inductance since the natural frequency of the closed-loop control system is quite low compared to the cut-off frequency of the implicit low-pass filter in the current-drive circuitry¹ [15].

Controlling the speed or position instead of the generated torque is slightly more complicated. First, a state-space representation of the system (or a transfer function) relating the input voltage v_a to the velocity of the rotor $\dot{\theta}_m$ is derived. Thereafter, any control strategy such as PID control or state-space control can be applied to achieve the desired response. A number of control strategies applied to this control problem in MATLAB and Simulink can be found in [68]. The mathematical modeling of the brushed DC motor in state-space is presented next.

Mathematical modeling of the brushed DC motor

For deriving a mathematical model of the brushed DC motor, we start with the two equations governing the physics of a DC motor. The first one is:

$$\tau_m = K_m i_a, \quad (4.1)$$

which relates the induced torque developed in a motor to the armature current by means of motor torque constant K_m , and the second one is:

$$e_a = K_e \dot{\theta}_m, \quad (4.2)$$

which relates the generated voltage in a motor to the rotating speed of the armature by means of back EMF constant. Basically, the torque constant and the back EMF constant have the same value. The only difference between these two parameters is the units used for presenting them. If the torque constant units is specified as $\frac{\text{Nm}}{\text{A}}$, the back EMF constant will be specified as $\frac{\text{Vsec}}{\text{rad}}$. From now on we will simply use K for both of these constants.

¹Another simplifying assumption we make here is ignoring the torque ripple which is an effect caused by the switching of the current due to commutation.

The equivalent circuit of the armature winding of a DC motor is given in Figure 4.1, where v_a is the voltage source, and L_a and R_a are the inductance and the resistance of the armature winding respectively. The differential equation governing this equivalent circuit is:

$$L_a \dot{i}_a + R_a i_a = v_a - K \dot{\theta}_m. \quad (4.3)$$

In Equation (4.3), $\dot{\theta}_m$ can be obtained by the time integration of $\ddot{\theta}_m$, which is in turn obtained from the following equation:

$$\ddot{\theta}_m = \frac{\tau_m - \tau_c - b_m \dot{\theta}_m}{I_m}, \quad (4.4)$$

where τ_c is the counter torque acting on the rotor, b_m is the viscous damping of the rotor, and I_m is the rotor inertia.

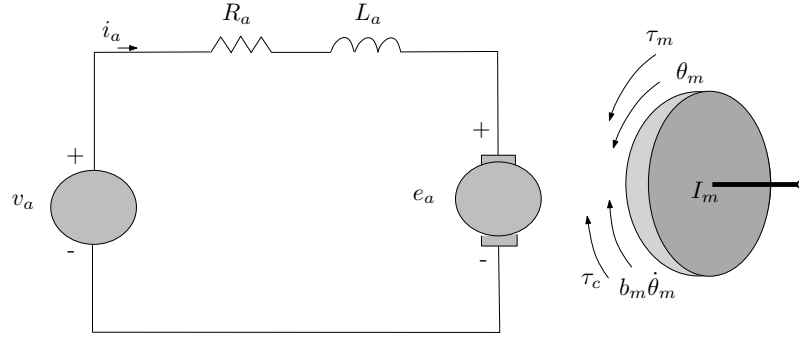


Figure 4.1: Brushed DC motor equivalent circuit and free-body diagram

From Equations (4.3) and (4.4) the state-space equations can be obtained as:

$$\dot{X} = AX + BU, \quad (4.5)$$

where

$$X = \begin{pmatrix} i_a \\ \omega_m \end{pmatrix}, \quad A = \begin{pmatrix} -\frac{R_a}{L_a} & -\frac{K}{L_a} \\ \frac{K}{I_m} & -\frac{b_m}{I_m} \end{pmatrix}, \quad B = \begin{pmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{I_m} \end{pmatrix}, \quad U = \begin{pmatrix} v_a \\ \tau_c \end{pmatrix}.$$

4.1.2 Brushless DC Motor (BLDCM)

A Brushless DC motor [71] is a synchronous motor which is electronically commutated as opposed to using brushes for commutation. A brushless DC motor has several advantages over a brushed DC motor, among them are better speed versus torque characteristics, higher dynamic response, higher efficiency, longer operating life, lesser noise due to operation, and higher speed ranges. It also has a better power to weight/volume ratio and is better suited for applications where weight and volume are of the essence. A BLDC comes in three different types: the 1-phase, the 2-phase, and the 3-phase BLDC, where the 3-phase BLDC

is the most widely used one. Because of its popularity, only the 3-phase BLDC is considered here.

Generally there are two types of 3-phase BLDCs depending on the structure of their stator windings. The two winding configurations result in two different back EMFs. These two different types are generally referred to as the trapezoidal and the sinusoidal back EMF forces. In this section, a mathematical model in state-space form for a 3-phase BLDC with a trapezoidal back EMF is given [51].

Mathematical modeling of the brushless DC motor

A typical BLDCM has three windings in the stator and one permanent magnet on the rotor. The equivalent circuit for each of these phases is similar to a brushed DC motor. Hence, the circuit equations of the three windings in phase variables are [40, 37]:

$$\begin{pmatrix} v_a \\ v_b \\ v_c \end{pmatrix} = \begin{pmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{pmatrix} \begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix} + \begin{pmatrix} L & M & M \\ M & L & M \\ M & M & L \end{pmatrix} \frac{d}{dt} \begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix} + \begin{pmatrix} e_a \\ e_b \\ e_c \end{pmatrix}, \quad (4.6)$$

where it is assumed that the stator resistance in all the windings are equal and there is no change in the rotor reluctance due to change in rotor angle. In this equation v_a , v_b , and v_c are the voltage sources connected to each winding, e_a , e_b , and e_c are the back EMFs in the armature windings, L is their inductance, M is the mutual inductance between them, and R is their resistance. One extra equation governing the 3 phase BLDCM's equivalent circuit is:

$$i_a + i_b + i_c = 0, \quad (4.7)$$

from which we obtain:

$$Mi_b + Mi_c = -Mi_a. \quad (4.8)$$

Hence, Equation 4.6 can be rewritten as:

$$\begin{pmatrix} v_a \\ v_b \\ v_c \end{pmatrix} = \begin{pmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{pmatrix} \begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix} + \begin{pmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{pmatrix} \frac{d}{dt} \begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix} + \begin{pmatrix} e_a \\ e_b \\ e_c \end{pmatrix}. \quad (4.9)$$

On the other hand, the generated torque by the motor is:

$$\tau_m = (e_a i_a + e_b i_b + e_c i_c) / w_m, \quad (4.10)$$

where w_m is the rotor speed. In this equation, e_a , e_b , and e_c can in turn get substituted by

$$\begin{aligned} e_a &= f_a(\theta_m) \lambda w_m, \\ e_b &= f_b(\theta_m) \lambda w_m, \\ e_c &= f_c(\theta_m) \lambda w_m, \end{aligned} \quad (4.11)$$

where $f_a(\theta_m)$, $f_b(\theta_m)$, and $f_c(\theta_m)$ are functions having the same shapes as the back EMFs, with maximum magnitude of 1, and λ is a constant representing the total flux linkage. The equation of motion is:

$$\frac{d}{dt}w_m = (\tau_m - \tau_c - b_m w_m)/I_m, \quad (4.12)$$

where τ_c is the counter torque acting on the rotor, b_m is the viscous damping of the rotor, and I_m is the rotor inertia. Moreover, electrical rotor speed and position are related by

$$\frac{d\theta_m}{dt} = (P/2) w_m, \quad (4.13)$$

where P is the the number of poles in the motor. Finally, combining these equations and assigning $X = [i_a \ i_b \ i_c \ \omega \ \theta]$ as the state of the system, the state-space equations will be of the form $\dot{X} = AX + BU$, where

$$A = \begin{pmatrix} -R/L' & 0 & 0 & -(f_a(\theta)\lambda)/L' & 0 \\ 0 & -R/L' & 0 & -(f_b(\theta)\lambda)/L' & 0 \\ 0 & 0 & -R/L' & -(f_c(\theta)\lambda)/L' & 0 \\ (f_a(\theta)\lambda)/I_m & (f_b(\theta)\lambda)/I_m & (f_c(\theta)\lambda)/I_m & -b_m/I_m & 0 \\ 0 & 0 & 0 & P/2 & 0 \end{pmatrix},$$

$$B = \begin{pmatrix} 1/L' & 0 & 0 & 0 \\ 0 & 1/L' & 0 & 0 \\ 0 & 0 & 1/L' & 0 \\ 0 & 0 & 0 & -1/I_m \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad U = \begin{pmatrix} v_a \\ v_b \\ v_c \\ \tau_c \end{pmatrix},$$

and where $L' = L - M$.

Control of the brushless DC motor

Compared to a brushed DC motor where a comparatively simple controller such as a rheostat can be used for regulating its speed, BLDCs require complex electronic speed controllers to run. For the interested reader, a complete Simulink model of the brushless DC motor discussed in this section along with its controller can be found in [18].

4.2 Transmission and Reduction System

In general, many actuators, including DC motors, are better suited for low torque/high speed applications and cannot always be used in a direct drive configuration. Hence, they are usually accompanied by a speed-reduction mechanism. In some situations where the location of the motor is important, a transmission system is also used. An example is the

ARL monopod [1], where to make the inertia of the body high compared to that of the leg, the hip actuator is placed at one end of the body, while power is transmitted to the hip through pulleys and cables.

Gears

The most common elements used for speed reduction are gears, usually characterized by their gear ratio, η . The relationship between input and output speeds and torques are given by

$$\begin{aligned}\dot{\theta}_o &= (1/\eta)\dot{\theta}_i \\ \tau_o &= \eta\tau_i\end{aligned}\tag{4.14}$$

where $\dot{\theta}_o$ and $\dot{\theta}_i$ are the output and the input speeds, and τ_o and τ_i are the output and the input torques respectively. However, as mentioned earlier, adding a gear reduction mechanism has consequences [52]. Firstly, it adds additional dynamics to the system and makes the dynamics equations more complicated. Secondly, a significant amount of energy is dissipated for merely accelerating the gear inertia, and lastly, high gear ratio mechanisms are notorious for their significant torque losses.

Harmonic Drives

A harmonic drive is a gearing mechanism that has several advantages over the conventional gear trains. Extremely high reduction ratios in a very compact and lightweight package, zero backlash, consistent performance, high positional accuracy, high torque to weight ratio, and cost effectiveness, are all reasons to use a harmonic drive as opposed to the conventional gear trains [44].

Ball Screws

A ball screw is a highly efficient speed reduction system for converting rotary to linear motion compared with a lead screw which is a less costly, lower performance device. Ball screws are generally characterized by their lead, L . Ball screw lead is the linear distance the end screw travels during one revolution of the ball screw.

Calculating the torque requirements of a ball screw is divided into two phases: the driving phase and the back-driving phase [34]. An explanation of these phases with the associated formulas for calculating the torque requirements is given next.

Driving torque

Driving torque is the amount of torque required by the ball screw to move a load. It can be calculated from the following formula:

$$\tau_d = \frac{F_L \times l}{2\pi e} = 0.177 F_L \times l \quad (4.15)$$

where

τ_d = Drive torque

F_L = Load

l = Screw lead

e = Ball screw efficiency (90%)

Usually a torque to raise one pound or one kN of load is provided in the ball screw technical data sheets by the manufacturer, and the required torque can easily be calculated as a multiple of this value.

Backdriving torque

Due to the efficiency of a ball screw, a load applied to the ball nut will generate back-driving torque on the ball screw. The torque required to hold the load in position can be calculated by the following formula:

$$\tau_b = \frac{F_L \times l \times e}{2\pi} = 0.143 F_L \times l \quad (4.16)$$

where

τ_b = Back-drive torque

F_L = Load

l = Screw lead

e = Ball screw efficiency (90%)

4.3 Monopod Hopper Actuation System

The final actuation system used in the monopod hopper Simulink model consists of a brushed DC motor, a ball screw, and a spring for the leg, and a brushed DC motor along with a harmonic drive for the hip. Because of the complexity of modeling brushless DC motors and their controllers in Simulink, for the sake of simplicity, we have restricted ourselves to brushed DC motors. The selection of a ball screw for the leg actuation system is because it is the only practical transmission system which can be used to satisfy two design requirements for the hopper: firstly, the need for high energy efficiency, and secondly, the need for

converting rotary motor motion to a prismatic motion required for injecting energy to the leg spring. On the other hand, two structures that can be used for the hip actuation system are either use a ball screw in a similar fashion as in the ARL Monopod [1], or use a rotary gearing system such as a harmonic drive. In our Simulink model, the second option, i.e., a rotary gearing, is used in the hip actuator.

Choosing parts for the leg actuator

If we desire our hopper design to be physically realizable, it is important to choose models with correct parameter values which correspond to real part specifications. In this section, the procedure for obtaining part specifications for the leg actuation system, i.e., the leg spring, DC motor, and ball screw, is presented.

As soon as the robot makes contact with the ground, the leg motor exerts a torque on the ball screw to keep it from backdriving, and later in the thrust phase it exerts a torque to drive it forward. At any instant, the load on the ball screw can be calculated from the compression length of the leg spring according to the following formula:

$$F_L = K \Delta x, \quad (4.17)$$

where F_L is the load on the ball screw, K is the spring stiffness, and Δx is the compression length of the spring. Using Equations (4.15) and (4.16), the driving and the backdriving torques for the motor can be calculated.

The main design challenge in the driving phase is maximizing the energy added during the short stance period. To maximize energy efficiency, Rad et al. [52] searched for optimized values for DC motor power requirements, ball screw specs, and leg spring constant. They point out that practical considerations impose many constraints on part selections. As an example of these constraints, consider the case where an increased stance time wants to be achieved through decreasing the spring constant. Although this is theoretically valid, however it may not be practical. Since a lower spring constant implies a longer travel for the ball screw, therefore a longer ball screw should be used. A longer ball screw means a ball screw with a higher inertia which in turn requires a heavier motor to be used. As can be seen, there is always a tradeoff between the specifications of different components in a system which should be considered for part selection.

Selecting the leg spring stiffness

For selecting a proper spring stiffness, a number of things need to be taken into account. First is the fact that the spring stiffness is inversely proportional to the stance time. Generally a short stance time is not desired, as it limits the period in which energy is injected into the

system. This places an upper bound on the spring stiffness value. On the other hand, too high a stance time will cause the hopper to tip over if the horizontal speed is to exceed an upper limit, which restricts the maximum achievable speed dictated by the passive dynamic response of the robot. Therefore, there should be a compromise between the leg stiffness value and the horizontal speed. Above that, a low stiffness will cause the spring to compress too much, so that the prismatic joint in the leg reaches its limit of travel. An internal collision then occurs, in which the leg strikes the hip, with consequent loss of energy and a possibility of damage to the mechanism. To find a proper leg stiffness, the passive dynamic response of the hopper was investigated for different horizontal velocities. A value for leg stiffness in the range which produces stable hopping can be chosen as a candidate for the leg stiffness. In our simulations, the value selected for the leg stiffness was 2 kN/m. This leg stiffness would produce a smooth passive hopping at $1 \frac{\text{m}}{\text{sec}}$, and moreover, the period of the stance time is high enough for the injection of the required energy to the system.

Selecting the DC motor and the ball screw

In Chapter 2, we assumed that the robot body mass of 6 Kg already includes motors, transmission drives, and sensing equipment. To find out a suitable DC motor which can lift this body weight, the first point of reference was our hopper simulator with ideal actuators. The stance period was found to be 0.2 sec, from which only a fraction is the thrust phase (0.07 sec). Moreover, it was found out that the maximum required energy to be injected to the system would be achieved by a 16 mm compression of the spring during the thrust phase. On the other hand, the maximum load on the joint happens during the maximum compression of the spring. Having the stance time, the maximum spring compression length (which is the position change of the ball screw), and the maximum load on the ball screw, one can find a suitable DC motor and ball screw pair which satisfy these requirements. Of course, there is no one set of parameter values which satisfies these needs. Because of the order of the complexity of the system and practical constraints, it is not possible to theoretically calculate optimized values for these parts. Simulations can be run using different part combinations to find the most promising one. In this section, a discussion on how we obtain a proper DC motor is given.

