The Bow Leg Hopping Robot

Garth Zeglin

CMU-RI-TR-99-33

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Robotics.

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania

October 21, 1999

Abstract

The Bow Leg Hopper is a new type of running robot with an efficient, flexible leg. A one-legged planar prototype has been developed that passively stabilizes body attitude and is efficient enough to use on-board batteries. It is controlled by a real-time planner and has demonstrated crossing of simple artificial terrain including stepping stones and shallow stairs.

The machine hops using a Bow Leg, a new type of resilient, flexible leg named for its similarity to an archery bow. The Bow Leg comprises a curved leaf spring, foot, freely pivoting hip, and the Bow String that holds the leg in compression. The Bow String is used to control the leg potential energy: it may be retracted to store energy by bending the leg, held in place, and released to perform useful work. The leg is positioned using a hobby servomotor coupled to the foot with control strings. During locomotion, the machine is controlled by actuation during *flight*: the leg is positioned, and the Bow String retracted to store energy that is automatically released during stance. During ground contact all the strings become slack, and the hopper bounces passively off the ground with no forces or torques supported by actuators. The hip joint is attached to the body slightly above the center of mass so the body effectively hangs from the hip during ground contact and the natural pendulum forces passively stabilize body attitude.

In this design a single spring provides the leg structure, elasticity, and energy storage. The high forces of ground impact are carried conservatively by the spring and hip bearing. This addresses four problems central to dynamic legged locomotion: a low-power actuator may be used for thrust by storing energy in the leg; low-force actuation may be used to position the leg; the free hip minimizes body disturbance torques; and the hopping cycle is energy efficient since negative work is eliminated and the spring has high restitution. The machine is a form of "programmable mechanism" configured by leg position and stored energy during flight to control the evolution of the bounce dynamics.

The physics of the machine have been modelled in closed form using a combination of idealized analysis and empirically determined functions. These models are used by a planner that finds sequences of foot placements across known terrain to a goal position by searching a graph representing the trajectories reachable from any given landing. The planner uses heuristics to discretize the continuous control space and estimate path costs. Paths are generated in real time as needed in conjunction with a feedback controller that rejects local disturbances.

The dissertation also includes graphical methods for terrain analysis, discussion of mechanical design details, details of the real-time graph-search planner and heuristics, and experimental data from the planar prototype.

Acknowledgments

I would especially like to thank Professor Matt Mason for his perpetual patience with me through my whole tenure as a graduate student. This thesis would not have been possible without his guidance and resources, and I am grateful for the opportunity to learn so much from him.

And equally essential was Ben Brown, who has volunteered countless hours and untold energies on my behalf. I would like to thank him for agreeing to let me develop his original idea into an entire thesis of my own, and then helping me at each step of the way.

Many, many others have contributed to my effort. The weekly MLAB meeting has been a continual source of inspiration for clear thinking and academic comraderie. Professors Alfred Rizzi and Illah Nourbakhsh each offered illuminating discussions about combining control and planning. Professor Daniel Koditschek, Professor Martin Buehler, William Schwind, and Professor Andrew Ruina provided valuable commentary at conference meetings. My officemate Arthur Quaid provided much patient listening and useful comments on my thesis draft. My fellow students Chris Lee and Martin Martin listened through many discussions. And I am grateful to Dr. Marc Raibert, who brought me to this work in the first place and provided much inspiration over the years.

My work was supported in part by a National Science Foundation Graduate Fellowship. This work was also supported in part by a fellowship from the Engineering Research Program of the Office of Basic Energy Sciences at the Department of Energy.

Contents

List of Figures

Chapter 1

Introduction

As human beings we have remarkable abilities to go almost anywhere on this planet under our own power, using arms and legs, instinct and cunning. A large chapter of the history of technology is about inventing machines that augment our abilities to move ourselves and our materials. Now there is a category of machines emerging that are intended to move about the world on their own to carry out human purposes. Many of these machines roll, some fly, but a number have adopted the peculiar advantages of legs.

We have become accustomed to machines outperforming our abilities because we invent devices to overcome our limitations: cars travel much faster than we can run, airplanes fly higher than we can jump. To build autonomous machines we need to recreate some part of our own abilities—and as with most human abilities, we have little appreciation of our own talents.

But with each walk up a flight of steps we casually travel where our machines are clumsy and limited. Each time we hike up a mountain trail, tiptoeing across streams on slippery stones, hopping across fallen logs, we rediscover elegance and balance, our inherited wisdom about moving through this world.

This dissertation is about developing a new kind of legged machine that can run over uneven terrain. Its design is simple and principled. The work includes new design concepts and details of a battery-powered laboratory prototype. The scope includes new mechanisms and the mathematics and software to animate them.

1.1 Robot Legs

Legs are a viable choice for general robot locomotion because they offer agility and speed. Wheels may be faster on level ground, and wings or rotors offer the third dimension, but legs have demonstrated an agility moving about the surface of the

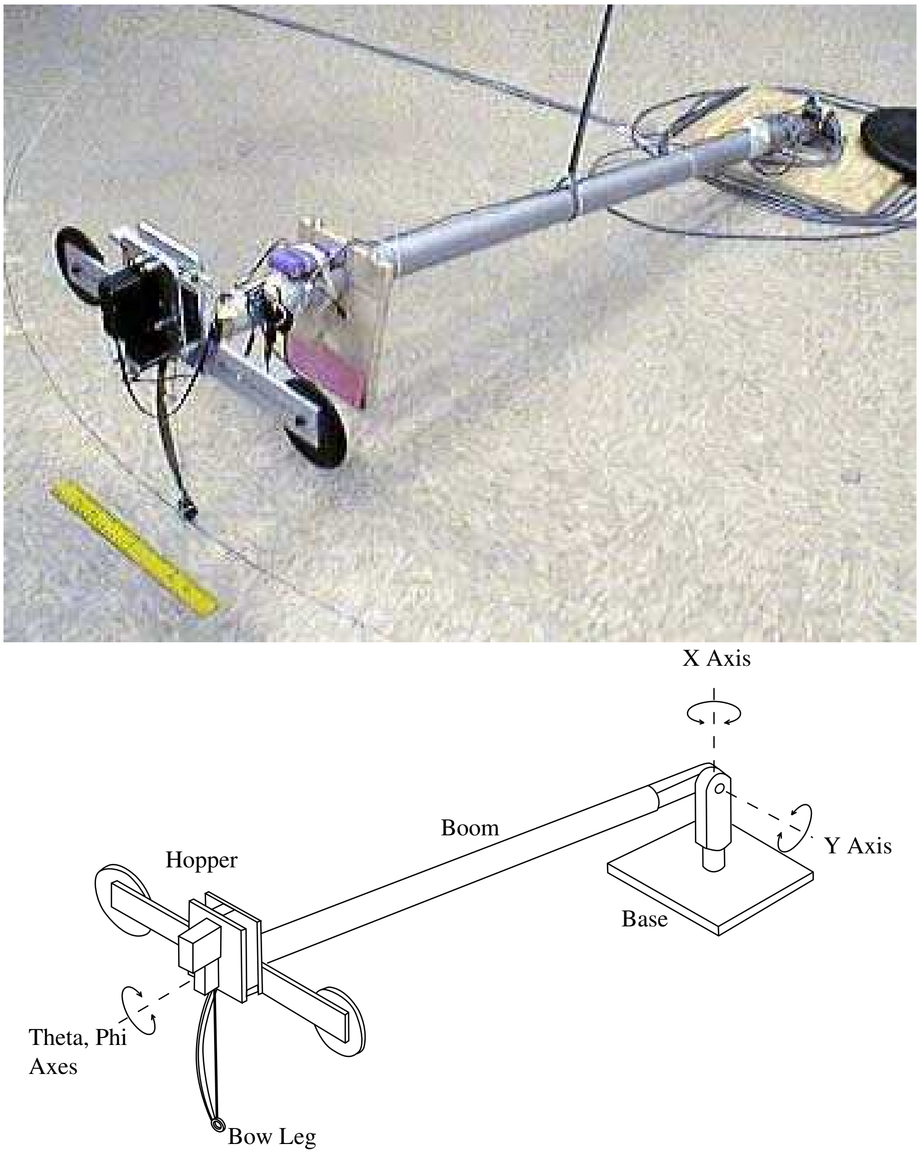


Figure 1.1: Photograph of the hopper prototype and schematic of the constraint boom. The hopper runs in a circle defined by the boom. The boom allows three degrees of freedom on the surface of a sphere: (x, y) position and θ body rotation. The leg rotates around the hip (ϕ axis) parallel to the body rotation. The boom is instrumented to measure x, y, and θ. The leg is 25 cm long and the boom radius is 1.5 m.

earth unsurpassed by any other form of locomotion. Walking machines are good for moderate speed across terrain but cannot cross holes wider than their reach. Better still are running machines because they may jump across holes and up and down steps; they are not limited by the physical span of their legs but only by their capacity to store energy.

This dissertation was motivated by the ruggedness and energy capacity of the Bow Leg, a new leg concept invented by Ben Brown and developed in conjunction with this thesis. Figures 1.1 and 1.2 depict our prototype Bow Leg Hopper. The leg is a bow-shaped spring made of laminated fiberglass. Attached to the foot is the Bow String used to store energy in the compression of the leg. A pair of control strings driven by a hobby servomotor position the leg during flight. The prototype is constrained to three degrees of freedom (DOF) by a radial boom, and is powered by a NiCd battery pack mounted near the body.

The first part of the story is concerned with incorporating the leg into a hopping machine, and the second with developing planning software to control it across terrain. Before explaining these ideas it is worthwhile to think a bit about the nature of dynamic locomotion.

Running is about flying through the air, at least until ballistic flight inevitably ends with a collision with the ground. When a ball runs, we usually say it is "bouncing"; at each collision, it compresses and takes off again if enough energy is stored and returned to vertical motion. Humans and robots generally aren't as symmetric or compliant—they require a more specialized bounce using legs. The cycle is similar, except legs can add additional energy during the collision by exerting forces against the ground. And unlike the ball, legs can change the direction of travel at will by applying ground forces asymmetrically.

The agility of running comes about for two reasons: the foot touches the ground only at isolated points, and the high forces during ground contact (stance) allow for rapid changes in velocity. This agility is possible even with only one leg—hopping is one-legged running. While two or more legs offer more variety of gaits and the ability to move legs about during flight without inducing body rotation, one leg is viable for running machines, especially a lightweight leg that can be positioned with small torques.

The ultimate goal is the design of fully autonomous running machines that can cross natural terrain. Such machines need high efficiency, good terrain perception, mechanical ruggedness, and the ability to stop and start and get up from falls. The Bow Leg Hopper does not solve all these problems, but does offer efficient and stable hopping with a simple robust mechanism.

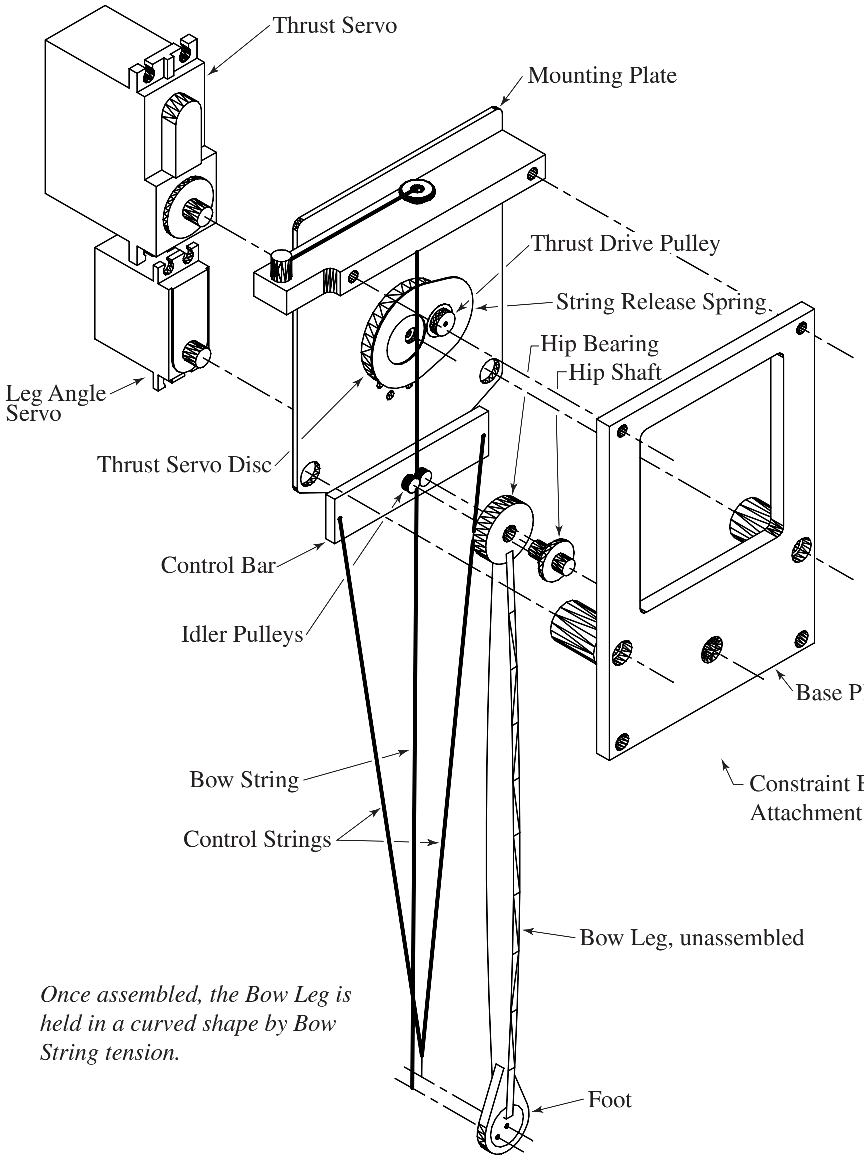


Figure 1.2: Exploded view of the Bow Leg Hopper. The top servo rotates a disk carrying the drive pulley that can engage the Bow String in order to compress the leg. The bottom servo positions the leg. The hip is an unactuated joint with a ball bearing. The ballast and several parts are omitted for clarity. The body is 4 inches wide.

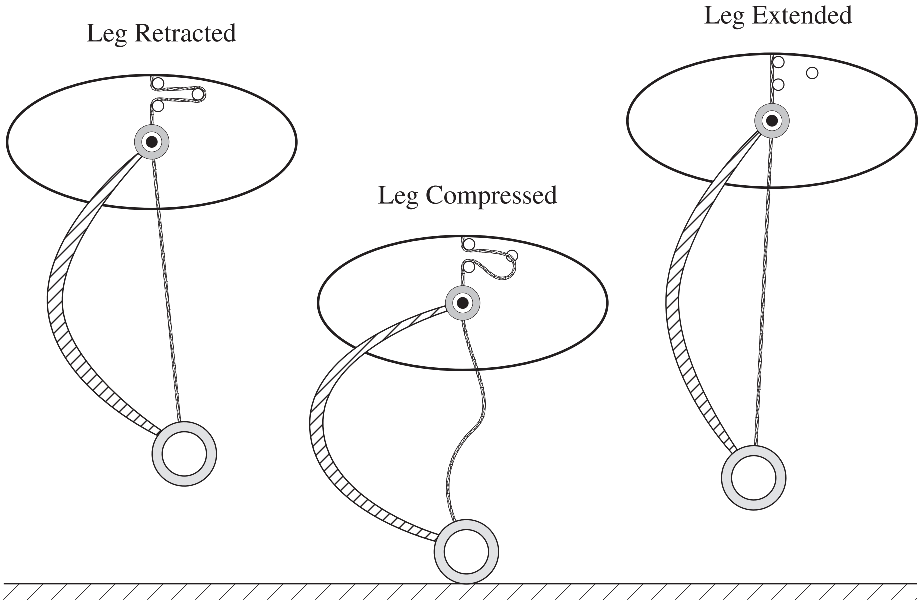


Figure 1.3: Three phases of the Bow Leg Hopper thrust mechanism. From left to right: the Bow String retracted to store energy in leg compression; the leg compressed from the impact, and the strings slack; the leg relaxed after energy is transferred from the leg to motion. Subsequently the leg is retracted again and the cycle repeats for each bounce.

1.2 Key Ideas

The starting point of this dissertation is the idea that a legged robot can mimic the essential biological aspects of running—i.e., travel using a leg to bounce off the ground—yet be designed to best take advantage of mechanical components. For example, muscles pack amazing force and energy capacity and are difficult to mimic mechanically. Our solution is to take the actuators out of the load path. The key insight is that if energy storage and control take place during flight, then the motors never carry the weight of the machine and can be low-power. This is a core principle of the Bow Leg, which is described in the following section in its most general form.

1.2.1 Bow Leg

The Bow Leg comprises a curved leaf spring, foot, freely pivoting hip, and a string that holds the leg in compression. The name comes from the resemblance to an archers bow. The function of the Bow String is to control the potential energy in the leg. It may be retracted during flight to store energy in leg compression and released

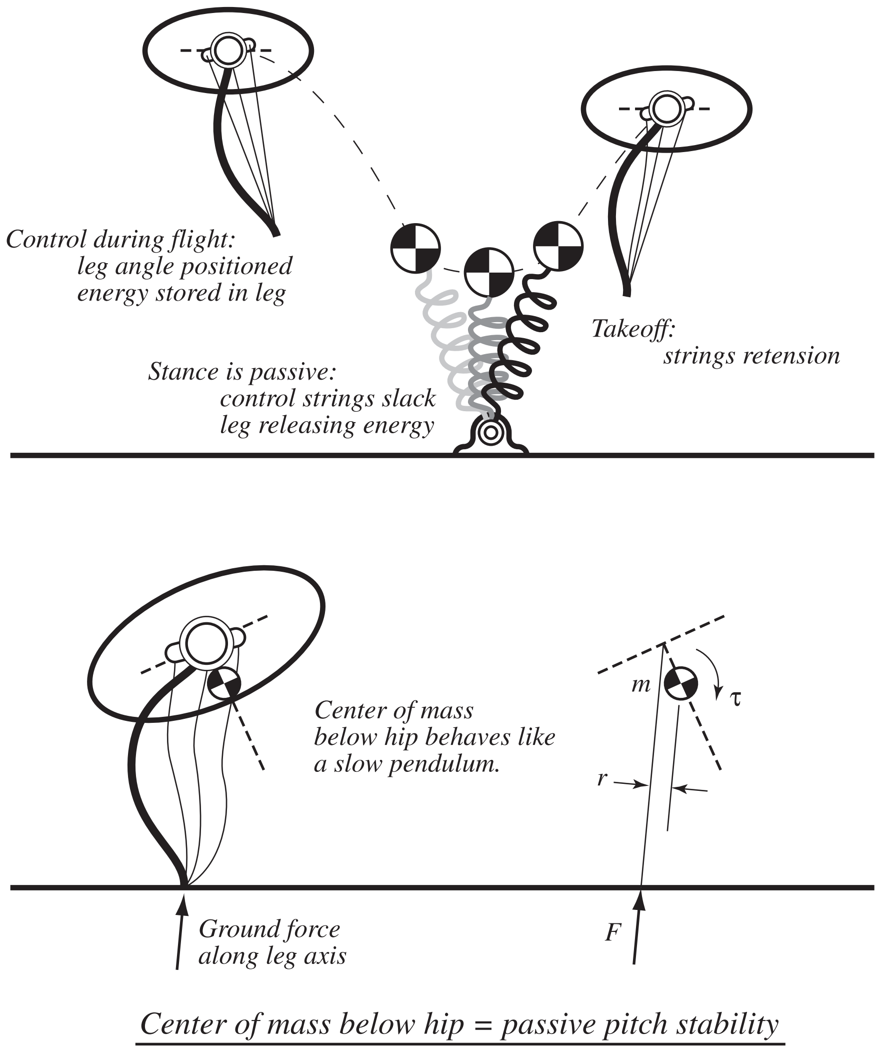


Figure 1.4: Illustration of the passive physics of the hopper. The top represents that the hopper is a passive spring–mass system when in contact with the ground. The bottom figure illustrates how the placement of the body mass beneath the hip allows the body to be passively stabilized like a pendulum.

to full length upon impact to release that energy into body motion. The leg pivots freely at the hip to minimize body disturbance torque.