The available ball screws have a nominal lead of 2, 3, 4, 5, 10 and up [66] (values are given in $\frac{\text{mm}}{\text{rev}}$). If a ball screw with a lead of $4 \frac{\text{mm}}{\text{rev}}$ is chosen, the specifications for the required leg DC motor can easily be found from this ball screw lead assumption along with the torque and speed requirements obtained from simulations. The values for these parameters are:

$$\begin{aligned}
\text{ball screw lead } l &= 4 \frac{\text{mm}}{\text{rev}}, \\
\text{max output load on the ball screw } F_L^{max} &= 268 \text{ N}, \\
\text{max output velocity of the end screw } V^{max} &= \frac{\text{max end screw position change}}{\text{thrust time period}} \\
&= \frac{0.016 \text{ m}}{0.07 \text{ sec}} = 0.23 \frac{\text{m}}{\text{sec}}.
\end{aligned}$$

From Equation (4.15), we can obtain the output torque and velocity requirements of the motor as:

$$\begin{aligned}
\tau_m^{max} &= 0.177 \times F_L^{max} \times l \\
&= 0.177 \times 268 \times 4 \times 10^{-3} \\
&= 0.1897 \text{ Nm}
\end{aligned}$$

$$\begin{aligned}
\omega_m^{max} &= \frac{V^{max}}{l} = \frac{0.23 \frac{\text{m}}{\text{sec}}}{4 \times 10^{-3} \frac{\text{m}}{\text{rev}}} \\
&= 57.5 \frac{\text{rev}}{\text{sec}} = 3450 \frac{\text{rev}}{\text{min}}
\end{aligned}$$

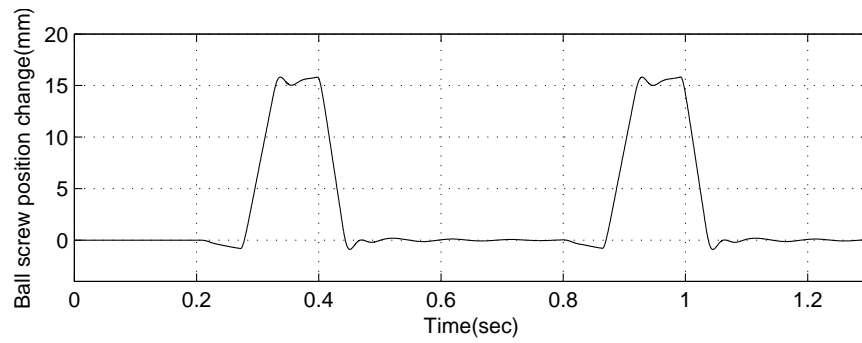
An example of a DC motor which satisfies these requirements is the *Faulhaber Group 32570-24CR* [22] with the specifications given in Table 4.1. The weight of this motor is 0.242 Kg which falls well below the upper limit of 5 Kg for the body weight. Figure 4.2 shows the obtained results of running the simulations with this motor. The figures correspond to a steady hopping with a body height of 0.6 m.

Considerations for the DC motor Simulink models

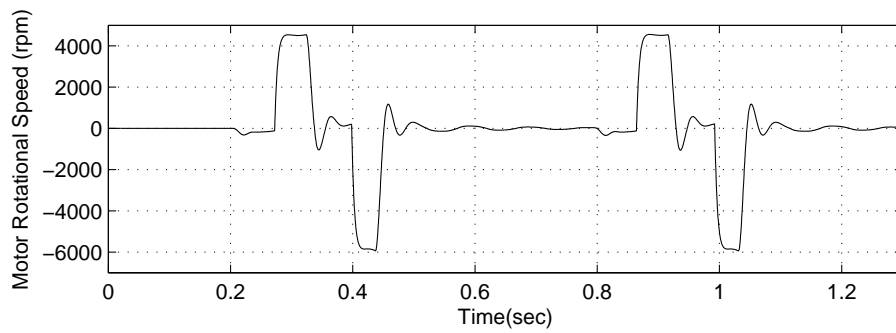
For modeling the hip and leg actuation systems in Simulink, two slightly different DC motor models were used. For the leg actuator, the model was simply based on Equations (4.3) and (4.4). However, to model the hip actuator, since τ_c in Equation (4.4) was not explicitly known, a quick and dirty method popular in robotic applications was used [23]. In this approach, the motor is modeled by decoupling the rotor inertia from the motor model, thus needing to apply the generated output torque through the gearbox to the body inertia connected to that joint, plus the rotor inertia as seen from the output of the gearbox (i.e., multiplied by the square of the gear ratio). The joint velocity is then fed back to the motor through the gearbox, dictating the rotor velocity and the back EMF force. To do so, we only needed to add the square of the rotor inertia as an additional inertia term to the inertia of joint 4 in the dynamics calculation software.

Parameter	Metric Units
Nominal Torque	0.07 Nm
Stall Torque	0.539 Nm
Nominal Speed	5000 rpm
No Load Speed	5900 rpm
Rotor Inertia	$41 \times 10^{-7} \text{ Kgm}^2$
Motor Weight	0.242 Kg
Nominal Output Power	83.2 W
Armature Resistance	1.63 Ω
Rotor Inductance	0.27 mH
Nominal Voltage	24 V

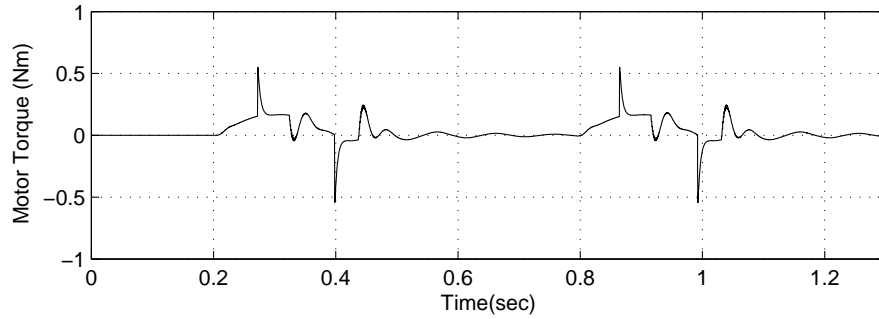
Table 4.1: Faulhaber Group 3257024CR DC Motor Specs



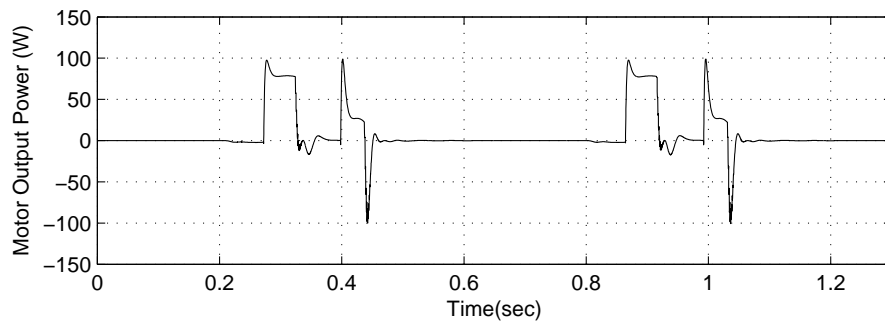
(a) Ball Screw Position Change



(b) Motor Rotational Speed



(c) Motor Output Torque



(d) Motor Output Power

Figure 4.2: Leg actuator variables for a hopping height of 0.6 m in place

Chapter 5

Contact and Friction

Foot-ground interaction is an important part of all legged machine simulations. Understanding the qualitative behaviors of the contact forces helps one to better predict and control the motion of a robot in a contact scenario. On the other hand, having accurate ground models helps to accurately measure the energy dissipation that occurs as a result of collision and sliding on a surface. That is why we have dedicated a whole chapter to the study of the foot-ground interaction. Generally, this interaction can be studied in terms of the normal and tangential forces that act on a body while making contact with the ground (Figure 5.1 shows the force diagram for a block on the ground). Based on this categorization, this chapter is divided into two main sections. In the first section, normal contact forces are discussed. Tangential contact forces referred to as friction are the focus of the second section. Towards the end of this chapter, proper models that can be trusted for representing contact and friction forces and which capture the qualitative behaviors of these forces are selected. Detailed dynamic simulation has been made possible with the right choice of models and parameter values for these forces.

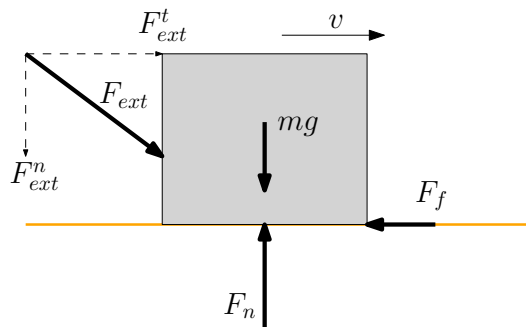


Figure 5.1: Force diagram for a block on the ground, showing external force (F_{ext}), friction force (F_f), force due to gravity (mg), normal ground force (F_n), and velocity (v)

5.1 Contact and Impact

The choice of the contact model largely depends on the nature of the application or analysis that is carried out. Generally, contact models are divided into two categories: purely rigid contacts and compliant contacts. A rigid contact is usually referred to as *impact*. Impacts between two rigid bodies have a very brief duration and involve high force levels, rapid dissipation of energy, and high acceleration and decelerations. *Contact* is a term which is frequently used interchangeably with impact, however, it better describes a compliant contact which is a continuous process and occurs over a finite time [26]. Descriptions of both the rigid contact models and the compliant contact models are given in this section.

5.1.1 Rigid Body Models

Also known as impulse-momentum or discrete models [39], the rigid-body model's major characteristic is that no deformations are allowed at the points or surfaces of contact between two bodies. Hence, these models are mainly used for describing contact between two bodies where little or no deformation takes place. Although rigid-body models are straightforward to use and lead to computationally efficient planning algorithms, they are inherently incapable of describing the full range of contact phenomena. For example, these models are basically inefficient for multi-contact scenarios [59, 9, 48], or contact scenarios where considerable deformations occur [9, 39]. Moreover, inclusion of friction in the contact [13, 45] causes considerable problems with these models. To overcome the shortcomings of rigid models, one must introduce compliance.

5.1.2 Compliant Models

Known by various names such as continuous, force based, or regularized models [39], the major characteristic of a compliant model is that deformations occur as a result of applied forces at the contact. Above that, interaction forces act in a continuous manner during impact and are explicitly known as a function of the local indentation and its rate. Thus by merely having these two values, contact forces can be obtained and dynamics analysis can be carried out by adding the contact forces to the equations of motion during their action period. The main reason for using compliant contact models is that not every contact phenomenon is between two rigid bodies and many contact phenomena involve a hard and a soft body or two soft bodies. In these contact scenarios, rigid-body models are inadequate for dynamics analysis. Another reason for using a compliant model is that implementing a compliant contact is easier than a rigid contact since there is no need for impulsive dynamics calculations. The main disadvantage, however, is that for very stiff surfaces, such as a tile floor, the impact duration or transient phase of the impact model is very short, hence the

corresponding differential equations will be numerically very stiff, and including them can greatly complicate the simulation and analysis of walking or running; moreover, determining physically reasonable parameters for a compliant impact model is itself a challenging problem. It is also notable that an exact model of the deformations that occur in a contact scenario is quite complex, and it is common to use lumped parameters or reduced-order compliance models.

Because of the nature of legged robot research, which involves multi-contact situations, flexible surfaces, and the need to accurately measure friction and contact forces, and on the other hand the shortcomings of rigid-body models in these situations, rigid contact models are excluded from further analysis. However, some rigid contact concepts that are required for understanding compliant models can be found throughout this text and are described in their place. In what follows, a summery of some existing compliant contact force models is given. In this context we only deal with explicit models, i.e., contact models in which normal contact force F_n is described as an explicit function of local indentation x and its rate \dot{x} , formalized by [26, 9]

$$F_n \equiv F_n(\dot{x}, x) = F_{\dot{x}}(\dot{x}) + F_x(x). \quad (5.1)$$

Linear Spring-Damper Model

This is the simplest compliant model, where a linear damper in parallel with a linear spring, accounting for the dissipation of energy and the elastic behavior respectively, are used to represent the contact force [27]. This model is shown in Figure 5.2(a), and consists of a massless surface connected to a fixed surface by a spring and a damper. While in contact, the contact force for a colliding body is defined as:

$$F_n = b\dot{x} + kx, \quad (5.2)$$

where b is the damping and k is the stiffness of the surface.

Figure 5.2(b) shows the curve of the *contact force vs. indentation* for this contact model. Although the simplicity of this model has made it a popular choice in various research work, there are some weaknesses attributed to this model which make it unsuitable for representing real impact scenarios. Among these shortcoming are [46, 26] :

- The contact force is not continuous at the beginning of impact (see Figure 5.2(b)) due to the form of the damping term.
- At the point of separation, a negative force holding the objects together is present and causes an sticking effect (see Figure 5.2(b)).

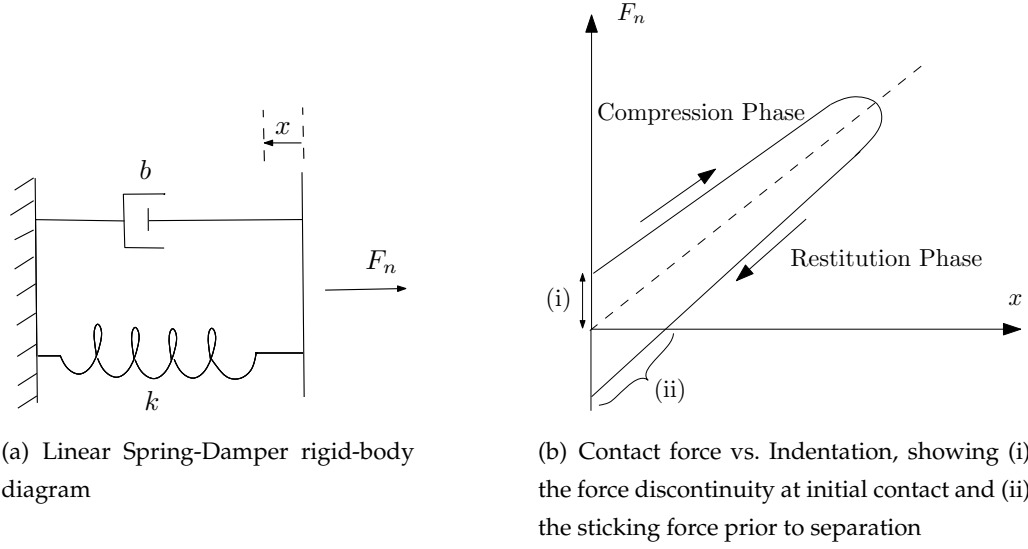


Figure 5.2: Linear spring-damper contact model, (a) Rigid-body diagram and (b) Contact force vs. Indentation

- Based on empirical results from [27], the *coefficient of restitution*¹ is dependent on impact velocity. However, the equivalent coefficient of restitution obtained for this model does not depend on impact velocity.

Hertz Model

The Hertz model for normal contact force only considers elastic deformations and no damping is taken into account. However, the elastic deformation is modeled with a nonlinear spring which represents a more accurate elastic contact behavior than a linear spring. For the Hertz model, the impact force is defined as [27]:

$$F_n = kx^n, \quad (5.3)$$

where k and n are constants depending on material and geometric properties, and are computed by using elasto-static theory [26]. There exist some modified versions of the Hertz model which account for plastic deformations as well [45], but these models are generally hard to implement and are not used in practice.

¹Impacts are often characterized by the coefficient of restitution, e , defined as $e = -v_o/v_i$, where v_i is the relative normal velocity before collision, and v_o is the relative normal velocity after collision. It has been shown that at low impact velocities and for most materials with a linear elastic range [33, 27] the coefficient of restitution can be approximated by the equation $e = 1 - \alpha v_i$, where α is a constant and depends on the damping and springiness of the surface.