The basic principle of the Bow Leg is that a single spring can provide the functions of leg structure, elasticity, and energy accumulation. A single spring can be very lightweight to minimize the losses associated with sweeping the leg. The absence of other joints means that the high forces of ground impact are carried by the mechanical components—the spring and the hip bearing—and no forces or torques need to be supported by actuators during stance.

The leg is controlled with strings that become slack as the leg compresses with the ground force. As a result, all energy input to the leg occurs during flight and the stance is passive. Tension elements are good for this role because they guarantee the decoupling of the actuators from the leg during stance. They are low-mass and may be accelerated without significant force. In the simplest case, the Bow String can be used for both energy input and position control by exerting lateral forces on the string to position the leg and axial forces to tension it. Alternatively, an additional harness of position control strings may attach to the foot.

These principles address the typical losses of legged systems: negative work and leg sweep. Negative work occurs in articulated legs when an actuator applies force in the direction opposite its motion and thus absorbs energy from the system [Ale90] [Rui91]. This can waste a significant amount of energy, but negative work can be eliminated by design by avoiding articulation. Leg sweep losses result from the need to accelerate the foot to match the ground speed and are kept small by using a very low-inertia leg. Also, this solution is compatible with the passive sweep oscillation solution proposed by [AB97].

1.2.2 Bow Leg Hopper

This thesis is concerned with the application of the Bow Leg to hopping. The Bow Leg could conceivably be used on machines with multiple legs and walking modes, but a one-legged machine is a simpler mechanism that retains much of the speed and dexterity of running. This discussion presents the most general ideas; the actual prototypes do not implement every possibility.

A fundamental problem for any locomoting machine is the regulation of body orientation and rotational velocity. Extreme body tilt can cause the joint limits to interfere with leg positioning as well as interfere with sensing and payload. Even a moderate angular velocity can integrate during flight to a body orientation that prevents a safe landing. However, the largest torques that upset body attitude generally come from the leg itself.

A fundamental principle of the Bow Leg Hopper design is to minimize body attitude disturbances. Foremost to this goal is the freely pivoting hip: the leg rotation

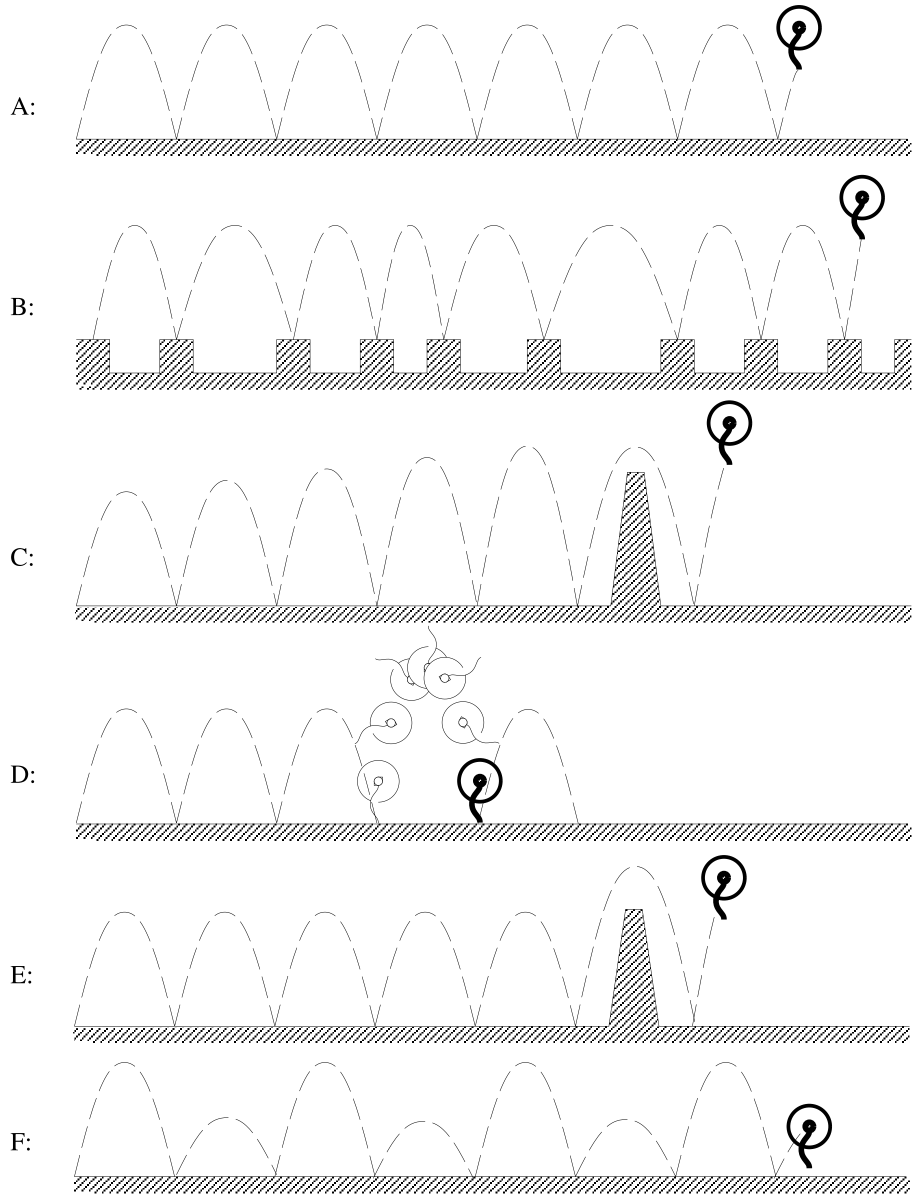


Figure 1.5: Sample hopper tasks; only items A-C were tested experimentally. A: Hopping at constant height. B: Hopping between footholds. C: Accumulating kinetic energy to cross an obstacle. D: A forward flip. E: Accumulating potential energy in the leg to cross an obstacle. F: A high/low gait created by alternately storing and releasing energy on successive steps. Items E and F require an enhanced Bow Leg with partial energy release.

is decoupled from the body during stance to preclude hip torques. The second element is a low height body design with the center of mass near the hip to minimize the torques produced by ground forces. The body attitude of the hopper prototype is passively stabilized by design by locating the center of mass slightly below the hip. Alternatively, body orientation could be actively controlled by manipulating the relative location of the hip and center of mass during flight to control the torque impulse produced by the leg during stance. This form of attitude control would involve no work since the low-mass leg would be repositioned when no forces were acting upon it.

Another basic locomotion problem is the control of kinetic energy. Energy is constantly flowing from gravitational potential to kinetic energy to leg spring potential and back. Energy must be provided to make up for losses in this cycle. In the hopper, thrust energy is transferred from a low-power actuator to body motion using the leg spring as a buffer. Energy is stored in the leg spring by retracting the Bow String during flight. Typically, flight is longer than stance—this approach allows the accumulated energy of a low power actuator to be released in a high energy burst during stance. The Bow String could also be used to capture kinetic energy by allowing only a partial leg extension during takeoff. During the impact the leg stores kinetic energy until the point of maximum leg compression. If the string were brought taut before the leg fully extended then energy would still be stored as leg potential energy. In this mode, kinetic energy would be retained as spring potential; this feature could be used to store gravitational potential from a high fall to be later released as kinetic energy.

The trajectory of the hopper is governed through control of energy and choice of foot placement. During stance, the leg exerts a large force on the body. By placing the leg far forward, the net horizontal velocity change will be negative and the hopper will slow down or move backward. Of course, the foot placement must also be compatible with the terrain—this is the principal constraint that motivates the development of a planner to control the hopper across uneven terrain.

A usual disadvantage of a one-legged machine is the need for "leg recovery," the swing of the leg during flight from the takeoff position to the next landing position. The Bow Leg is lightweight enough to be moved during flight without significantly upsetting body attitude. The swing motion does require time, however, so at low altitudes or high speeds the leg recovery time can become an operational limit. Within these bounds, this new design makes hopping feasible as a primary gait.

A typical one-legged disadvantage that is not addressed is the difficulty of controlling yaw rotation in an unconstrained "3D" hopper. The leg force is along the leg axis, which symmetry dictates should lie near the center of the body. It might be possible to displace the hip sideways to produce yaw torques but they will always be small. One solution might be to use a spherical joint at the hip so the hopper is symmetric around the vertical axis. Such a hopper could not control yaw but instead would be insensitive to it—it would bounce along, freely rotating, and still be able to change direction at will.

1.2.3 Hopper Physics

An essential aspect of the Bow Leg concept is that the natural dynamics of the mechanism produce hopping motions. Even with no power or control applied, the prototype can be dropped and will bounce several times before falling over. The Bow Leg embodies a minimalist design philosophy that emphasizes fashioning mechanical oscillators with motions that solve a task. This leads to simplicity and energy efficiency.