Nonlinear Spring-Damper Model

A nonlinear spring-damper combination provides a more accurate representation of the physical behavior for colliding solids compared to the two previous models. Developed by Hunt and Crossley [33], this model provides an alternative representation of energy dissipation in a contact. While having the advantages of both the Hertz and linear spring-damper models, this model overcomes the weaknesses of the latter. The contact force for this model is defined as:

$$F_n = bx^p \dot{x}^q + kx^n, \quad (5.4)$$

where b is the damping, k is the stiffness, and p , q , and n are constants depending on the surface geometry and material of the contact. It is standard to set $p = n$ and $q = 1$, hence obtain:

$$F_n = bx^n \dot{x} + kx^n. \quad (5.5)$$

In Equation (5.4) and (5.5), the damping coefficient b is related to the coefficient of restitution, since both are related to the energy dissipated in the impact. Other than Hunt and Crossley [33], Marhefka and Orin [46] and Lankarani and Nikravesh [43] also use a similar continuous contact model with local compliance in the normal direction for direct central and frictionless impacts. The advantage of this model over the linear one is that while the computational requirements are similar, the contact forces evolve continuously with time, they are able to handle impacts in the sense that the equivalent coefficient of restitution is dependent on the impact velocity, and, moreover, no sticking effects are present in these models. These properties make the nonlinear spring-damper model an attractive choice for robotic simulation systems, and it is one of the most widely used compliant models in the robotics field.

Double-State Visco-Elastic Model

In fact, the double-state visco-elastic model is an extension of the simple linear spring-damper model. It consists of a spring-damper pair in series with a spring as shown in Figure 5.3. The difference between this model and the previous ones is that two state variables are used to represent this model, whereas only one state variable is used in the previous models. The major advantage of this model over the linear spring-damper model is because of the series spring, the contact forces evolve continuously with time. From Figure 5.3, the contact force for this model is obtained as:

$$F_n = k_2(x - y), \quad (5.6)$$

where the value of y is the solution to the differential equation:

$$b\dot{y} + k_1y + k_2(x - y) = 0. \quad (5.7)$$

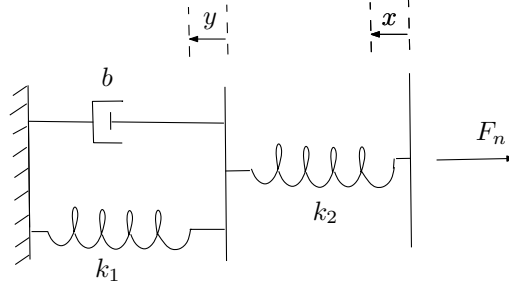


Figure 5.3: Double-state visco-elastic contact model

5.2 Friction

While damping due to the normal contact force is a significant phenomenon through which energy is dissipated during the impact phase, to more rigorously measure energy dissipation, we must also consider the effect of the friction acting along the surface of the contacting bodies in the region of contact.

Friction is a natural phenomenon that is quite hard to model, however, the qualitative mechanisms of friction are fairly well understood. In short, friction is the tangential reaction force between two surfaces in contact [49] (Figure 5.1, p. 32). There exists a rich body of literature on friction modeling and representation. A 1994 survey cites 280 articles addressing issues on friction modeling, control and applications [6]. The motivation for improved friction models originates in the need for better prediction and control of dynamic systems. However, in our project an accurate friction model is desired so that we can closely represent the qualitative behaviors of friction and accurately measure energy dissipation in the contact. Table 5.1 lists some of the most important aspects of friction observed in experimental studies. A model is considered competent if it captures most if not all of these behaviors.

Earlier in this chapter, it was discussed that the choice of a rigid contact model causes limitations in introducing friction. One of these limitations is that rigid models basically restrict us to the choice of Coulomb's law of friction. This can be attributed to the fact that rigid-body models are based on an impulse momentum formulation, and Coulomb's law is the only friction model easily representable in the impulse domain [26]. As will be discussed in this section, the Coulomb's model is not the best representation of friction in reality because it does not capture some of the important observations in friction behavior listed in Table 5.1. The choice of a compliant model in this work has allowed us to freely choose from all the available friction models in the literature. After presenting compliant contact models in the previous section, this section is dedicated to the review of the most widely used friction models.

Behavior	Description/Figure
Coulomb's	Figure 5.5(a)
Steady velocity friction	Friction as a function of velocity (Figure 5.5(b))
Static friction (Stiction)	Friction when sticking (Figure 5.5(c))
Breakaway force ¹ relation with force rate	Figure 5.4(b)
Stribeck effect	Continuous behavior of the friction force (Figure 5.5(d))
Frictional lag	Hysteretic behavior observed in friction force while varying the velocity periodically (Figure 5.4(a))
Pre-sliding displacement	Spring-like behavior before gross sliding occurs (Figure 5.4(c))
Stick-slip behavior	Friction as a function of displacement rather than velocity (Figure 5.4(d))

¹ Breakaway force:

Force required to overcome static friction

Table 5.1: Some important qualitative behaviors of friction observed in experimental studies

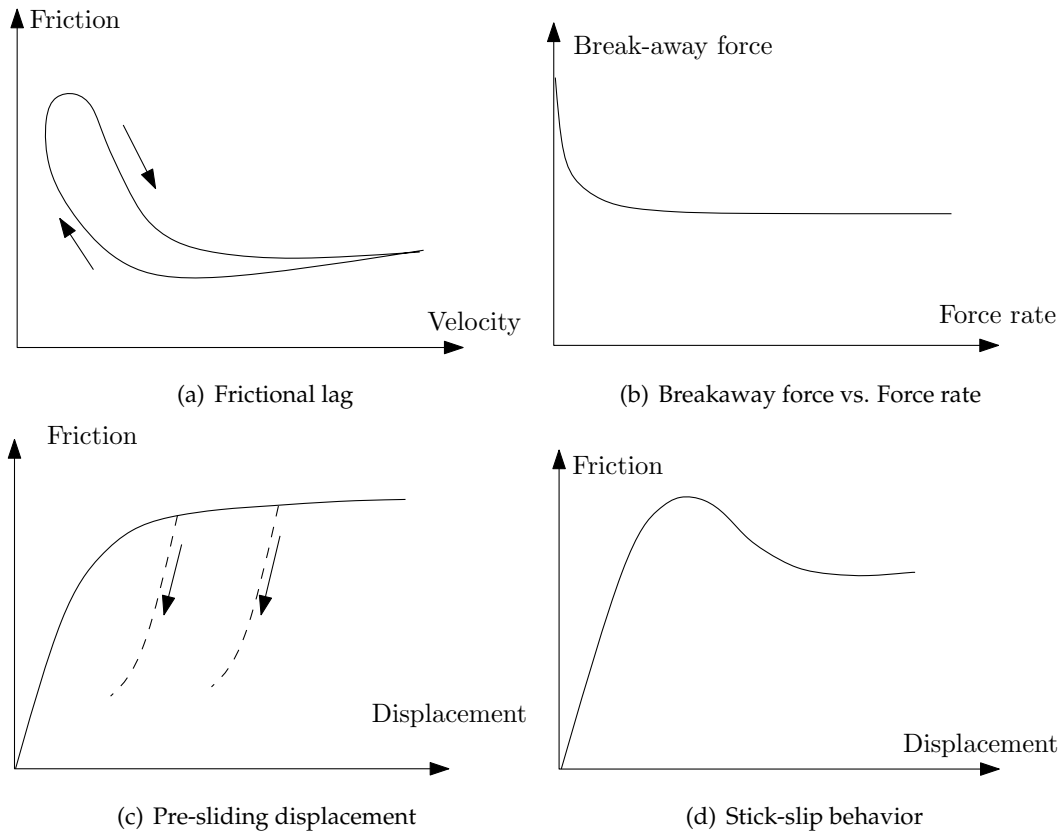


Figure 5.4: Four of the friction behaviors mentioned in Table 5.1

5.2.1 Static Friction Models

In the static models of friction, the presupposition is that the relationship between the friction force and velocity is through a static function. Different force components have been considered in a static model, such as: Coulomb's friction, viscous friction, stiction, and stribeck effect. These force components are generally referred to as the classical friction models. Of course, a mixture of these components altogether gives a more realistic overall friction force. A brief description and formulation for each of these subcomponents, mostly taken from [49], is presented in this section.

Classical models

Coulomb's friction

Coulomb's description of friction classifies friction into two regimes: static and kinetic. Static friction is the friction force at zero velocity and can take any value in the range $[-F_c, F_c]$, where F_c is characterized by the coefficient of Coulomb's friction as:

$$F_c = \mu_c F_n. \quad (5.8)$$

In this equation, μ_c is the coefficient of Coulomb's friction and F_n is the normal force from the surface acting on the body. On the other hand kinetic friction is given in terms of F_c as:

$$F_f = F_c \operatorname{sgn}(v), \quad (5.9)$$

where v is the velocity of the sliding body. The Coulomb's friction force as a function of velocity is shown in Figure 5.5(a).

Viscous friction

Viscous friction describes the friction force that causes the viscosity of lubricants [58]. It consists of a linear relation between friction force and velocity and is described as:

$$F_f = \sigma v, \quad (5.10)$$

where σ is the coefficient of viscous friction and v is the velocity of the sliding body. Viscous friction is usually coupled with Coulomb's friction. The resultant force as a function of velocity is shown in Figure 5.5(b).

Stiction or static friction

Stiction describes the friction force between two touching bodies when they are static relative to each other. It is characterized by the coefficient of static friction μ_s which has a higher value than the coefficient of Coulomb's friction μ_c . Based on this description of friction, the friction force between a body and a surface at zero velocity is equal to the tangential external force applied to the body and increases up to the stiction force limit which is in turn defined as:

$$F_s = \mu_s F_n, \quad (5.11)$$

where μ_s is the coefficient of static friction and F_n is the normal force from the surface acting on the body. Based on this explanation for the static friction, the combination of the static, Coulomb's, and viscous friction is (Figure 5.5(c))

$$F_f = \begin{cases} \sigma v + F_c & \text{if } v \neq 0 \\ F_{ext}^t & \text{if } v = 0 \text{ and } |F_{ext}^t| < F_s, \end{cases} \quad (5.12)$$

where F_f is the friction force, F_{ext}^t is the tangential external force acting on the body, F_s is the static friction force limit, F_c is the Coulomb's friction term, and σv is the viscous friction term.

Stribeck effect

Stribeck [63] observed that the friction force does not decrease discontinuously as soon as sliding begins, but that the velocity dependence is continuous. The formulation for this friction model is:

$$F_f = \begin{cases} F_{st}(v) & \text{if } v \neq 0 \\ F_{ext}^t & \text{if } v = 0 \text{ and } |F_{ext}^t| < F_s, \end{cases} \quad (5.13)$$

where F_f is the friction force, F_{ext}^t is the tangential external force acting on the body, F_s is the static friction force limit, v is the velocity of the sliding body, and $F_{st}(v)$ represents Stribeck effect. Several parameterizations for $F_{st}(v)$ have been proposed in the literature [5]. The following parametrization is one of the common ones used:

$$F_{st}(v) = F_c + (F_s - F_c)e^{-|v/v_s|^2} + \sigma v, \quad (5.14)$$

where v_s is called the Stribeck velocity, F_c is the Coulomb's friction term from Equation (5.8), F_s is the static friction term from Equation (5.11), v is the velocity of the sliding body, and σv is the viscous friction term from Equation (5.10). The continuous behavior of friction or the Stribeck effect is shown in Figure 5.5(d).

Double-state friction models

These models are basically modifications of the classical models of friction. However, we refer to them as double state friction models since two separate equations are used for describing the sticking and slipping behaviors respectively. Through these means, these models are capable of including some of the observed dynamic friction behaviors which will be discussed in the next section. The double-state model that we used in our work consists of a linear spring-damper (similar to the spring damper contact model discussed earlier) for capturing the presliding behavior during sticking, and the Coulomb's law of friction for capturing the sliding behavior. Another example of such a model which uses two separate equations for describing the sticking and slipping behaviors is Armstrong's model introduced in [6].

5.2.2 Dynamic Friction Models

In later studies by Armstrong-Helouvry [5, 6], the necessity of having a friction model involving dynamics was investigated. In fact, the classical models of friction did not explain some important aspects of friction behavior shown in Table 5.1, among them are the hysteretic behavior observed when studying friction for non-stationary velocities, variations in the break-away force with the experimental condition, and the small displacements that oc-

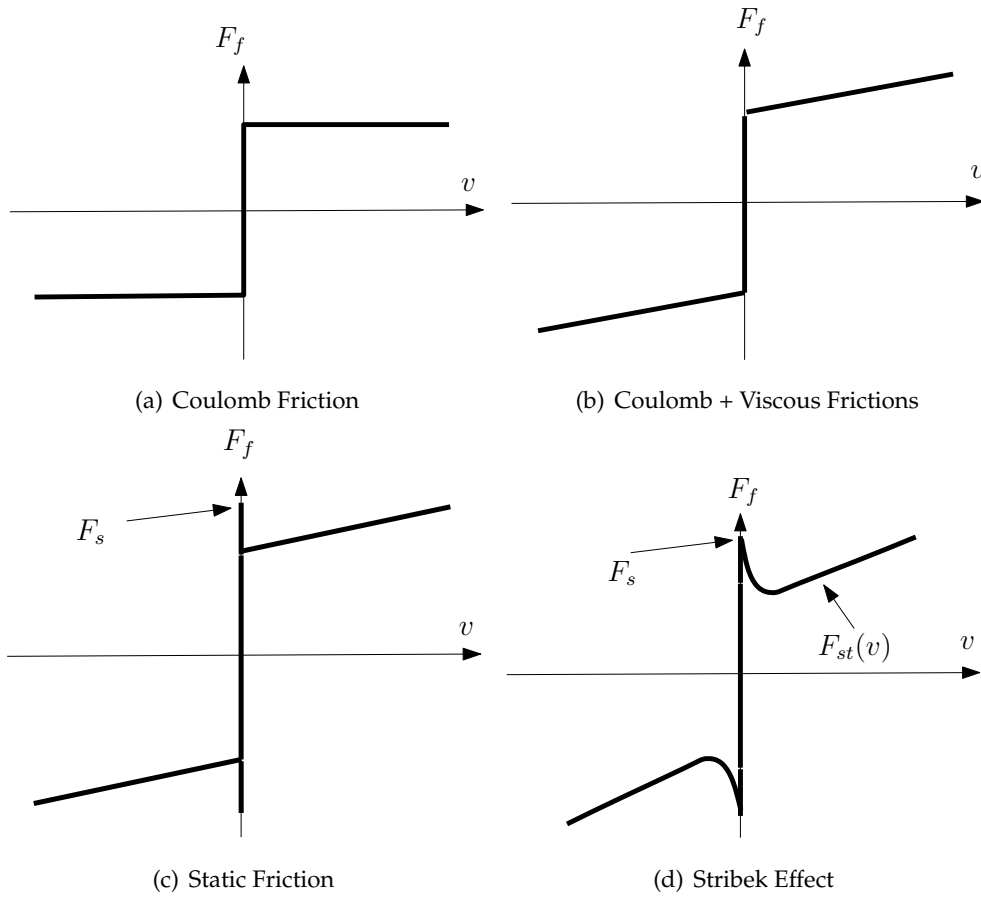


Figure 5.5: Classical friction models

cur at the contact interface during stiction (presliding behavior¹). Various friction models that include dynamics are described in this section.

Bristle models

At a microscopic level, two surfaces making contact can be visualized through elastic bristles between them (Figure 5.6). An applied tangential force will deflect the bristles and cause a friction force. If the forces become sufficiently large, some of the bristles deflect so much that they slip. Due to the irregularity in the surfaces, this phenomenon is highly random. Haessig and Friedland [31] present two models of friction, namely, the bristle model and the reset integrator model, which capture this physical phenomenon. Among these two models, the first is computationally inefficient, but the latter which describes the aggregated behavior of the bristles, is well suited for simulation purposes. Many of the later models include this bristle interpretation of friction in their equations.