The Bow Leg Hopper is a simple configuration of a mass, pivot, and spring. It closely resembles an idealized hopper with a massless leg and frictionless hip, and exhibits similarly predictable physics. The ideal model assumes mechanical stability of the body attitude, a massless leg, a frictionless hip, body mass centered at the hip, and an infinitely stiff leg. Since the mechanism passively stabilizes body attitude the model may neglect body orientation. The physics of this model is very much like a stiff ball bouncing on a surface whose slope can be controlled at each impact. That is, the impact is instantaneous, the slope of the effective surface is perpendicular to the leg, and the angle of reflection is equal to the angle of incidence. With the addition of fixed dissipation in the leg and a controllable thrust energy, there are two degrees of freedom to control a planar hopper or three degrees of freedom to control an unconstrained hopper.

In practice, this model offers a reasonable approximation of the actual bounce. The chief difference is that real hoppers have a finite stance time and so the body moves during stance. This violates the pure reflection model but straightforward empirical modelling can compensate sufficiently for control purposes.

Another aspect of physics-based design is the consideration of the energy involved in the phases of locomotion. At impact, kinetic energy is transformed to potential energy in the leg in a high-power transfer, then back again at takeoff. If external work on the system is performed during the longer flight interval, the actuator power can be much lower. The Bow Leg design keeps all high power in mechanical form and uses only small electrical actuators to restore losses and provide control forces.

In the same vein is the idea of eliminating negative work. Articulated legs always have the possibility that an actuator may move in a direction opposite its force, and thus absorb energy from the system. This typically represents dissipation of energy being added by an actuator on another joint. Animals avoid this to some degree with elastic tendons that connect across multiple joints. The Bow Leg solution is to have

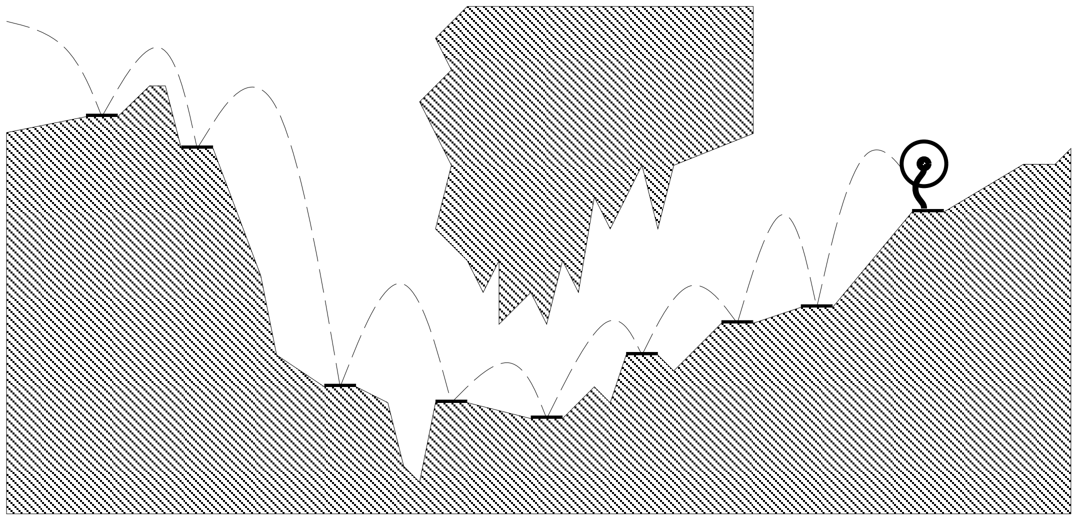


Figure 1.6: A view of the long range goal. The Bow Leg Hopper illustrated can capture impact energy in the leg and release it later. As it leaps down from the high ledge it stores kinetic energy in the leg that is released during successive hops. (The energy storing leg and terrain sensing are beyond the scope of this thesis.)

only one joint, the hip joint, which is not actuated at all during stance. It is a simple mass-spring system: any energy absorbed from the motion is stored in the leg, to be either dissipated, stored, or released.

Energy analysis also motivates control design. For example, total system energy (kinetic plus potential) may be invariant across a path comprising several bounces. The energy may be transformed from altitude (gravitational potential) to lateral velocity (kinetic energy) along the way as the hopper trades off altitude for speed and back. Energy makes a suitable control variable because it reflects the dissipation process: on each bounce a predictable fraction of energy is lost, a controlled amount may be added by actuators or stored mechanically, and so total energy may be a property of a path that varies slowly. Energy constraints may be derived from terrain geometry; energy bounds are defined by the set of paths navigating each obstacle.

The choice of energy as a control variable is especially natural for the Bow Leg because the thrust mechanism directly controls the energy added by the power source. Also, the dissipation of the leg as a spring compressing and relaxing comes close to an ideal restitution in which energy dissipated is proportional to energy stored.

1.2.4 Planning for Terrain

The ultimate goal of building legged machines is for them to move freely about the world. In large environments, getting to a location involves both route planning and controlling each foot placement to safely traverse the local terrain. This thesis is concerned with the second problem: planning a short term path across uneven terrain and using it to control the machine in the presence of errors and disturbances.

When locomotion is treated as a planning problem the emphasis is on finding solutions to a constraint problem rather than generating gaits. In general the problem is underconstrained and heuristics play an important role. The idea of a gait is transformed from a specific state machine or oscillation to a set of constraints and heuristics that define a family of hopping trajectories.

The critical constraint is making sure the foot always lands in a safe location without slipping. Of course, a safe footfall can still launch the hopper on a trajectory toward disaster, and so paths comprising multiple footfalls must be considered. Although the terrain provides geometric constraints on the foot, each foot placement is associated with an arc of body positions. This means each terrain constraint applies to a set of adjacent trajectories.

This research assumes the planner has available a geometric model of the terrain in which the regions are labelled as safe or unsafe footholds. The "unsafe" regions may be either unsafe or unknown; the controller never chooses to place the foot inside them. The geometric model is also necessary for avoiding collisions between any body part and the terrain. Terrain geometry also induces energy constraints; each terrain feature such as a hole or high point defines a minimum energy required to cross it.

The full physical state of a hopper includes the body DOF, control DOF, and the corresponding velocities. However, plans can be represented in a smaller space by defining the hopper state as the set of parameters that describe a flight trajectory; for the planar hopper, this can be just three numbers that describe the position and horizontal velocity at the apex of a parabola. Although a hopper may follow a complex trajectory during stance, ultimately it breaks contact with the ground and the center of mass follows a parabola. For this reason, a path for any hopper can be represented as a series of parabolic segments: the gap from the end of one to the beginning of the next is the distance that the hopper body moved during a stance phase. With this representation, a planner can leave the details of the stance phase up to the hardware or low-level controller, provided the planner is given a modelling function that represents the trajectory resulting from an initial trajectory and a vector of controls.

With this formulation, the control takes the form of a function specifying control parameters as a function of the flight trajectory. This function is applied once during flight to set the actuators for the next stance. If this function is a linear function, the controller is a discrete-time linear controller. If the function is a plan specifying a particular trajectory, it is defined for isolated points corresponding to the trajectories in the plan, and another control function is needed to specify actions for other states that converge back to the plan.

This research makes the assumption that the planner must operate in real time. That is, the hopper will traverse distances greater than the range of any terrain sensor without stopping, so it will need to produce plans while moving. If the plans do not cover every local state, then errors may suddenly leave the hopper far enough off the plan that it must immediately replan during a single flight phase. There are higher level strategies that might deal with this problem, such as running to safe points and staying in place while computing, but this work makes the simplifying assumption that the planner must always produce a useful result within one flight period.

Real time operation is the primary constraint on the choice of planning algorithm. The correct choice is highly dependent on the available computing hardware; with enough computing power, this problem can be solved with brute-force dynamic programming or a breadth first search backwards from the goal. This thesis demonstrates that a simple graph search with appropriately chosen heuristics can successfully cross simple laboratory terrain using current inexpensive computing hardware.

1.3 The Bow Leg Hopper Project

The previous section sketches the general ideas developed in this dissertation. Many of these ideas could be applied to any hopping robot including future Bow Leg machines. This section focusses on the specifics of the Bow Leg Hopper prototype development, including the laboratory work, planner development, and analysis.

The project has involved construction of a series of Bow Leg Hoppers constrained to three DOF. The first few operated on an inclined air table and used a Bow Leg made of piano wire, followed by several versions of a boom-mounted hopper with a laminated fiberglass leg. This progression of prototypes is addressed in more detail in Chapter 2. It is the final version of the boom-mounted hopper that is presented in the most detail. This prototype has demonstrated crossing of simple uneven terrain in the laboratory using the graph search planner.

1.3.1 Bow Leg Hopper Prototype

The boom-mounted prototype is depicted in Figure 1.1. The radial boom constrains the hopper to three DOF on the surface of a sphere that allows unlimited "forward" travel, as well as measuring the position of the hopper using sensors on the boom joints. A length of rubber tubing attached between the ceiling and the boom acts

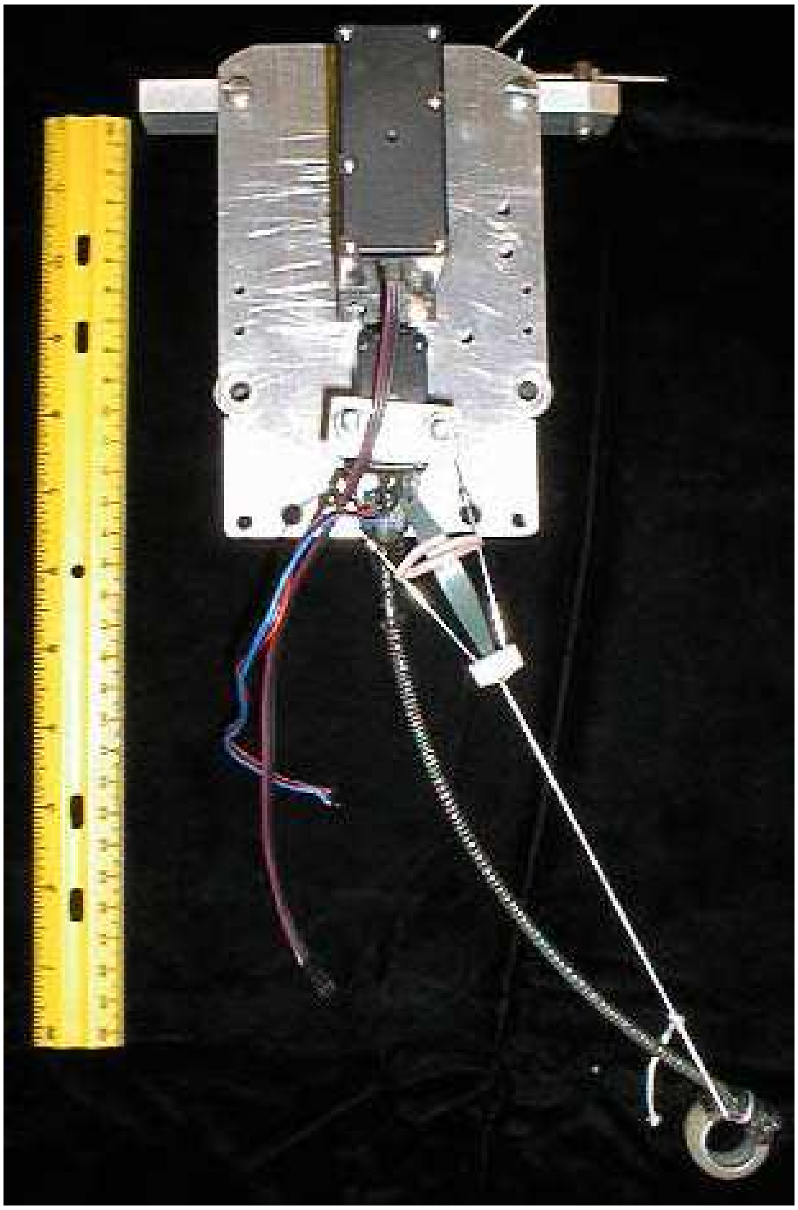


Figure 1.7: Photograph of the dismounted planar hopper with ballast weights removed. This version uses a positioning lever to move the leg by driving the Bow String.

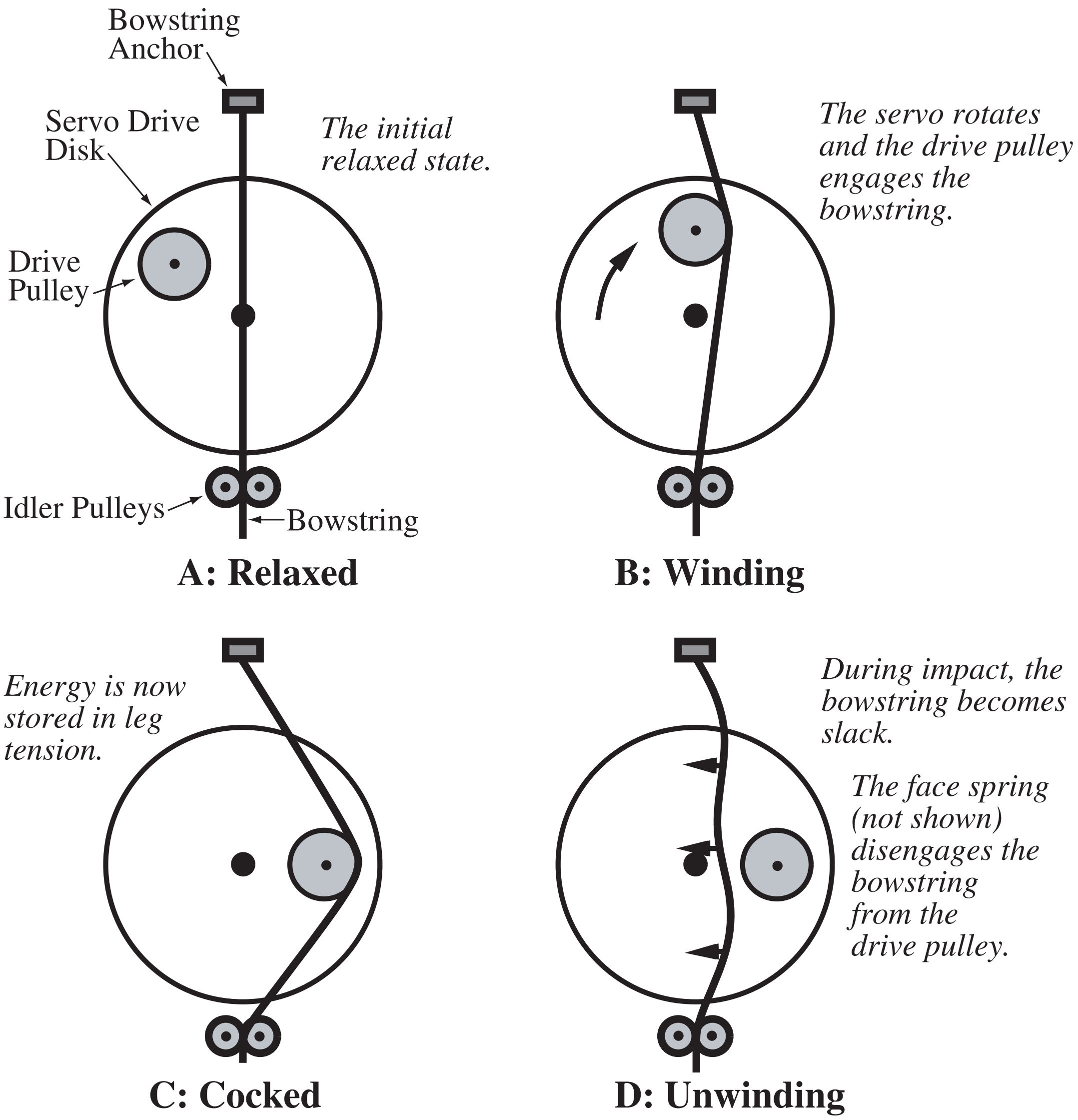


Figure 1.8: Schematic of the prototype thrust mechanism which stores energy in the leg during flight. The cycle begins in the relaxed state. During winding, the servo disk rotates, the drive pulley engages the Bow String, and the displacement of the Bow String compresses the leg (not shown). The energy stored is a function of rotation angle. During the impact, the string goes slack, the string release spring (not shown) nudges the Bow String off the pulley, and the leg extends to full length. Not shown are the servo body or the leg. The winding direction and string displacement alternate left-right.

as a gravity compensation spring to simulate running in about 30% of normal Earth gravity. This is primarily an experimental convenience that slows the hopping rate for easier observation.

The Bow Leg is constructed of laminated fiberglass, with a cylindrical foot bonded at one end and a ball bearing at the other that serves as the hip joint. It is constructed flat and is compressed into an arc by preloading the Bow String, which attaches at the foot, passes through idler pulleys on the hip axis, and then through the thrust mechanism. The leg is positioned using a pair of control strings that are driven by a hobby servomotor. The thrust mechanism tensions the string using an offset pulley driven by a second hobby servo. The two motors are powered by batteries mounted on the boom next to the hopper. The hopper is controlled by an off-board PC using off-the-shelf I/O cards.

This hopper can be powered by four on-board NiCd "sub–C" cells for about 30 minutes. The computation is off-board but is modest enough that the hopper could probably be entirely self-contained if the ballast were replaced with a commercial single-board computer and a battery for it. The leg is 25.4 cm from toe to hip; the hopper weighs about 0.5 kg and carries about 1.5 kg ballast (including the boom and batteries).

The thrust mechanism is illustrated in Figure 1.8. It operates by driving a pulley sideways into the Bow String between a ring constraint at the top and the idler pulleys at the hip. This pulls the leg taut during flight. During stance, the string goes slack as the leg compresses further; this allows the string to be nudged off the drive pulley by the "String Release Spring." With slack string released, the leg extends to its full length during takeoff and the stored energy is released. The prototype can add kinetic energy but not capture it; kinetic energy is reduced by applying a thrust smaller than the normal dissipative losses.

1.3.2 Real Time Planner

This thesis seeks to demonstrate that a planner using graph search with heuristics can compute paths across simple terrain in real time. Some general ideas for constructing hopping planners are outlined in section 1.2.4, and the prototype planner is described in detail in Chapter 5. What follows is a brief overview of a particular solution tested in this thesis.

Part of the system looks like a traditional real time controller. Position sensors are sampled at uniform intervals and the body velocity computed by an estimator. The phases of the hopping cycle are tracked by a state machine triggered by the sensor data. The leg actuator command is continuously updated to keep the leg pointed along a designated direction in world coordinates using a simple control law that negates the body pitch.

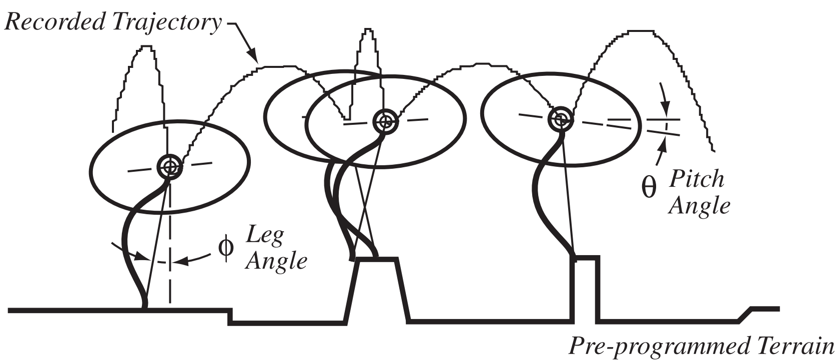


Figure 1.9: Experimental data from April 16, 1998, run 4. The center of mass trajectory is plotted at each time step and the body is illustrated at each ground impact. The wide stepping stone is a brick and the narrow one a piece of wooden 2x4 on edge.