¹The presliding behavior very much resembles that of a connection with a stiff spring with damper and is sometimes referred to as the Dahl effect.

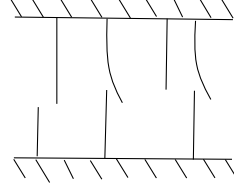


Figure 5.6: The friction interface as a contact between bristles

Dahl's model

Dahl's model introduced in [16] is essentially Coulomb's friction with a lag in the change of friction force when the direction of motion is changed. Dahl describes friction as the macroscopic result of quantum mechanical bonds between two contact surfaces as captured by bristle models. Based on Dahl's observations, the transition from static to kinetic friction is alike that of elastic to plastic deformation in ductile materials. At low loads, the bonds behave like springs, but for stronger loads, permanent displacement similar to plastic deformation will affect the bonds. This results in a hysteresis behavior also found in plastic material deformation. Increasing the load beyond the strength limit of the bonds will cause a rupture in them. Based on his observations, Dahl formulated friction as:

$$\begin{aligned} F_f &= \sigma_0 z, \quad \sigma_0 > 0 \\ \dot{z} &= \dot{x} \left(1 - \frac{\sigma_0}{F_c} \operatorname{sgn}(\dot{x}) z \right)^i, \end{aligned} \quad (5.15)$$

where F_f is the friction force, σ_0 is the contact stiffness, $z(t)$ specifies the state of elastic strain in the frictional contact (bristle deflection), x is the rigid body displacement, and F_c is the Coulomb's friction term from Equation (5.8). Dahl's model has the advantage of being simple while at the same time capturing many aspects of friction and has been extensively used for modeling the friction force in simulations. However, Dahl's model has a number of shortcomings which are discussed in the next section.

Bliman-Sorine and LuGre models

Bliman-Sorice and LuGre models are dynamic friction models which are closely related to Dahl's representation of friction. As mentioned before, Dahl's model has a number of shortcomings, in particular, it does not include two of the important friction behaviors. Firstly, the friction force is dependent on displacement only and does not consider the velocity dependence of friction which is especially present in the friction behavior of lubricated contact surfaces. Secondly, the model does not include the stiction regime and the Stribeck effect. These shortcomings were later addressed in the Bliman-Sorine [8] and LuGre [17] models.

These two models are considered similar in the sense that they are both modifications of Dahl's model; however, they are different in many aspects. The main differences between these models are: rate dependency, model order, and damping. Overall, the LuGre model includes more friction phenomena than the Bliman-Sorine model and is a better representation of friction in reality [25].

Based on Bliman and Sorine, friction is not explicitly dependent on velocity, but is only dependent on the distance traveled after a velocity zero crossing. Because of this rate independence attribute of friction, it can be expressed in terms of distance. By replacing the time variable t , with a space variable s , the Bliman and Sorine model can be expressed as a linear state-space system with state variable s as:

$$\begin{aligned}\frac{dx_s}{ds} &= Ax_s + Bv_s \\ F_f &= Cx_s,\end{aligned}\tag{5.16}$$

where $s = \int_0^t |v(\tau)| d\tau$, $v_s = \text{sgn}(v)$, and v is the velocity of the sliding body.

The LuGre model, on the other hand, is related to the bristle interpretation of friction. It also captures the Stribeck effect, viscous friction, rising static friction, and frictional memory during slip. It is formulated as:

$$\begin{aligned}F_f &= \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 \dot{x}, \quad \sigma_0, \sigma_1, \sigma_2 > 0 \\ \dot{z} &= \dot{x} \left(1 - \frac{\sigma_0}{|F_{st}(\dot{x})|} \text{sgn}(\dot{x}) z\right),\end{aligned}\tag{5.17}$$

where $F_{st}(\dot{x})$ is the steady-state friction force versus rigid-body velocity, mentioned earlier as the Stribeck curve. A proposed parametrization for $F_{st}(\dot{x})$ which captures the Stribeck effect is of a form similar to Equation (5.14), i.e.,

$$F_{st}(\dot{x}) = F_c + (F_s - F_c) e^{-(\dot{x}/v_s)^2},\tag{5.18}$$

where F_c , F_s and v_s are the Coulomb's friction term, stiction force term and the Stribeck velocity respectively. F_c and F_s can in turn be written in terms of Equations (5.8) and (5.11). With this description, the model is characterized by 6 parameters: $\sigma_0, \sigma_1, \sigma_2, \mu_c, \mu_s, v_s$. It's worth mentioning that the LuGre model reduces to Dahl's model by setting $\sigma_1 = \sigma_2 = 0$ and $F_{st}(\dot{x}) = F_c$.

Extensions of the LuGre model

The LuGre model has been subject to several criticisms. The nonphysical drift phenomenon which results from modeling presliding as a combination of elastic and plastic displacement is addressed by Dupont et al. [21] in their so called elasto-plastic friction model. However, this model in turn does not address the problem of lack of nonlocal memory behavior. Later,

Swevers et al. [64] proposed a model called the Leuven model, which further extends the LuGre model to include the presliding hysteresis with nonlocal memory behavior. Yet, this model did not remain attractive because of its implementation difficulties. To overcome these difficulties, Lampart et al. [42] incorporate a Maxwell-slip model representation. Extensions of this model by the same authors were studied in [3, 4]. In the latter, the Coulomb's slip law is replaced by a rate-state law, which makes it suitable for quick simulation and control purposes, being both easy to implement and of high fidelity.

Final remarks

It is proper to add that the models mentioned in this section are generally judged on two basic criteria: their fidelity in the sense of how closely they model real physical behavior, and their ease of implementation. For example, there exist both single-state and multi-state models in the literature for capturing presliding behavior. As can be expected, the multi-state models are capable of producing a more sophisticated presliding behavior. In fact this can be observed in the multi-state friction model of Swevers et al [64] as compared with the single-state LuGre model. However, the single-state models still hold appeal in applications such as machine tools due to their comparative simplicity for system analysis, controller design and implementation[21]. Therefore, among the dynamic models, we have only used the single-state ones in our work.

5.3 Final Choice of Contact and Friction Models

Contact models

We did not use the linear spring-damper model due to its shortcomings as discussed in Section 5.1.2. Subsequently as a result of their advantages over the linear spring-damper model, the double-state visco-elastic and the nonlinear spring-damper models were both implemented in Simulink. The parameter values selected for these two models are given in Table 5.2.

For the double-state model, the data given do not represent a real surface but are synthesized. However, as will be shown in experimental results, for our selected values, the magnitude and shape of the normal contact force from this model closely resembles that of the nonlinear spring-damper model for which real data was available.

The first set of parameter values given for the nonlinear spring-damper model in Table 5.2, are obtained from [46] in which the authors present a number of parameter values for the model, yet they do not state what surface geometry and material the data correspond to. Nevertheless, the model parameters they have provided for a foot-ground interaction of

a walking machine is used in our simulations. Other parameter values given in Table 5.2 are the ones produced by Wight [70]. The data correspond to a surface made of a 3.88mm layer of neoprene mounted on 12.5mm of plywood for a robot with rounded-profile contact points made from hard nylon. Wight also explains the full procedure for obtaining these values, in which he uses static compression tests to find parameter values of the second term in Equation (5.4), and dynamic tests for the first term. It is worth mentioning that there also exist algorithms developed by researches for online estimation of parameter values for the nonlinear-damper model [19] .

Model	Data Source	Parameters
Double-State Visco-Elastic	Synthesized	$k_1 = 5.0 \times 10^4$, $k_2 = 2.5 \times 10^5$ $b = 100$
Nonlinear	Marhefka and Orin [46]	$\alpha = 0.5$, $k = 5.0 \times 10^4$, $b = 3.7 \times 10^4$ $n = 1.0$, $p = 1.0$, $q = 1.0$
Spring-Damper	Wight [70]	$\alpha = 0.5$, $k = 7.21 \times 10^7$, $b = 3.8 \times 10^4$ $n = 2.31$, $p = 1.1$, $q = 1.0$

Table 5.2: Parameter values for the compliant contact models

Friction models

Initially, four of the friction models described in this chapter were modeled in Simulink. These models were: the double-state friction model which is a combination of the Coulomb's law and the spring-damper model, Dahl's model, and the LuGre model. However, in the simulation of the Dahl and the LuGre models for the hopper, a problem was confronted. In both of these models, the dependance of the state variable on the normal force prevented us from directly using them as the ground model. In fact, the differential Equations (5.15) and (5.17) become very stiff in periods of the hopping cycle like loading and unloading where the normal force is near zero. A remedy to this problem is to modify these models so that the differential equations would represent the coefficient of friction rather than the friction force itself, which results in a differential equation for the state variable that is independent of the normal force. Such a model, which is an adapted version of the Bliman and Sorine model, was found in the literature (Wight [70]). The equations representing this adapted friction model are:

$$\begin{aligned} \dot{u} &= \frac{-3|\dot{x}|}{s_p} u + \frac{3f_k}{s_p} \dot{x} \\ F_f &= u F_n, \end{aligned} \quad (5.19)$$

where \dot{x} is the tangential velocity of the contact point relative to the surface, f_k is the coefficient of kinetic friction, s_p is a rough estimation to the displacement before saturation of the friction occurs, and u is the state variable representing the coefficient of friction. Wight also presents an experimental procedure for measuring parameter values for the same surface given in Table 5.2. These parameter values are given in Table 5.3.

Parameter values used for the double-state friction model are also given in Table 5.3. However, these data are not obtained from a real surface, but are synthesized. Nevertheless, in the simulations, this choice of parameters captures fairly the qualitative behaviors of friction.

Model	Data Source	Parameters
Double-State Friction	Synthesized	$k = 5.0 \times 10^4$, $b = 1.0 \times 10^2$ $\mu_c = 0.5$
Bliman and Sorine	Wight [70]	$f_k = 0.6$, $s_p = 0.001$
LuGre	de Wit [17]	$\sigma_0 = 10^5$, $\sigma_0 = \sqrt{10^5}$, $\sigma_0 = 0.4$ $F_c = 1.0$, $F_s = 1.5$, $v_s = 0.001$

Table 5.3: Parameter values for the friction models

Chapter 6

Simulation and Results

As mentioned in Chapter 1, MATLAB and Simulink were used for simulation purposes. For the dynamics calculations, Featherstone's dynamics calculation routines available as a package in MATLAB were used. The end result of the simulations was a Simulink model for the monopod hopper, where successful control of hopping height and speed of the robot were achieved. Moreover, a block for measuring energy expenditure is also included in the models. In addition to the monopod hopper, a Simulink model for the bipedal runner was also developed, where stable control was successfully achieved, however, no energy efficiency measurement was performed in this case.

In the first part of this chapter, we present simulation results for the monopod hopper by first showing the contact and friction forces that act on the robot during stance, and briefly discussing the qualitative behavior of these forces. Next, the results of height and velocity tracking are presented. A large part of this chapter is dedicated to the energetic aspects of the monopod hopper. Electric input and mechanical output energy for a stable hopping are presented, and various energy efficiency measures for the robot are calculated. After showing the energetic cost of hopping from the actuation point of view, towards the end of this chapter a discussion on the improvements that would be gained as a result of the leg design modifications and an alternative control law for the hip are presented. Other possible improvements are also mentioned in brief.

6.1 Contact and Friction Force

We presented parameter values for two contact models in Chapter 5: the nonlinear spring-damper model and the double-state visco-elastic model. As for the friction models, parameter values for the double-state model, as well as the Bliman-Sorine model were presented. In this section, these two forces are shown for two experimental runs, where in the first experiment, the Bliman-Sorine friction model and the nonlinear spring-damper model are

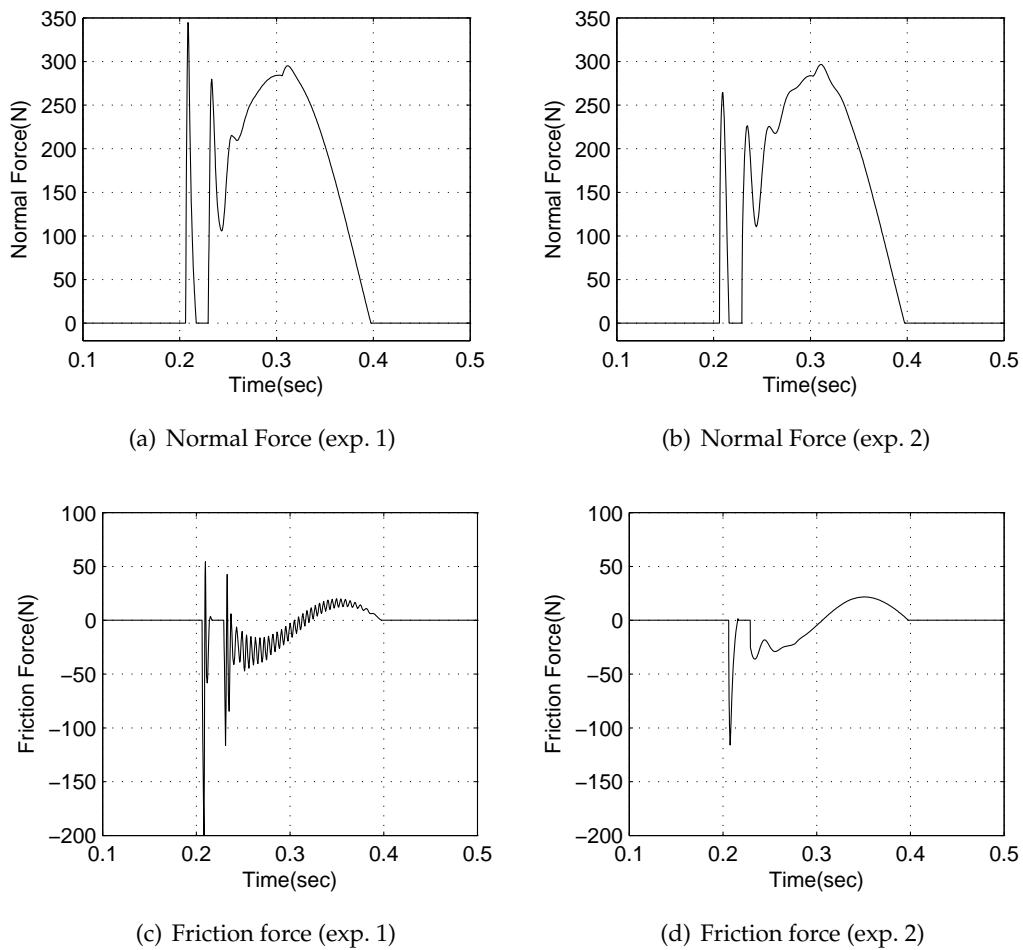
used. The second experiment uses the double-state friction model along with the double-state visco-elastic contact model. The contact and friction forces obtained for a hopping cycle corresponding to these two experiments are presented in Figures 6.1(a) to 6.1(d). Basically, the contact and friction forces are functions of vertical and horizontal positions and velocities of the foot-tip during the stance phase. We have shown the foot-tip vertical and horizontal positions during the stance phase for the two experimental runs in Figures 6.2(a) to 6.2(d).

In Figure 6.1(a) and 6.1(b), where *normal contact force vs. time* is shown, the parts of the curves that correspond to a positive value for the normal force correspond to the periods of the hopping cycle where the robot is making contact with the ground. As can be seen, when the robot touches the ground for the first time after the flight phase, there is a peak in the normal force. In fact, because of this large normal force, after its first contact, the robot's foot-tip will bounce off the ground and the robot loses contact for a short period. The value of this peak force is higher for the first curve than for the second one. This is due to the difference in the parameter values and the nature of the used contact models. But other than that, the contact forces obtained for the two experiments resemble each other. Indeed, the ground stiffness values selected for the second experiment, where we use synthesized data, were chosen close to the parameter values of experiment 1 for which real data were available. Later in this chapter, the energetic losses of the second contact model, i.e., the nonlinear spring-damper model, are discussed. The applicability of these two ground models for sustained contact conditions, as well as transitions to and from sustained contact, can be clearly seen from the figures. For both of these models, the contact force builds up and vanishes continuously with time.