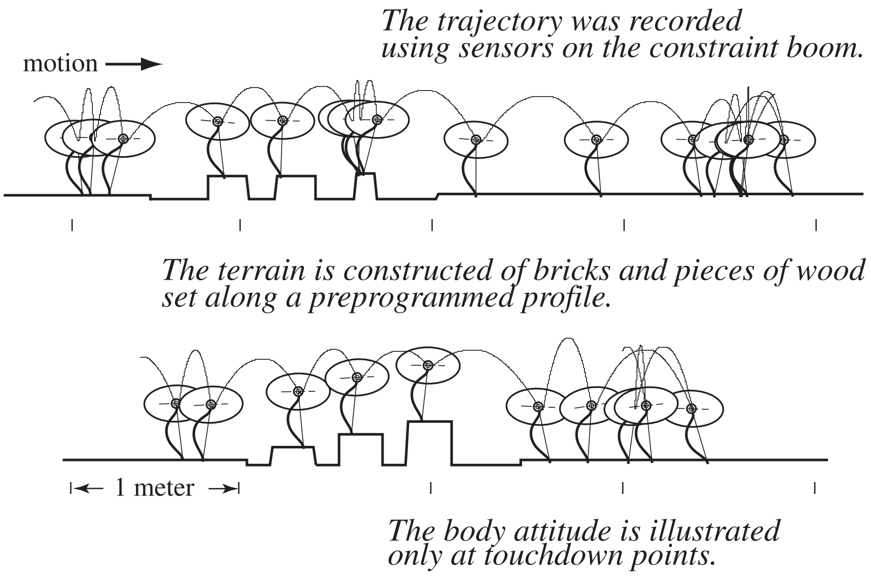


Figure 1.10: Experimental data from April 16, 1998, runs 3 and 5.

However, the interesting control occurs just once per bounce. Because the machine is passive during stance, the transition from one trajectory to another is determined by the leg angle and stored thrust energy at the moment of impact, so trajectory control is performed by selecting these parameters during flight. The "real-time" infrastructure is needed to process the continuous-time sensor information into an estimation of the flight parabola. Once the flight trajectory is accurately determined, however, a single calculation may pick the control parameters to govern the next impact. This means the linear controller that hops at constant velocity over level ground takes the especially simple form of a single proportional feedback rule evaluated once per bounce.

This work tries to blur the boundaries between traditional planning and control. When the planner is consulted for outputs once per flight, the control calculation it performs may include feedback modification of an existing plan, switching to a linear controller, or initiating a planning operation that uses graph search to find a sequence of trajectories.

Most of the discussion of the planner is concerned with that "planning operation," a graph search that tests sequences of trajectories, using heuristics to discretize the continuous control vector associated with each bounce. The planner is real time in the sense that it completes a plan within one hopping cycle. The search algorithm is a best-first search that computes path cost based on a heuristic combination of energy consumption and estimated risk. A plan is defined to be a sequence of trajectories with control parameters for the intervening impacts, followed by an optional transition to a linear controller that hops in place at the goal.

The evaluation of paths is cheap because the impact transition is computed using a closed form model based on an ideal hopper with empirically determined corrections. The terrain model is relatively cheap, involving intersection of a descent parabola and a set of connected line segments. There are a few behavior parameters that specify preferred velocity and total energy.

The laboratory experiments with the planner have focussed on crossing simple artificial "terrain" in the laboratory. The laboratory floor itself is concrete covered with asphalt tile. Configurations of bricks are stacked on the floor to form stepping stones, staircases and walls. "Holes" are simulated by designating regions in the planner as unusable foot placements.

The emphasis of the experiments is on stability and dexterity rather than speed or altitude. Sample runs are shown in Figure 1.9 and Figure 1.10.

1.4 Related Work

The Bow Leg Hopper has much in common with one-leg hoppers built by Matsuoka, Raibert, Papantoniou, Buehler, and several others. It has a rigid body and a compliant leg with an angular positioning axis and an axis that controls thrust. The tasks are to maintain body stability and travel across rough ground. Raibert style three part control [Rai86] can be applied to this hopper with only minor modification: the leg angle controls foot placement, thrust controls hopping height, and the third part, pitch control with hip torque, is not required.

There are several differences. The leg is a flexible bow-shaped spring instead of a solenoid [Mat80], telescoping spring [GAB93] [LPK93], or linkage [BD96] [Pap91]. Pitch is stabilized passively by designing the center of mass below the hip. Most importantly, the hopper is "mechanically programmed" during flight to set the initial conditions for impact. Any control must take place once per hopping cycle. High bandwidth control is eliminated along with negative work, and the controller is discrete in the sense that the machine achieves widely separated states between each control cycle.

Philosophically, our emphasis is on planning each step individually instead of controlling a steady state oscillation [AB97] [HR91]. Rather than defining specific gaits, behaviors, and transitions between them, the planner finds physically feasible trajectories that satisfy the task constraints. In principle, this planning approach could be applied to other hopper designs by treating the closed-loop control of stance as a black box. This low-level control would be modelled by a function predicting a takeoff trajectory from a landing trajectory and set of control parameters. The planner would still search out sequences of trajectories and the output would be low-level controller parameters.

A more detailed discussion of related work may be found in Chapter 7.

1.5 Discussion

The Bow Leg Hopper ideas stem from some broad mechanical principles. The most fundamental are related to energy: use mechanisms that actively direct energy between available forms, pay attention to eliminating dissipation, and use natural oscillators to incorporate conservative physics. These lead to suggestions for efficient control: keep actuator forces orthogonal to dynamic forces, either in space or in time; and use mechanical feedback wherever possible. In mechanical terms, this means using configurable mechanisms so control forces are exerted when loads are minimal and dynamic forces are supported using bearings instead of actuators.

In this thesis these principles are applied to a rethinking of one-legged hopper design by exploring them in the form of the Bow Leg. Just because the machine has adopted a function of a biological system doesn't mean it needs to copy the form. The result is development of a hopping robot using a fiberglass spring as an efficient locomotion oscillator and bearings that decouple torque disturbances from the body. The design of the body, the control system, and the terrain planner are a response to create motions that take best advantage of the Bow Leg. The consistency and simplicity of the dynamics allow for closed form physics models which greatly ease the application of planning methods.

In a sense, dynamic locomotion is a process of manipulating the ground, and so many of the mechanism and planning ideas could apply to manipulators. The hopper is an intermittent dynamic system that is abstractly similar to juggling, tapping, or batting. In fact, if the Bow Leg were simply mounted as-is on a conveyor belt, it could be used as an impulsive manipulator [Hua97] that would be triggered by contact with a part. The direction of the impulse would be determined by positioning the leg (i.e., arm) before contact. Of course, the dynamic forces would be much lower and so the stiffnesses and trigger thresholds might need to be rescaled. Some other programmable mechanisms are discussed in Appendix D.

The ultimate goal of this work is the development of fully autonomous running machines that may bound their way across rugged terrain. The Bow Leg Hopper moves us closer to this goal: it has demonstrated the efficiency and natural stability that makes self-contained running robots feasible.

1.6 Contributions

I believe this thesis makes a number of contributions:

- the use of mechanically programmable passive mechanism to exploit natural system dynamics;
- development of the Bow Leg Hopper into the first running robot with on-board battery power;
- the application of planning methods to dynamic locomotion;
- demonstration of terrain crossing tasks using planning, including a solution to the stepping stone problem;
- graphical terrain analysis techniques;
- development of practical closed-form hopper models suitable for control.

1.7 Thesis Outline

Further details are presented in the following Chapters. The document is organized as a progression from mechanical details to high level control, but may also be read as self-contained chapters:

- Chapter 2 discusses the mechanical principles of the Bow Leg and the design of the mechanism.
- Chapter 3 discusses the physics of the hopper, the development of closed-form models for control, and the calibration procedures.
- Chapter 4 discusses the abstract control formulation, some simple controllers, and graphical terrain analysis.
- Chapter 5 discusses the terrain-crossing planner.
- Chapter 6 discusses the laboratory experiments.
- Chapter 7 presents the relation to previous work and a concluding discussion.

Following the Chapters are several appendices with subsidiary details.

Chapter 2

Bow Leg Hopper Design

The Bow Leg Hopper is a new type of hopping robot with a highly resilient leg that resembles an archery bow. The guiding principle is that the body and leg form a natural spring-mass oscillator that can hop by bouncing off the ground. The design goal of the mechanism is to control the trajectory while still allowing this oscillation to proceed as freely as possible. Our solution is to treat the hopper as a "programmable mechanism" with several degrees of freedom that guide the trajectory by determining the initial conditions of the bounce. These freedoms are the position of the leg and the elastic energy stored in leg compression; they are controlled during flight and determine the change of velocity that occurs during stance.

Another principle is to create stability through careful location of the natural forces and torques. The application to the hopper is the placement of the body mass below a freely pivoting hip. The free pivot allows the leg to swing freely during stance to preclude applying disturbance torques to the body. This means the ground force always points along the axis of the leg. Placing the hip above the center of mass allows the body to "hang" from the hip much like a pendulum, so the natural forces during stance cause torques that tend to restore body attitude to a neutral position.

Much of the motivation for these principles is efficiency. The leg spring can load and unload very efficiently if no "negative work" is performed by actuators. Indirectly, shifting the control forces to flight means the actuators are low power, low force, and lightweight.