As for the friction force, again the overall shapes of the two models look similar, however, high frequency dynamics are present for the Bliman-Sorine friction model. Moreover, for this model, small horizontal displacements of the foot-tip occur during the stance phase, which represent the slight slipping of the foot on the ground (Figures 6.2(a) to 6.2(d)). For the second model, slipping is negligible and the foot's horizontal displacement is mostly elastic.

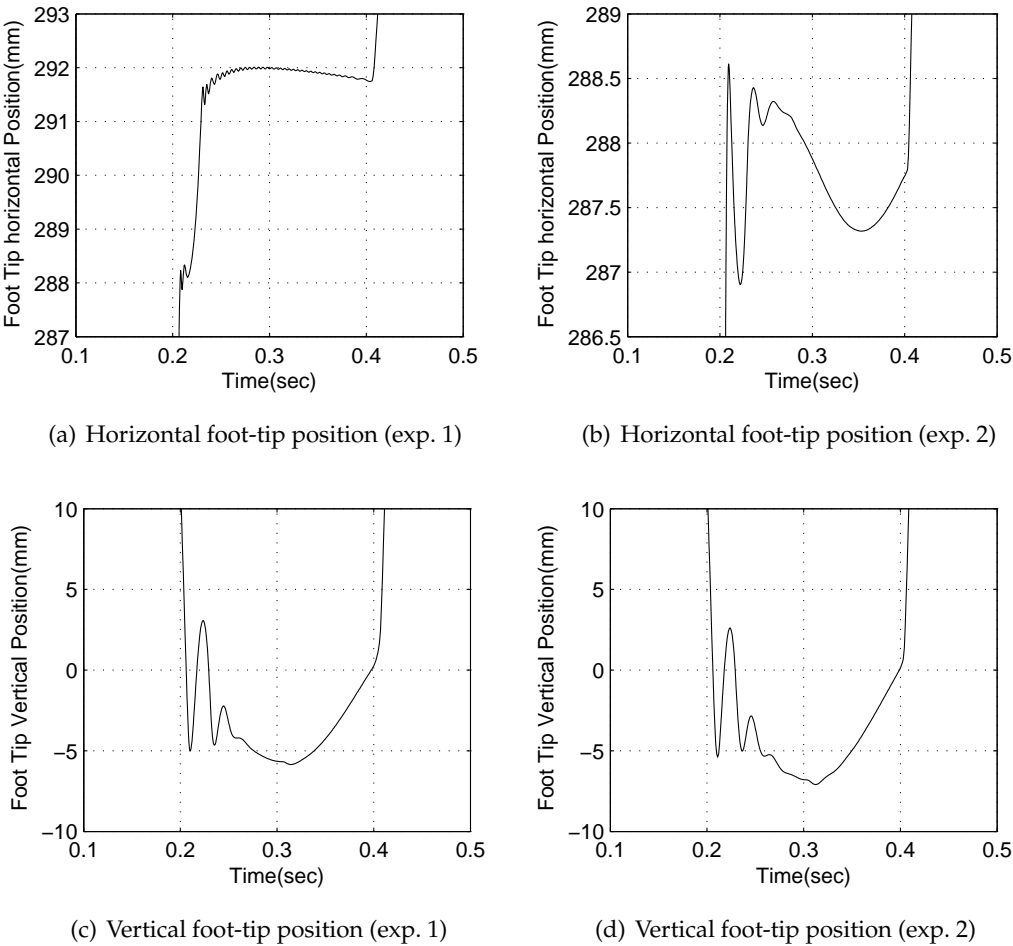
Contact model energy dissipation characteristics

After showing the contact and friction forces, some of the major characteristics of the nonlinear spring-damper contact model, discussed in more detail by Marhefka and Orin [46], are shown and discussed here to give a better understanding of the foot behavior after impact. The phase diagram for this contact model is given in Figure 6.3, where a mass of 0.5 Kg (equal to the leg mass of our monopod) hits a massive object with different initial velocities. From the figure, it can be seen that the contours converge to a fixed separation velocity as



	Experiment 1	Experiment 2
Normal force model	Nonlinear spring-damper	Double-state visco-elastic
Friction force model	Bliman-Sorine	Double-state friction
(e) Models used in each experimental run		

Figure 6.1: Contact and friction force for two experimental runs of the hopper



	Experiment 1	Experiment 2
Normal force model	Nonlinear spring-damper	Double-state visco-elastic
Friction force model	Bliman-Sorine	Double-state friction

(e) Models used in each experimental run

Figure 6.2: Foot-tip horizontal and vertical positions for two experimental runs

the initial penetration velocity v_i becomes large. This value can be shown to be $v = -\frac{2}{3\alpha}$, because as soon as the velocity reaches $v = -\frac{2}{3\alpha}$ the forces due to the spring and damper cancel out each other, while the mass is still returning to the surface, which implies zero force acting on the mass and zero change in velocity from that point.

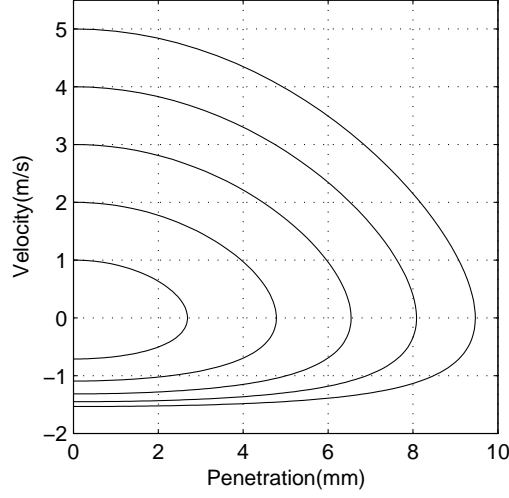


Figure 6.3: Phase diagram for $v = 5, 4, 3, 2$, and 1 m/s (outer to inner loop), where $m = 0.5$ Kg, $k = 50$ kN/m, and $\alpha = 0.4$ s/m

The impact response characteristics of the nonlinear spring-damper model are given in Figures 6.4(a), 6.4(b), and 6.4(c) (again the data is produced for a mass of 0.5 Kg hitting a massive object). Figure 6.4(a) shows the *contact force vs. penetration depth* during impact for different values of α . The energy dissipation is calculated as the area inside the loops, i.e.,

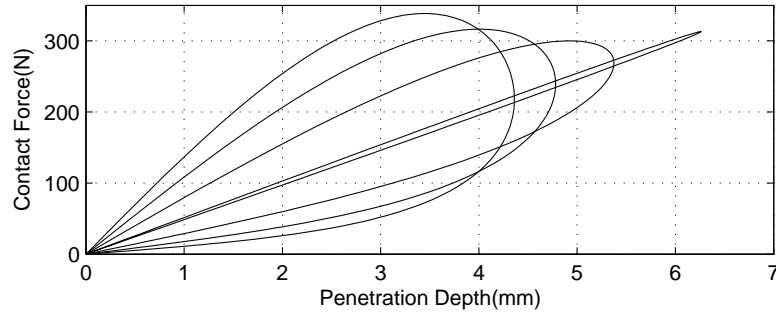
$$\Delta E = \oint F dx. \quad (6.1)$$

From the figure, it is clear that energy loss increases with increasing α . Figure 6.4(b) shows the variation of contact force vs. penetration depth whereas Figure 6.4(c) shows the *contact force vs. time* for different values of k . It is clear that the area within the curves is constant for all values of k , emphasizing the independence of energy loss from this parameter. The only things that get affected from varying k are the shape of the hysteresis loop and the penetration depth.

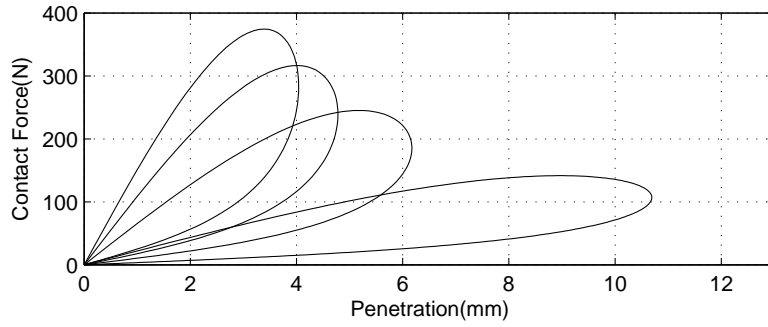
The energy loss is also directly proportional to the mass of the impacting object m . This relationship can be expressed in terms of the coefficient of restitution as:

$$\Delta E = 1/2 m (V_i^2 - V_o^2) = 1/2 m V_i^2 (1 - e^2), \quad (6.2)$$

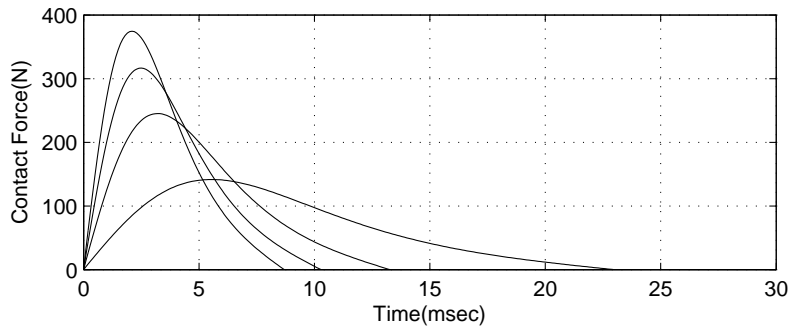
where ΔE is the energy loss, m is the mass of the moving object, V_i and V_o are the velocities of the moving object before and after contact, and e is the coefficient of restitution.



(a) Contact force vs. penetration depth for $\alpha = 0.01, 0.2, 0.4, \text{ and } 0.6 \text{ s/m}$ (inner to outer loop), where $v_i = 2 \text{ m/s}$, and $k = 50 \text{ kN/m}$



(b) Contact force vs. penetration depth for $k = 10, 30, 50, \text{ and } 70 \text{ kN/m}$ (bottom to top), where $v_i = 2 \text{ m/s}$, and $\alpha = 0.4 \text{ s/m}$



(c) Contact force vs. time for $k = 10, 30, 50, \text{ and } 70 \text{ kN/m}$ (bottom to top), where $v_i = 2 \text{ m/s}$, and $\alpha = 0.4 \text{ s/m}$

Figure 6.4: Impact response characteristics of the nonlinear spring-damper model (energy dissipation is the area under the curves)

Concluding Remarks

From our discussions in this section, using the nonlinear spring-damper model as the ground, effectively damps out the oscillations of the foot due to collision. Moreover, the greater the values of α , the greater the damping would be.

6.2 Control

In Chapter 3, we presented the basics of the monopod controller. To achieve the control objectives presented there, basically discrete time control was used, where the sampling time for discretization was set to 1 msec. To design a controller for the hopper, we started by designing a controller for a monopod hopper with ideal actuators. Using ideal actuators means that the outputs of the controller are pure torque signals to the robot model, where there are no restrictions on the joint speeds or on torque values, and above that, torque is instantly applied to the joints. By decoupling the control task into three parts, i.e., vertical height control, horizontal velocity control, and body attitude control, we were able to successfully stabilize the hopping motion for different forward velocities and hopping heights. The structure of the controller is given in Figure 6.5. The calculations for the controller were done in an *m-file*, shown as the *Main Controller*. In each time step, the input to this block includes three sets of signals: feedback from the robot and the ground, the desired command, and a number of internal states of the controller itself from the previous time step. With the ideal actuation assumption, the outputs of the controller are two torque signals to the robot leg and hip joints, whereas for the robot with DC motors as actuators, the outputs of the controller are voltages to the hip and leg DC motors. For both of these motors, position control is used.

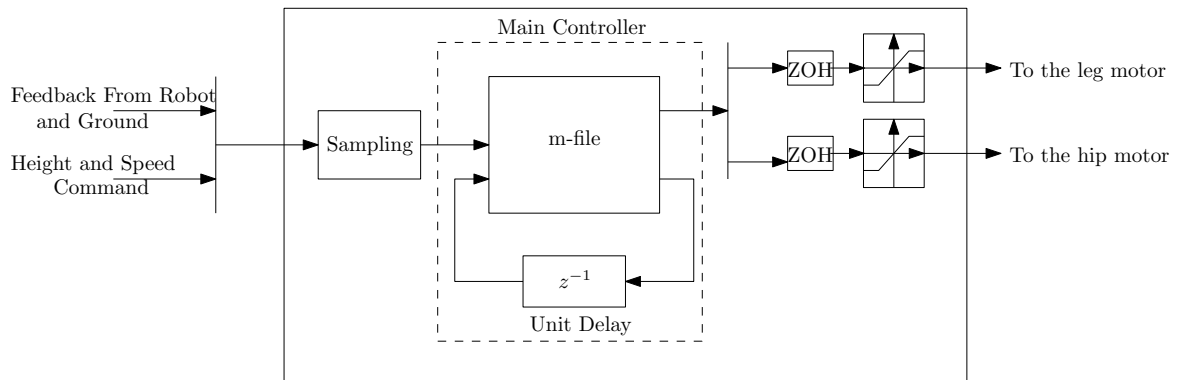


Figure 6.5: Structure of the controller

For the leg DC motor, during the thrust phase, the position set point is the required

length for the leg spring compression, and during all other phases, this set point is set to zero to maintain the ball screw's position at zero and prevent it from back-driving. An observation with the leg spring is the oscillation of the leg after lift off which is not a desired effect. This oscillation dampens partly due to the parallel damper and partly by short circuiting the motor in the flight phase which causes the motor to act as a brake.

As for the hip motor, different position set points are used depending on the hopping state. Since the hip motor is responsible for foot placement during the flight phase, and for body attitude control in the stance phase, while in the flight phase the position set point is calculated from the foot placement algorithm, and during the compression and thrust phases of the stance, this reference position is set to a value to keep the body in an upright position. The final results for height and velocity tracking are given in Figures 6.6 and 6.7 respectively. There are two things that are clearly observed from Figure 6.7. Firstly, there are spikes in the velocity just at the beginning of the flight phase, which are especially prominent for higher velocities. This is due to the step change in the hip motor command signal, which results in the kicking of the leg forward in the flight phase for foot placement. For lower velocities, where the leg is not extended as much, this is less severe. Using a smooth trajectory instead of a step input will solve this problem. Secondly, there is a steady state error in velocities other than $1 \frac{\text{m}}{\text{sec}}$, which is the one velocity for which the controller gains were tuned. As mentioned in Chapter 3, this is the nature of Raibert's hopping controller, and can be remedied by an adaptive control algorithm.