This chapter presents a discussion of how these principles affect each of the mechanical design choices embodied in the hopper. The palette of bow leg ideas includes: an efficient leg, a freely pivoting hip, storing energy during flight, performing control actions during flight, placing the center of mass for passive stability, capturing energy during impact, and controlling body torque through hip offset. Of these ideas, all except the last two are implemented in our prototype.

2.1 The Bow Leg

The Bow Leg itself is a combination of a bow-shaped spring and a string from tip to tip that may hold it in compression. The operating principle is that the spring can slowly store energy by tightening the string, quickly store energy by applying a large external force, and release energy quickly by releasing the string. It may be used as a leg or a bumper, or even for general purpose energy storage. In locomotion, the string tightening is performed by an actuator during flight, the external force is from the ground during stance, and the energy is quickly released during takeoff.

The leg needs a pivot at each end to function correctly as a compression spring. For a hopper, the round foot rolls slightly on the ground to function as one pivot, and a bearing at the hip is the other. A pivot cannot transmit a moment, so this guarantees that leg forces act along the axis of the leg and transmit no moment about the hip. The only torque on the body is due to any offset between the leg axis and the center of mass.

Some of the early air table prototypes used piano wire for the leg material. The boom-mounted prototype uses a leg constructed of laminations of unidirectional fiberglass (all fibers parallel), bonded to a rigid foot at the bottom and a ball bearing at the top. The laminations vary in width to provide a relatively constant force and uniformly distribute the strain. The leg is constructed flat, but is pulled into a shallow arc by preloading the Bow String during assembly. This leg is about 25 cm long and weighs about 30 grams.

The Bow String is a tension element that attaches to the foot and runs up through the hip centerline. Its chief purpose is to control the leg compression during flight. In the lowest energy state it holds the leg at the preloaded tension; the thrust mechanism can further tension it to store energy in the leg. A future possibility is to use the string to capture impact energy (discussed later). The Bow String can also be used as an attachment for the leg positioner; this possibility is discussed in Section 2.2.2.

The Bow String becomes slack during stance as the leg compresses under the ground force. During takeoff, it snaps tight as the leg extends to the string length. This "string collision" imposes competing requirements on the stiffness of the string material. The stiffness must be high so energy isn't stored in the string stretch; that energy is coming from the leg and would be better put into kinetic energy. However, the stiffness must not be so high that the shock loading will break it. In the boom-mounted prototype the string is a piece of 275 lb-test Spectra® brand polyethylene line. This has performed well in terms of shock loading and abrasive wear against the pulleys.

During the string collision the leg stops extending and the foot is accelerated from zero velocity during ground contact to takeoff velocity. This is an inelastic collision; the relative velocity between the body and the foot goes to zero. Ideally, the foot would have zero mass to minimize lost energy. This is balanced against the needs for foot rigidity, sufficient contact area and rounded shape to allow pivoting, ground traction, and attachment points for strings. In the prototype the foot is a plastic cylinder with a rubber traction surface.

Bow Leg Spring Design[1]

The design of a Bow Leg for a particular application depends on a number of factors, including the elastic energy storage, the force/deflection characteristics, the length, and the maximum deflection. Current implementations of the leg have been fabricated from unidirectional fiberglass composites as used in archery bows, and exhibit specific energies on the order of 1000 N-m/kg. A 100 g leg can thus store about 100 N-m of elastic energy, sufficient to lift the weight of the leg approximately 100 meters in earth gravity. If this leg were used on a hopping machine massing 1 kg total, the elastic energy of the leg could lift the whole machine (i.e., hop) about 10 meters. Similarly a 10 kg machine should be able to hop about 1 meter high based on the energy storage. For maximum energy storage, the leg must be designed to have nearly constant bending stress along its length; this can be accomplished with a constant material thickness and a width that varies from a maximum at the mid-length to theoretically zero at the tip, with an approximately sinusoidal width profile. Further improvements in performance can be obtained by using lightweight core laminates, prestressing the laminations, tailoring the stiffness and elongation characteristics of individual laminations, and other techniques well known in the composite materials industry.

The force/deflection characteristics of the leg can be affected by the laminating process and the preloading of the leg. If the leg is laminated in a straight shape (according to the thickness and width constraints described above) the compressive force is effectively that of a column in compression. The force is nearly constant, being reduced by only about 20% from the initial straight shape until the spring is bent to a 180 degree curve. In this design, it behaves nearly as a constant-force spring. If the leg is laminated to an initial curvature, the compressive force will be zero initially and increase monotonically to maximum at the maximum deflection defined by the allowable bending stress. In the limit, the leg will behave like a conventional compression spring with a fixed spring rate. The force/deflection characteristics can then be tailored by means of the initial curvature to get constant force, constant rate, or somewhere in between. One additional factor is the preload produced by the tensioned Bow String; this causes in initial a discontinuity in the force such that the applied compressive force changes from zero to the preload force with negligible

[1]Thanks to Ben Brown for these design notes.

deflection.

The designs fabricated thus far have utilized a single, unidirectional fiberglass material. However, because the bending strain varies from zero at the neutral axis (roughly the middle plane of the laminate) to maximum at the outer fiber (surface), using different laminate materials in different layers or prestressing individual laminates might produce improved energy storage, reduced weight or lowered cost. For example, a lightweight core laminate, such as wood, plastic foam or a honeycomb material, might be used for the middle laminations to reduce weight and cost without greatly reducing energy storage. Laminations of different stiffness could be used (stiffer closer to the neutral axis) such that each laminate is stressed to its limit, maximizing energy storage. Another technique is to prestress each layer such that a more beneficial stress distribution is achieved at the fully loaded state; for example, laminating the leg beam in a curved shape, then flexing it past straight and operating it with the curvature reversed can produce a more nearly constant stress profile in each laminate layer.

2.2 Hopper Design

In the following section is a discussion of specific design issues and analyses for each of the other hopper components.

2.2.1 Thrust Mechanism Design

The thrust mechanism is responsible for controlling the leg compression and is used to control the hopping energy. During flight the Bow String is retracted a variable amount to store a controlled quantity of energy in leg compression. In our hopper prototypes all stored energy is released at takeoff, but in principle the thrust mechanism could control the extended Bow String length for partial leg extension.

The current mechanism is illustrated in Figure 1.8. The Bow String is essentially pinched between the upper constraint and the lower fixed pulleys by the drive pulley that rotates sideways into the string. This action increases the length of string between the idlers; the additional length is pulled up from the leg. The release is self-timed: when the leg compresses during stance, the string becomes slack and is nudged off the drive pulley by the string release spring (made of plastic shim). The slack string allows the leg to extend to full (preloaded) length and all energy stored in it is released. The drive pulley must rotate back to the center point to re-engage the string and then may drive it the other direction for the next flight cycle.

The full drive point is at $\pi/2$ radians. At this angle the torque on the motor drops to zero as the force of the string is supported by the drive shaft. This toggling effect

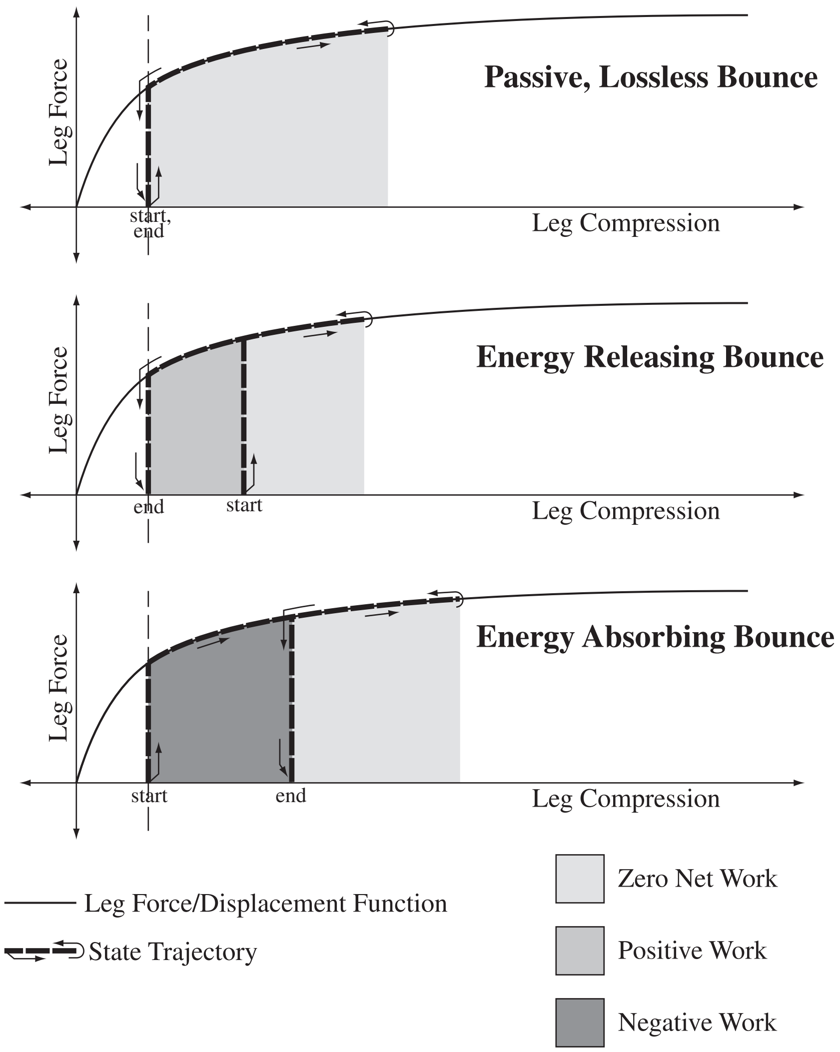


Figure 2.1: Operation of the thrust mechanism. Each case shows the work performed by the mechanism as area under the state trajectory, plotted as leg force vs. leg compression. The top case is a neutral bounce with no net work. The middle case shows energy release; the leg is initially compressed but releases to the full (preloaded) length. The bottom case shows energy storage; during takeoff the leg is clamped before extending to full length. The leg potential function illustrated is an approximation of the fiberglass leg. Our prototype can release but not absorb energy.

means that the full drive position can be held indefinitely. For smaller angles, the hobby servomotor in use as the thrust motor expends a significant power holding the pulley under constant load.