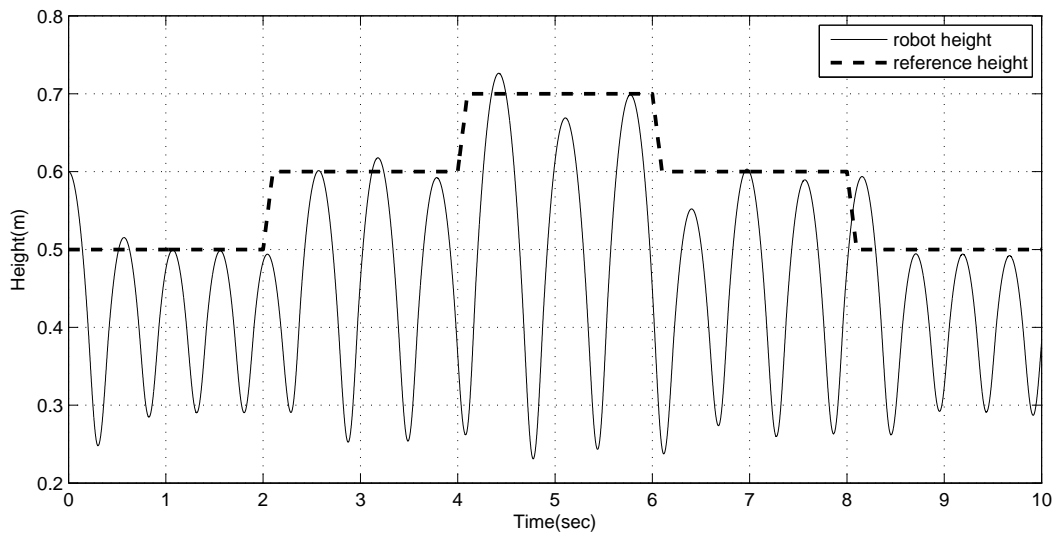


Figure 6.6: Height tracking of the monopod hopper

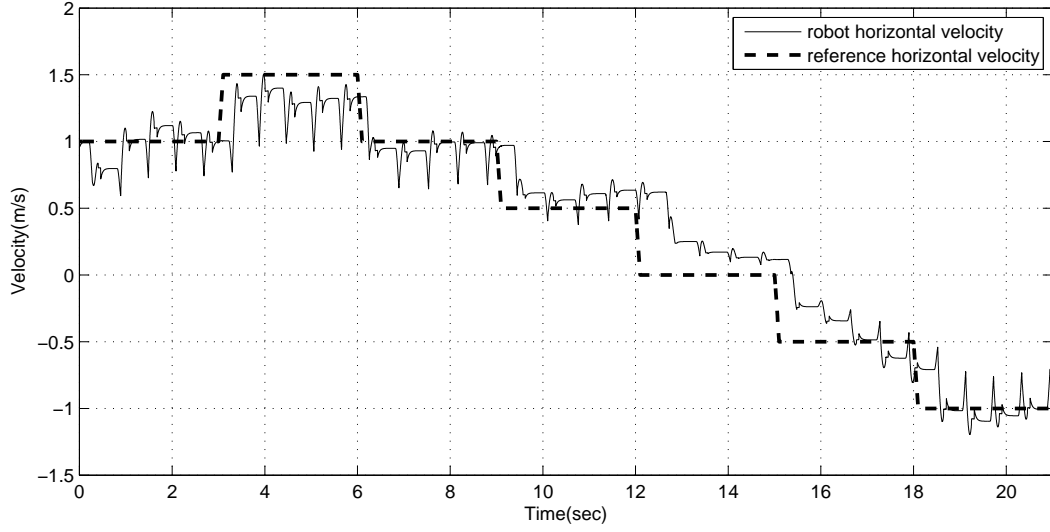


Figure 6.7: Velocity tracking of the monopod hopper

6.3 Energetics

Energy efficiency is one of the main requirements for any kind of mobile robot. It enhances robot range, capacity to perform primary tasks, and independence. Moreover, efficiency is of the essence when we are limited by the power source which is integrated into the robot structure. For a hopping robot, in each hopping cycle, energy is dissipated in different forms: actuation losses, losses due to damping and friction in the contact, losses due to friction in the moving parts of the system, losses due to accelerating the unsprung part of the leg, and so on. In this section, an analysis of the energetic aspects of the monopod hopper is given and possible improvements for increasing the efficiency are suggested.

Actuator Energy Dissipation

In a robot with electric motors as actuators, power consumption is regarded as the input electrical power to the motors. The instantaneous input power can be calculated from the following formula:

$$P_{in}^{inst} = VI \quad (6.3)$$

where V is the voltage provided by the voltage source and I is the current passing through it. If a negative value for P is obtained, it is implied that power is being generated. However, regenerative power was not considered in our work, so the electric input power was chosen to be:

$$P_{in}^{inst} = \max(VI, 0) \quad (6.4)$$

Another measure of vehicle performance is the mechanical output power, which is related to the input power through motor efficiency. Motor efficiency is in turn a function of a motor's generated torque and speed. The mechanical output power is often used to select an appropriate electric motors for a robot actuator. The instantaneous power output of the motor can be found from the following formula:

$$P_{out}^{inst} = \tau\omega, \quad (6.5)$$

where τ is the generated torque by the motor and ω is the rotational speed of the rotor. This instantaneous power can be positive or negative depending on which quadrant the motor is operating in (motoring vs. braking quadrants). Regenerative mechanical power, i.e., regenerative braking which can be used during the breaking quadrant to regenerate energy, is not considered in our work and it has been assumed that the mechanical output power is simply obtained as:

$$P_{out}^{inst} = |\tau\omega|. \quad (6.6)$$

The electrical input energy or the mechanical output energy for any hopping period $[T, T + t]$ can be easily calculated by integrating the actuators input or output power in that time period, i.e.,

$$E(t) = \int_T^{T+t} P(t) dt. \quad (6.7)$$

From Equation (6.7) the mechanical output energy and the electrical input energy for the stance phase, flight phase, and the overall hopping cycle, for both the leg and the hip actuators were calculated and are shown in Figures 6.9(a) to 6.9(f). The data correspond to a horizontal speed of $1 \frac{\text{m}}{\text{sec}}$ and a hopping height of 0.6 m. Table 6.1 shows the efficiency values obtained for the leg and hip motors from these figures.

	Efficiency
Leg motor	62%
Hip motor	58%

Table 6.1: Motor efficiency results for a hopping height of 0.6 m and a horizontal speed of $1 \frac{\text{m}}{\text{sec}}$

An observation from the figures is that the mechanical output energy is very low compared to the electrical input energy, which shows that a lot of input energy is dissipated and does not perform any mechanical work. This is due to the DC motors operating at very low speeds for which the motor efficiencies are very low. Figure 6.8 shows typical *torque/power/efficiency vs. speed* curves for a generic DC motor. If these curves are investigated for the hopper leg motor, it reveals that during some periods of the hopping cycle,

the motor is operating at low power regions of the curve where the efficiency is also low. That is particularly true when the motor is resisting back-driving in the loading phase and its speed is close to zero and the output torque has a high value. Because the rotor is not moving, this force does not perform any mechanical work and energy is dissipated in the form of resistive losses in the motor windings.

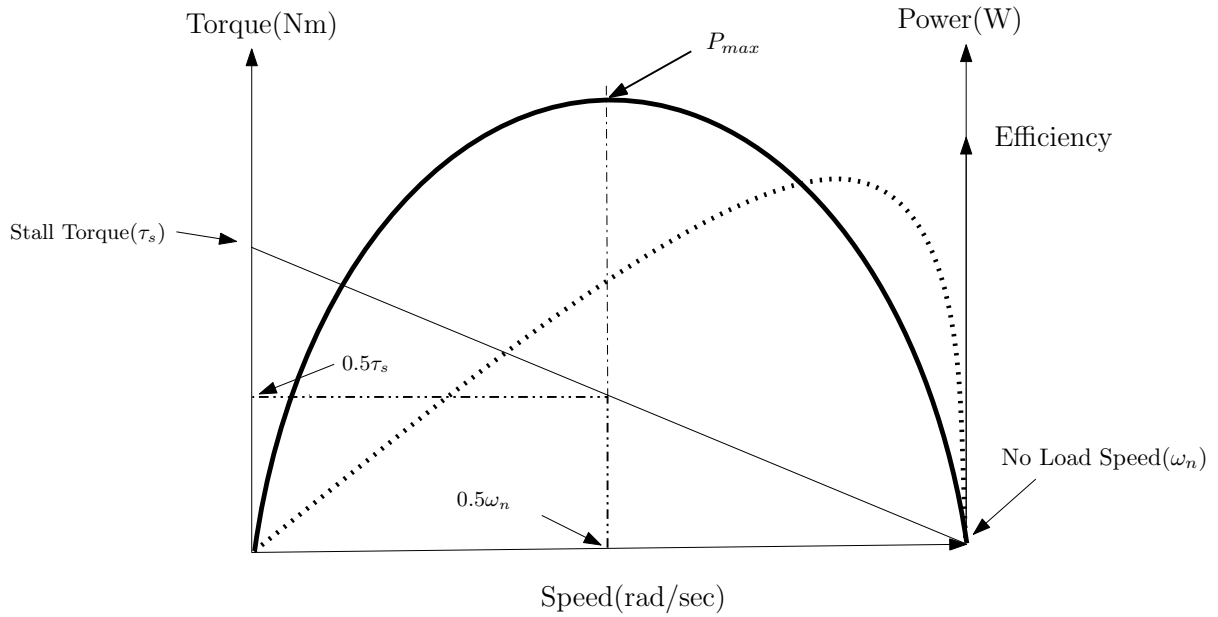


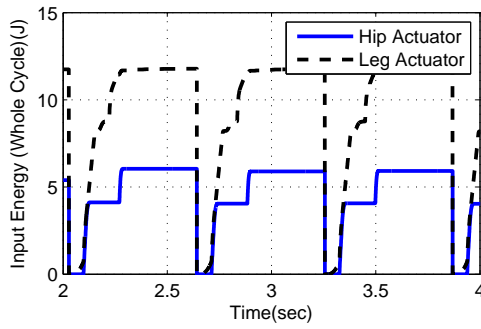
Figure 6.8: Motor torque (thin line), power (thick line), and efficiency (dotted line) vs. rotor speed

Energy Efficiency Measures

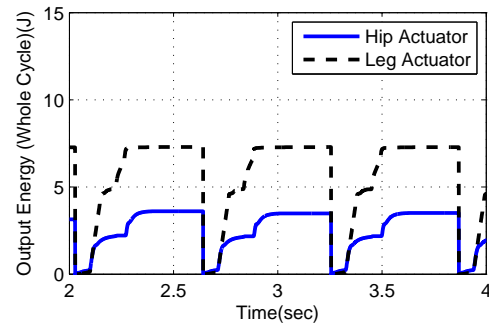
To make a comparison between our hopper and other vehicles from an energy economy point of view, in this section, two measures of energy efficiency are presented. First, which is perhaps the most objective energy efficiency measure, is *Energetic Cost of Transport (COT)* [41], a dimensionless quantity which is defined as the energy consumed to move a unit weight a unit distance, i.e.,

$$\text{COT} = \frac{E_{in}}{Mgd}, \quad (6.8)$$

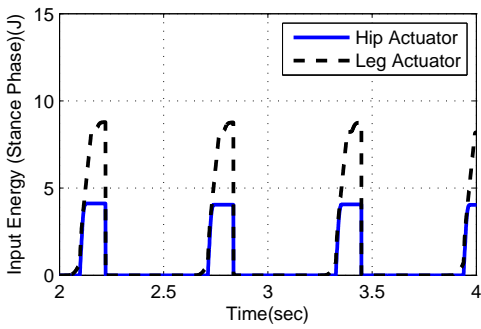
where E_{in} is the input electrical energy, d is the traveled distance, M is the total mass of the vehicle, and g is the acceleration due to gravity. For a perspective of this measure, human walking has $\text{COT} = 0.3$ [20], and the COT value for one of the most energy efficient biped robots (Cornell Efficient Biped [14]) is $\text{COT} = 0.2$. However, this robot is designed only for walking economy, and can perform no other task. If we consider versatility as well, the modern Honda ASIMO has $\text{COT} = 2.3$ [14] which is about ten times that of human walking. A comparison of minimum cost of transport as a function of body mass for a variety of



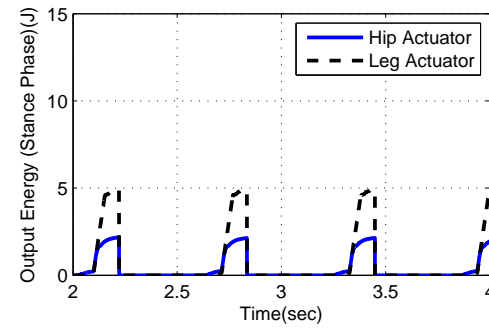
(a) Electrical input energy, Overall



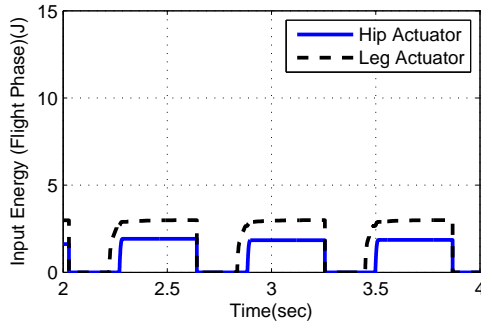
(b) Mechanical output energy, Overall



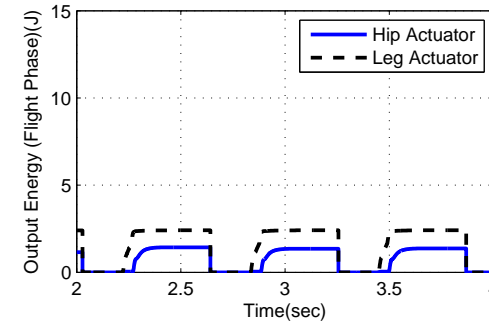
(c) Electrical input energy, Stance phase



(d) Mechanical output energy, Stance phase



(e) Electrical input energy, Flight phase



(f) Mechanical output energy, Flight phase

Figure 6.9: Electrical input energy and mechanical output energy of the hip and leg motors for configuration 1 leg (Figure 2.3) (energy is reset after each hopping cycle)

robots, animals, and vehicles is given in Figure 6.10. As for our monopod, for the speed of $1 \frac{\text{m}}{\text{sec}}$, the COT value was calculated to be $\text{COT} = 0.5037$. For obtaining this COT value, the losses that are taken into account are:

- Losses due to the collision of the unsprung leg mass with the ground.
- Losses due to the presence of the damper in the leg, which, considering that we haven't accounted for friction losses in the moving parts of the system (e.g., the leg spring), can partly represent these losses.

- Actuation losses, which include motor viscous damping, resistive losses in the motor windings, and losses in the reduction system.

Another measure that has been used by researchers for representing energy efficiency is the *Specific Resistance*. Specific resistance is defined as (Gabrielli and von Karman [36]):

$$\epsilon = \frac{E_{out}}{Mgd}, \quad (6.9)$$

where E_{out} is the output mechanical energy, d is the traveled distance, M is the total mass of the vehicle, and g is the acceleration due to gravity. The plot of specific resistance as a function of speed is referred to as the Gabrielli-von Karman diagram. Gregorio, Ahmadi and Buehler compared the energy efficiency of their hopping robot, ARL monopod, through this measure [29]. Their obtained data are presented in Figure 6.11. As for our monopod, for the speed of $1 \frac{\text{m}}{\text{sec}}$, the specific resistance was calculated to be $\epsilon = 0.4202$.

One interesting observation is that the specific resistance decreases with increasing speed, suggesting that the robot uses energy more efficiently when it runs faster. This is attributed to the fact that the energy used for maintaining vertical motion does not have any effect on moving the robot forward. Indeed, the specific resistance of the robot goes to infinity if it hops in place. For speeds other than zero, the increase in the hip actuator output power due to increased robot speed is small compared to the overall power cost, hence implying that specific resistance decreases with increasing speed even though more energy is consumed in each hopping cycle.

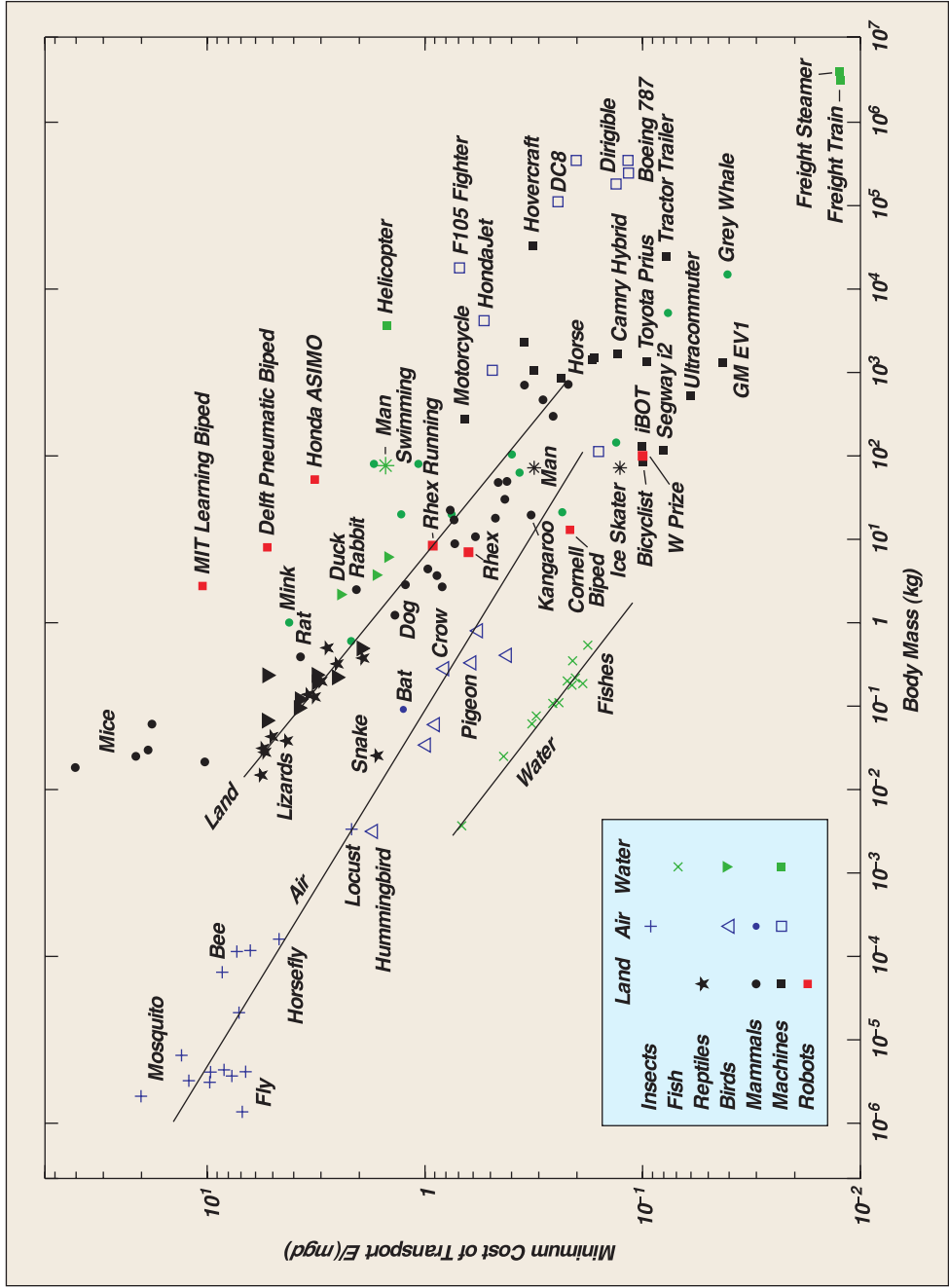


Figure 6.10: Comparison of minimum cost of transport as a function of body mass for a variety of robots, animals, and vehicles. This graph is taken from Kuo [41], where the data points are obtained from Tucker [67].

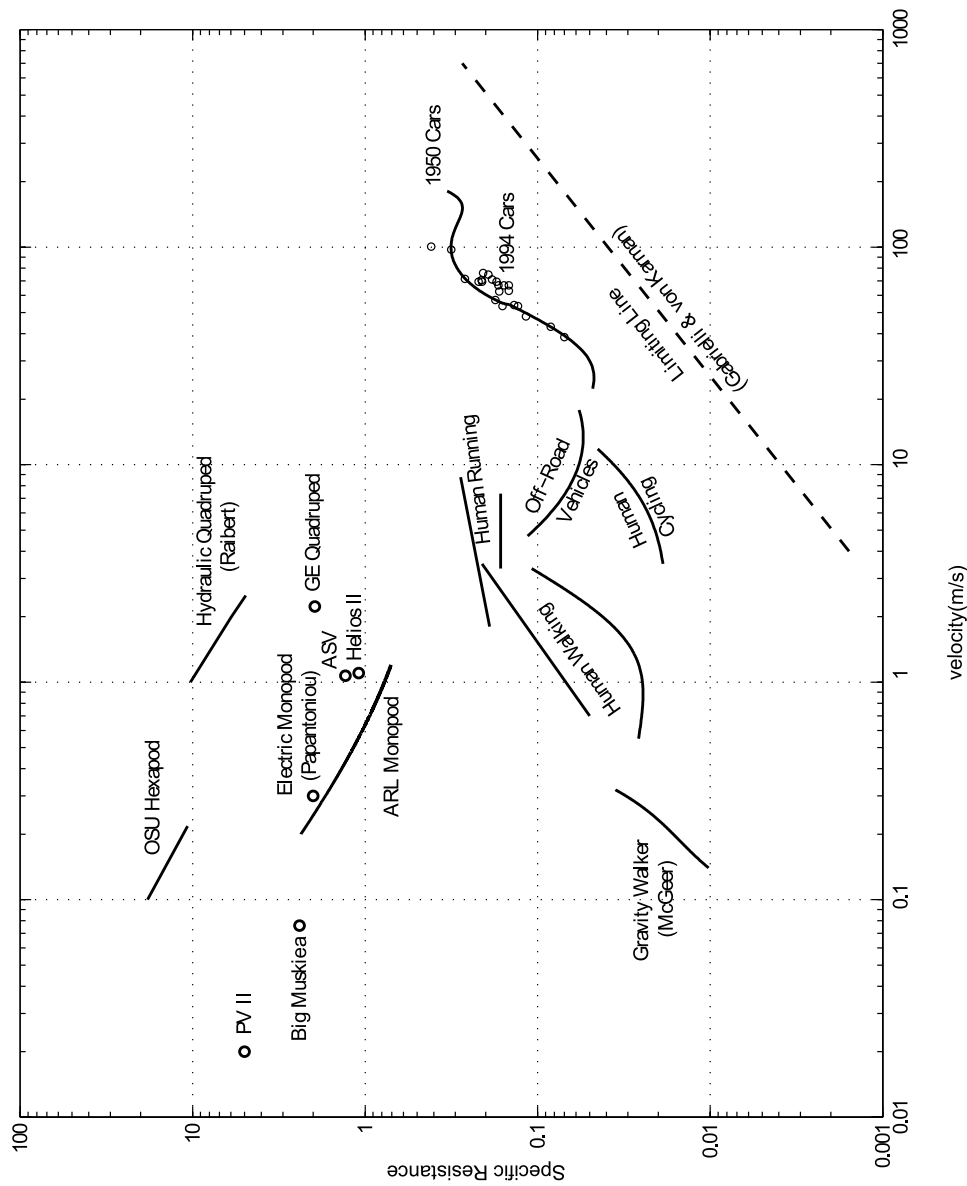


Figure 6.11: Specific resistance (ϵ) for a range of vehicles (Gabrielli-von Karman diagram). This graph is taken from Gregorio [28].

6.4 Suggested Improvements

From the Gabrielli-von Karman diagram (Figure 6.11), it is obvious that there is a lot of scope for improvement to achieve the efficiency of human walking and running. A large part of the problem lies in the actuator technology which is not suited for walking compared with the unique features of human muscles. Emergence of lightweight, compact, powerful, efficient and well-controllable actuators will play a key role in robotics problems in the future. McKibben muscles, piezoelectric actuators, and SMAs (Shape Memory Alloys) are all candidates for actuation, however they have energetic disadvantages [41]. Artificial muscle technology based on electro-active polymers [35] is another option which is currently being used for small mobile robots and micro machine applications (Figure 6.12) and may become a practical solution in the near future.

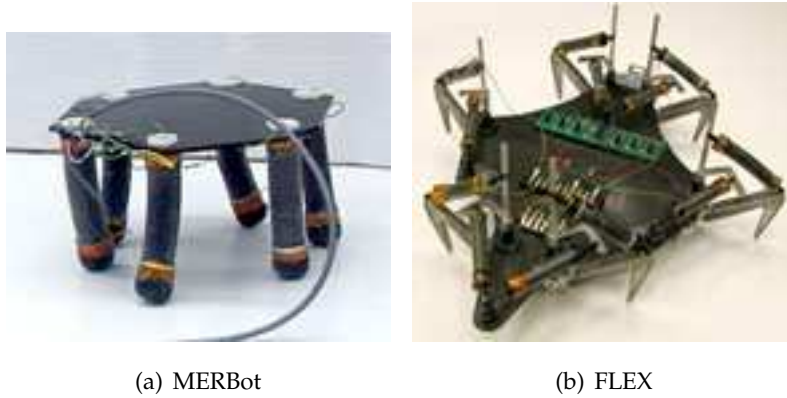


Figure 6.12: Two robots using artificial muscle technology [35]

Nevertheless, even with the current actuation technology, modifications and improvements can be made, mainly in terms of structure design and the choice of controller, to increase energy efficiency of legged robots.

Alternative Design

Parallel Spring

A practical solution, which we utilized in the leg actuation to reduce the back-driving torque on the motor, was the use of the leg configuration shown in Figure 6.13 instead of the leg configuration 1 shown in Figure 2.3 (p. 11), where a parallel spring is added to the leg actuator. In configuration 1, where only one spring is used, the back-driving force on the ball screw would be $K\Delta x$, where Δx is the compression length of the spring. Now consider the case where two springs, each with the stiffness of $\frac{K}{2}$, are used in parallel. Since the equivalent stiffness is still K , Δx would remain the same; however the back-driving force on

the motor in the second case is near $\frac{K}{2}\Delta x$. Hence adding a parallel spring reduces the back-driving torque on the motor. It also causes an increase in motor efficiency during the forward driving phase since to inject the same amount of energy, the series spring with stiffness $\frac{K}{2}$ needs to be compressed twice as much during the thrust phase. The motor will then reach higher speeds where it will operate in the more efficient quadrants of motor operation. Comparing the two plots in Figures 6.14(a) and 6.14(b) clearly shows the difference in the electrical input energy for the leg actuator when a parallel spring is added to the system. The mechanical output energy is almost the same in both configurations, hence the obtained value for specific resistance is the same for both. However, because the electrical input energy is lower for configuration 2, the obtained COT value is less and the robot is more energy efficient.

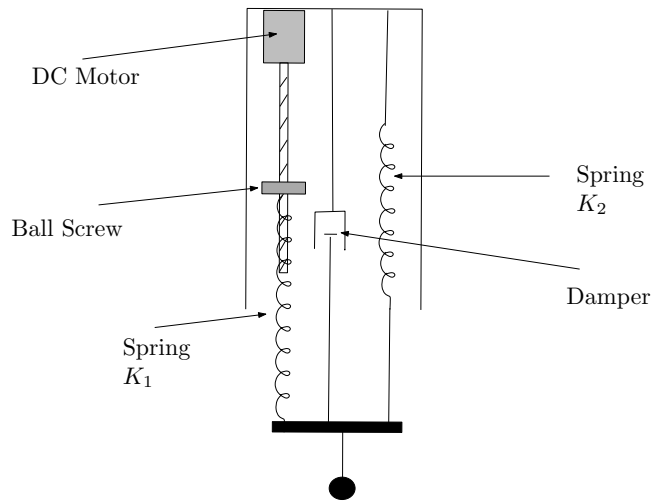
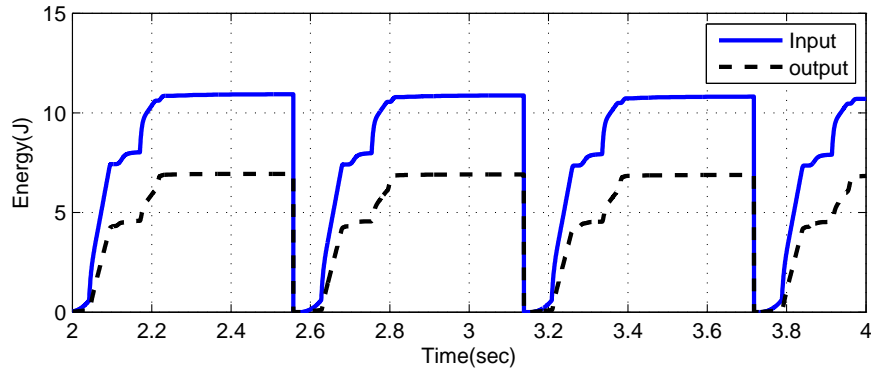


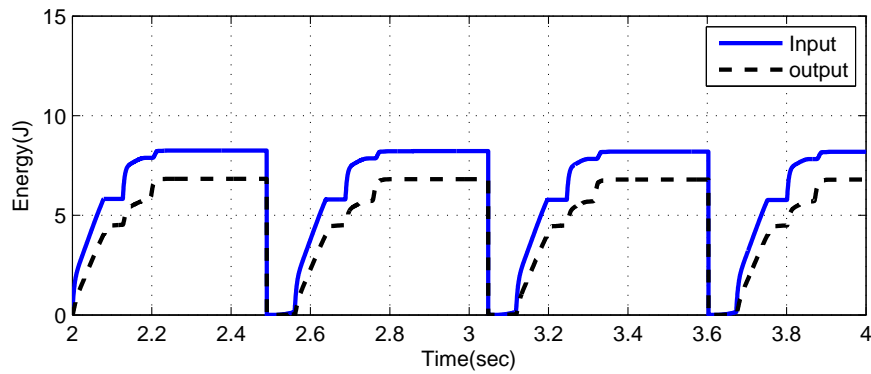
Figure 6.13: Leg configuration 2

Torsional Spring

One of the concepts that can be applied in the hardware design of the hopper to improve energy efficiency is *passive dynamics*. Passive dynamics is regarded as the unforced response of a dynamic system. An example of a physical system that uses passive dynamics for producing motion is the pendulum in a clock, which operates by cycling kinetic and gravitational potential energy. A similar concept can be exploited in walking and running robots. However, for a legged robot, purely passive walking or running is not practically viable due to the presence of energy dissipation, and stability issues. Seifert [62] was the first to propose this idea through a springy leg in a hopping machine. His hopping machine is very similar to the hopping machine in this research in that we have also made use of passive hopping behavior to produce vertical hopping motion. However, although a springy leg



(a) Input and output energy of the leg motor for configuration 1 (Figure 2.3)



(b) Input and output energy of the leg motor for configuration 2 (Figure 6.13)

Figure 6.14: Input energy and output energy of the leg motor for two leg configurations (the figures correspond to a hopping height of 0.6 in place where the energy is reset after each hopping cycle) (a) Configuration 1 (b) Configuration 2

will conserve energy for the vertical hop, it has no effect on the leg swing motion (i.e., the foot placement phase) which causes horizontal motion, and as we saw from the energy consumption plots, a large part of energy is spent on achieving this objective. So the question is whether or not it is possible to use passive dynamics to cause the leg swinging motion as well. It was McGeer who showed that completely passive walking can be achieved on a ramp under the influence of gravity alone [47]. He also showed through simulations that passive running can be achieved with the same springy leg robot mechanism with the addition of a torsional spring at the hip joint for producing the leg swing motion. Thompson and Raibert [57] showed through simulations that the addition of a torsional spring can also be applied to a monopod hopper. However, completely passive motion can't continue infinitely and it has major stability issues in the presence of small noise. Ahmadi [1] proposes a control strategy called *Controlled Passive Dynamic Running* to actively stabilize the hopper at a given speed while exploiting the passive hip swinging. According to him, this control

strategy results in dramatic energy savings for the hopper. Due to time constraint, a hip torsional spring has not been considered in this project.

Improved Control

Control is one of the areas where significant energy conservation can be achieved. The simple PD controllers used in this project are not the best solution to the control problem, most particularly they are not sensitive to energy output of the actuators nor are they effective in the presence of large parametric uncertainty in the system. Consider the hip actuator signal during the flight phase where a step of amplitude x_f is input to it at the start of the flight phase (refer to Chapter 3). A step input signal to a motor will result in high accelerations and large forces, and will increase energy expenditure as a result of high current levels. A solution for limiting the energy output of the motors is to force them to follow a smooth trajectory instead of a step input. Buehler et al. [11] have applied this technique to the ARL monopod through defining a variable which they call *phase angle*. Using this variable they map the unknown varying time interval involved in ballistic flight onto a known fixed interval. Nevertheless, we applied this technique based on the assumption that the flight period is available without the burden of calculating the flight time. The curves in Figure 6.15 show the improvements in hip motor energy consumption, if it is given a reference trajectory, compared with the case where a step input is used.

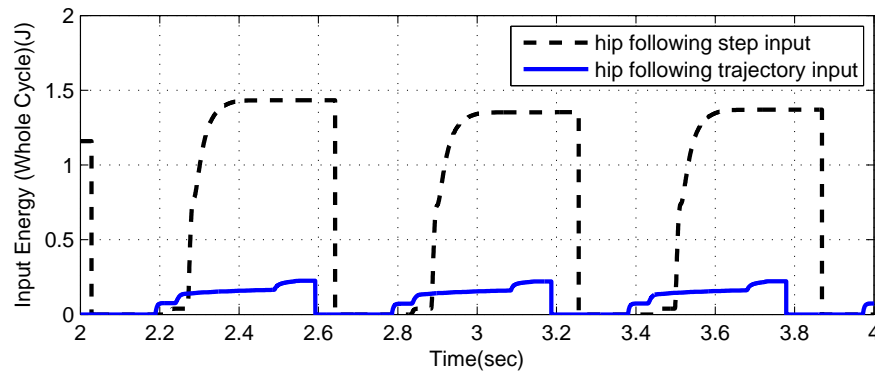


Figure 6.15: Comparison of the hip motor input energy when following a step input (dashed line) and a trajectory input (solid line) (energy is reset after each hopping cycle)

Optimal control techniques can also be applied to address the control problem for a hopping machine. These control approaches achieve a control objective by finding the optimal solution to a given control problem. However, they are generally model based and require the existence of an exact model of the system and are very sensitive to modeling errors and unmodeled dynamics.

Chapter 7

Conclusion

Legged locomotion is an attractive branch of mechanical machines, considering its versatility compared to other means of locomotion. A large part of the construction of such machines is focused on their design phase. Perhaps the most important tool for developing an accurate design are simulations. However, for such a design to be able to be realized, it is important to use models that are a true representation of reality. Indeed, many dynamics simulations are merely restricted to the computer game industry, where conforming with the reality is not relevant. Through using accurate subsystem models for a complex system, one is able to analyze, test, and understand the unpredicted behaviors of a system which are veiled within complexities. As a result of this, one will be able to apply modifications to the system at no cost and observe the consequences. Using this technique, we were indeed able to boost the energy efficiency of our robot.

One of the very useful tools for detecting errors and bugs in the design of a complex system is to visualize its motion in a virtual reality environment and observe its behavior. This is particularly true for a lay person who cannot make sense of the internal signals in the system and thus cannot compensate for the malfunctions that are present. Using visualization tools in this project greatly enhanced the detection of bugs in the system, which are cumbersome to detect through merely watching the internal signals in the system.

Energy efficiency is one of the main requirements for any kind of mobile robot. Detailed energetic analysis of the hopping robot helped us to identify the parts of the system which were consuming the most amount of energy, and select attractive candidates for energy efficiency improvements. In particular, we identified the high energy losses in configuration 1 (Figure 2.3) and suggested an alternative design based on configuration 2 (Figure 6.13). We also showed that the controller is an attractive choice for improvement by showing the gains in energy as a result of an alternative control law for the hip actuator. A very useful tool for assessing the performance of a robot is using standard efficiency measures to compare its performance with benchmarks, such as, other existing robot prototypes, humans, and

animals. In this thesis, two different measures of energy efficiency were presented and used for comparison purposes, and through them we demonstrated how well our robot model performs.

Future Work

Modular design of a system greatly enhances the possibility of adding new features and migrating to similar prototypes. Indeed, it allowed us to easily migrate from a monopod hopper to a bipedal runner without exerting much effort. The following is a list of the future improvements which can be built upon the current work to add to its sophistication and move it towards the state-of-the-art in the field:

- Including walking as well as running to more closely mimic anthropomorphic behavior.
- Migrating from 2D simulations to 3D ones.
- Using alternative controllers which are sensitive to energy consumption and are also robust to modeling errors and external disturbances in the system.
- Boosting the speed of the Simulink model by substituting the controller and dynamics calculation routines written as m-files with S-Functions. An S-Function is a computer language description of a Simulink block which can be written in a variety of languages such as MATLAB, C, and C++.
- Deploying the concept of passive dynamics for producing the robot motion. This can be achieved by choosing proper parameter values for the robot parts. In particular, for a running or walking machine, a torsional spring can be inserted in parallel with the hip motor for this purpose.
- Better structural design will always improve the versatility and energy efficiency of a robot. It can range from choice of the different robot parts to the actual configuration of the robot.
- Introducing noise in the measurements and including sensor models.

Appendix

An electronic copy of the thesis, the thesis presentation, the associated Simulink models and MATLAB m-files, along with their documentation, and animations of the monopod hopper and the bipedal runner are attached as a CD to this thesis. The Simulink files include:

- Monopod hopper
 - Ground model
 - * Various normal compliance models
 - * Various friction force models
 - Actuation model
 - * Models of brushed DC motors
 - * Ball screw and gearing models
 - * Springs and dampers
 - Robot model and dynamics calculation routines
 - Energy calculation unit
- Bipedal runner with ideal actuators
- Animation tools
 - VRML models of monopod hopper and bipedal runner
 - Virtual reality models in Simulink
 - 2D visualization function

As for the animation results, the following videos are available:

- 2D and 3D animations of monopod hopper
- 2D and 3D animations of bipedal runner

In addition to the CD, these material can be obtained from the following link:

<http://www.ptbibalan.com/projectpage.htm>

Bibliography

- [1] M. Ahmadi. *Stable control of a one legged robot exploiting passive dynamics*. PhD thesis, McGill University, 1998.
- [2] M. Ahmadi and M. Buehler. The ARL monopod II running robot: Control and energetics. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, 1999.
- [3] F. Al-Bender, V. Lampaert, and J. Swevers. A novel generic model at asperity level for dry friction force dynamics. *Tribology Letters*, 16(1):81–93, 2004.
- [4] F. Al-Bender, V. Lampaert, and J. Swevers. The generalized Maxwell-slip model: a novel model for friction Simulation and compensation. *IEEE Transactions on Automatic Control*, 50(11):1883–1887, 2005.
- [5] B. Armstrong-Hélouvry. *Control of Machines with Friction*. Kluwer Academic Pub, 1991.
- [6] B. Armstrong-Hélouvry, P. Dupont, and C. de Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138, 1994.
- [7] K. Berns. The walking machine catalogue, <http://www.walking-machines.org/>, May 2009.
- [8] P.A. Bliman and M. Sorine. Friction modeling by hysteresis operators. application to dahl, sticktion and stribeck effects. In *Proceedings of the Conference "models of Hysteresis", Trento, Italy*, 1991.
- [9] R. Brach. *Mechanical impact dynamics: rigid body collisions*. Brach Engineering, LLC, 2007.
- [10] G. Buche. Biped robot rabbit, <http://robot-rabbit.lag.ensieg.inpg.fr/English/index.php>, May 2009.
- [11] M. Buehler, D.E. Koditschek, and P.J. Kindlmann. Planning and control of robotic juggling and catching tasks. *The International Journal of Robotics Research*, 13(2):101, 1994.

- [12] S.J. Chapman. *Electric Machinery Fundamentals*. McGraw-Hill, 2004.
- [13] M. Ciavarella. The generalized cattaneo partial slip plane contact problem. II: Examples. *International Journal of Solids and Structures*, 35(18):2363–2378, 1998.
- [14] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.
- [15] J.J. Craig. *Introduction to robotics: mechanics and control*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.
- [16] P. Dahl. A solid friction model. Technical report, The Aerospace Corporation, El Segundo, CA, May 1968.
- [17] C. de Wit, H. Olsson, KJ Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, 1995.
- [18] R. Devendra. Simulink simulator for a brushless dc motor-usage manual, <http://www.mathworks.com/matlabcentral/fileexchange/5042>, May 2004.
- [19] N. Diolaiti, C. Melchiorri, and S. Stramigioli. Contact impedance estimation for robotic systems. *IEEE Transactions on Robotics*, 21(5):925–935, 2005.
- [20] J.M. Donelan, R. Kram, and A.D. Kuo. Mechanical work for step-to-step transitions is a major determinant of the metabolic cost of human walking. *Journal of Experimental Biology*, 205(23):3717–3727, 2002.
- [21] P. Dupont, V. Hayward, B. Armstrong, and F. Altpeter. Single state elastoplastic friction models. *IEEE Transactions on Automatic Control*, 47(5):787–792, 2002.
- [22] Faulhaber-Group. High power pm dc motors data sheets, <http://www.micromo.com/>, May 2009.
- [23] R. Featherstone. *Rigid Body Dynamics Algorithms*. Springer-Verlag New York, Inc., 2007.
- [24] R. Featherstone. Roy featherstone’s home page, <http://axiom.anu.edu.au/~roy/>, May 2009.
- [25] M. Gafvert. Comparisons of two dynamic friction models. In *Proceedings of the IEEE International Conference on Control Applications*, pages 386–391, 1997.
- [26] G. Gilardi and I. Sharf. Literature survey of contact dynamics modelling. *Mechanism and Machine Theory*, 37(10):1213–1239, 2002.

- [27] W. Goldsmith. *Impact: The Theory and Physical Behaviour of Colliding Solids*. Courier Dover Publications, 2001.
- [28] P. Gregorio. Design, control and energy minimization strategies for the arl monopod. Master's thesis, McGill University, 1994.
- [29] P. Gregorio, M. Ahmadi, and M. Buehler. Experiments with an electrically actuated planar hopping robot. *Lecture Notes in Control and Information Sciences*, pages 269–269, 1994.
- [30] P. Gregorio, M. Ahmadi, and M. Buehler. Design, control, and energetics of an electrically actuated legged robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 27(4):626–634, 1997.
- [31] D.A. Haessig and B. Friedland. On the modeling and simulation of friction. *ASME, Transactions, Journal of Dynamic Systems, Measurement, and Control*, 113(3):354–362, 1991.
- [32] J.J. Helferty and M. Kam. Adaptive control of a legged robot using an artificial neural network. In *IEEE International Conference on Systems Engineering*, pages 165–168, Aug 1989.
- [33] K.H. Hunt and F.R.E. Crossley. Coefficient of restitution interpreted as damping in vibroimpact. *ASME Journal of Applied Mechanics*, 42(2):440–445, 1975.
- [34] Power Trac Inc. Linear motion design guide, <http://nook.texterity.com/nook/linearmotiondesignguide/?pg=89&pm=1&ul=friend>, May 2009.
- [35] SRI International. Artificial muscle and biomimetic robots, <http://www.sri.com/robotics/epam.html>, May 2009.
- [36] T. Karman and G. Gabrielli. What price speed? specific power required for propulsion of vehicles. *Mechanical Engineering*, 72:775–781, 1950.
- [37] T. Kenjo and S. Nagamori. *Brushless motors: advanced theory and modern applications*. Sogo Electronics Press, Tokyo, 2003.
- [38] O. Khatib and B. Siciliano. *Handbook of Robotics, Chapter 16 Legged Robots*. Springer, 2008.
- [39] S.W. Kim. *Contact dynamics and force control of flexible multibody systems*. PhD thesis, McGill University, 1999.
- [40] R. Krishnan. *Electric motor drives: modeling, analysis, and control*. Prentice Hall Upper Saddle River, New Jersey, 2001.

- [41] A.D. Kuo. Choosing your steps carefully. *IEEE Robotics and Automation Magazine*, 14(2):18–29, 2007.
- [42] V. Lampaert, J. Swevers, and F. Al-Bender. Modification of the Leuven integrated friction model structure. *IEEE Transactions on Automatic Control*, 47(4):683–687, 2002.
- [43] H.M. Lankarani and P.E. Nikravesh. A contact force model with hysteresis damping for impact analysis of multibody systems. *Journal of Mechanical Design*, 112(3):369–376, 1990.
- [44] A. Lauletta. Harmonic drives, <http://www.gearproductnews.com/issues/0406/gpn.pdf>, May 2006.
- [45] C.T. Lim and W.J. Stronge. Oblique elastic-plastic impact between rough cylinders in plane strain. *International Journal of Engineering Science*, 37(1):97–122, 1999.
- [46] D.W. Marhefka and D.E. Orin. A compliant contact model with nonlinear damping for simulation of robotic systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 29(6):566–572, 1999.
- [47] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62, 1990.
- [48] J.L. Meriam, L.G. Kraige, and W.J. Palm. *Engineering mechanics: dynamics*. Wiley, 1987.
- [49] H. Olsson, K. J. Astrom, C. de Wit, M. Gafvert, and P. Lischinsky. Friction models and friction compensation. *European Journal of Control*, 4(3):176–195, 1998.
- [50] K.V. Papantoniou. Electromechanical design for an electrically powered, actively balanced one leg planar robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, volume 3, pages 1553–1560, Nov 1991.
- [51] P. Pillay and R. Krishnan. Modeling, simulation, and analysis of permanent-magnet motordrives. ii. the brushless dc motor drive. *IEEE transactions on industry applications*, 25(2):274–279, 1989.
- [52] H. Rad, P. Gregorio, and M. Buehler. Design, modeling and control of a hopping robot. In *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1778–1785, July 1993.
- [53] M.H. Raibert. Four-legged running with one-legged algorithms. In *Second International Symposium on Robotics Research*, pages 311–315, 1985.
- [54] M.H. Raibert. Legged robots. *Communications of the ACM*, 29(6):499–514, 1986.

- [55] M.H. Raibert, H.B. Brown Jr, and M. Chepponis. Experiments in balance with a 3d one-legged hopping machine. *The International Journal of Robotics Research*, 3(2):75, 1984.
- [56] M.H. Raibert and E.R. Tello. Legged robots that balance. *IEEE Expert [see also IEEE Intelligent Systems and Their Applications]*, 1(4):89–89, 1986.
- [57] M.H. Raibert and C.M. Thompson. Passive dynamic running. In *Experimental Robotics I*, 1989.
- [58] O. Reynolds. On the theory of lubrication and its application to mr. beauchamp tower's experiments, including an experimental determination of the viscosity of olive oil. *Philosophical Transactions of the Royal Society of London*, pages 157–234, 1886.
- [59] W.F. Riley and L.D. Sturges. *Engineering Mechanics*. Wiley, 1996.
- [60] A. Sato. A planar hopping robot with one actuator: Design, simulation, and experimental results. Master's thesis, McGill University, May 2005.
- [61] A. Sayyad, B. Seth, and P. Seshu. Single-legged hopping robotics research: A review. *Robotica*, 25(5):587–613, 2007.
- [62] H.S. Seifert. The lunar pogo stick. *Journal of Spacecraft and Rockets*, 4(7):941–943, 1967.
- [63] R. Stribeck. Die wesentlichen eigenschaften der gleitund rollenlager-the key qualities of sliding and roller bearings. *Zeitschrift des Vereins deutscher Ingenieure*, 46(38, 39):1342–48, 1902.
- [64] J. Swevers, F. Al-Bender, C.G. Ganseman, and T. Projogo. An integrated friction model structure with improved presliding behavior for accurate friction compensation. *IEEE Transactions on Automatic Control*, 45(4):675–686, 2000.
- [65] M. Sznaiier and M.J. Damborg. An adaptive controller for a one-legged mobile robot. *IEEE Transactions on Robotics and Automation*, 5(2):253–259, 1989.
- [66] H.D. Taghirad. Analysis, design, and control of hopping robot. Master's thesis, McGill University, April 1993.
- [67] V.A. Tucker. The energetic cost of moving about. *American Scientist*, 63(4):413–9, Jul-Aug 1975.
- [68] Carnegie Mellon University. Control tutorials for matlab, <http://www.engin.umich.edu/group/ctm/examples/motor/motor.html>, May 2009.
- [69] E.R. Westervelt, J.W. Grizzle, C. Chevallereau, J.H. Choi, and B. Morris. *Feedback control of dynamic bipedal robot locomotion*. CRC Press, 2007.

-
- [70] D.L. Wight. *A foot placement strategy for robust bipedal gait control*. PhD thesis, University of Waterloo, 2008.
- [71] P. Yedamale. *Brushless DC (BLDC) motor fundamentals*. ATMEL, 2003. Last checked: May 2009. http://www.atmel.com/dyn/resources/prod_documents/doc2596.pdf.