The advantages of this design are simple construction and reliable action. The string release is self timed so the only control required is a selection of a new thrust drive angle at the beginning of flight. The disadvantages are that partial wind angles consume power to hold the pulley against the string force, and that driving the pulley through $\pi/2$ radians to re-engage is wasted time.

An analysis useful for sizing the mechanism and the motors is presented in Section B.2. A result reproduced here is the ideal average motor power:

$$P_{\text{motor}} = (1-\epsilon^2)M\sqrt{\frac{g^3\Delta y}{8}} \tag{2.1}$$

The leg restitution is ϵ, gravity g, hopper mass M, and hopping altitude Δy. The thrust motor may have more time to wind during high-altitude flights, but needs to add more energy to make up for the higher losses. So not only does the thrust energy increase monotonically with hopping altitude, but so does the required motor power.

Also part of the thrust mechanism is a microswitch actuated by the taut string that is used to sense the onset of stance. The upper string constraint is mounted on a stiff cantilevered fiberglass spring that presses against a switch at the free end. As the leg impacts the ground, the string quickly goes slack and the switch opens.

The prototype initially used a thrust mechanism based on a ratchet. This had the advantages that it could hold the compression at any point with no power consumption, and that once the thrust motor compressed the leg it could be backed off to the neutral state: the leg extension reset the ratchet drum, and the motor could immediately begin winding again with no delay. The disadvantages were unreliability, fragility, and kinetic loss. The thrust pawl was not engineered sufficiently to reliably disengage, and if the leg prematurely hit the ground the thrust motor would not have backed off completely and the ratchet drum would collide with the motor rather than the limit stop. More fundamentally, the ratchet drum was accelerated to high velocity during stance only to collide inelastically with the limit stop. In other designs, only the low-mass string need move at high speed which reduces internal losses.

A design feature missing from both of these mechanisms is control of the degree of leg extension—the leg always extends to full length and releases all stored energy. With a variable extension and partial energy release, energy could be accumulated in the leg over multiple bounces, or the hopper could capture the energy stored during impact. This function would be useful to rapidly reduce hopping height or to absorb energy on descending terrains.

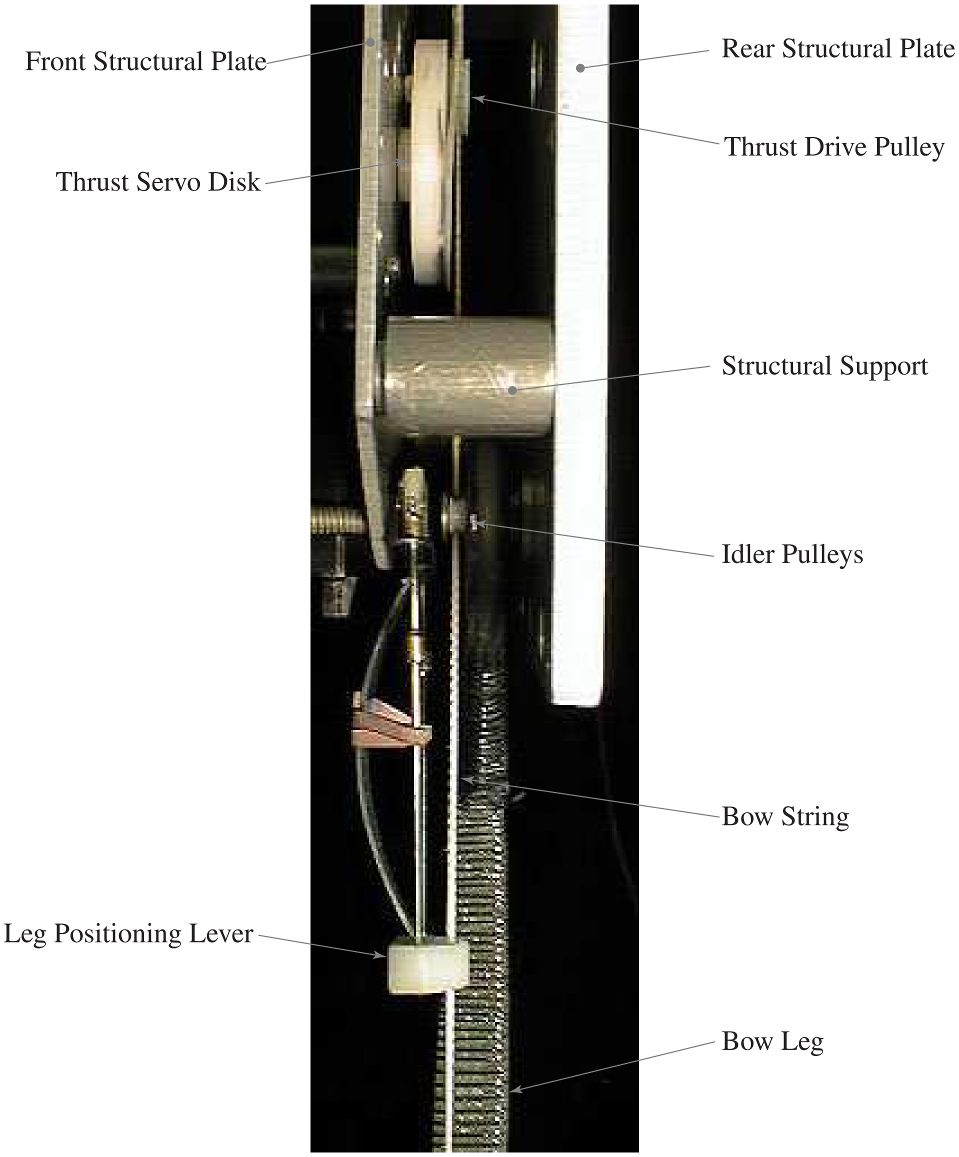


Figure 2.2: Close-up side view photograph of an early version of the boom-mounted prototype. Note the Bow String coming up from the toe through the positioning lever, through the idler pulleys, behind a support, and across the servo disk adjacent to the drive pulley. The string release spring is seen edge-on between the thrust servo disk and the Bow String. The positioning lever was later replaced by a string harness.

2.2.2 Leg Positioner Design

The function of the leg positioner is to control the angle between the body and the leg during flight. This is used to govern the angle between the leg and body velocity at the moment of impact; together with the stored energy this angle determines the trajectory the body follows during ground contact and the takeoff position and velocity.

Since the control strings go slack during stance, the positioner is automatically decoupled from the leg sweep. This has two effects. First, the positioner is detached from any high ground forces and need only apply forces commensurate with the leg inertia to move it during flight. However, the leg is also free to sweep during stance, so at takeoff there might be large force as the strings become tight and the leg jerks the positioner to the new leg position. The positioner must be designed to move quickly and precisely under small load during flight, but withstand a potentially large shock torque at each takeoff.

The current design (illustrated in Figure 1.2) uses a triangular harness of two control strings that attaches to the ends of a yoke on a hobby servomotor shaft. The strings are joined above the foot, then pass through a hole in the foot and are connected to the leg with a rubber band. The rubber band gives the harness compliance so the leg can be moved far to either side of the commanded position; at takeoff the leg extension stretches the rubber band and the leg begins to move back to the setpoint. The triangle arrangement gives the positioner some stiffness at the setpoint: the restoring force is fairly constant until both strings have tension when the leg is perpendicular to the yoke. Ideally, the two would meet right at the foot (pass separately through holes in the foot), but in practice the two are knotted a short distance above the foot. In general, the disadvantage of the control harness design is the need for two additional strings and an elastic element to be placed along the leg at risk from the sharp terrain.

An earlier design used the Bow String for positioning, as illustrated in Figure 2.2. A positioning arm with a spring-loaded torque limiter extended from the servo and moved a tube wrapped around the Bow String about halfway down. The Bow String could be pulled through the tube for thrust, and the tube moved sideways for positioning. This design had fewer parts near the ground, but was not as precise: the lateral stiffness at the commanded position is zero, so the leg was free to rattle around the desired leg angle.

It is conceivable that the leg could be moved by applying torques directly to it near the hip. This would place all fragile positioning components in the body, but would introduce a new calibration problem since the leg angle would depend on the leg curvature, which is a function of the thrust compression. Such a mechanism would also need a reliable method for detaching from the leg during stance to accommodate the hip rotation due to leg swing and compression; this decoupling occurs automatically with the string designs.

2.2.3 Body Design

The hopper body provides a structure for mounting the motors, power supply, thrust mechanism, and any other component or payload. In the boom-mounted prototype, the body houses the motors, thrust mechanism, and ballast weight, and is attached to a constraint boom. The batteries are mounted near the body on the constraint boom; the four NiCd sub-C cells are sufficient for about 30–45 minutes operating time.

The design issues for the body are mass distribution, ground clearance, rotational inertia, rigidity, and maintenance. The center of mass must be placed below the hip for passive stability, but the body must not be so low it is easily hit by the ground. The worst case for vertical clearance on level ground is a fall from a high altitude with the leg at the friction limit (far to the side) since the leg will compress a great deal and the geometry brings the body closest to the ground. The worst case for lateral clearance is climbing a step: since the hopper can easily climb or fall two to three times its own height, crossable obstacles can be taller than the body. If the body is wide, the hopper cannot approach the obstacle closely before ascending, and stairs with shallow treads are impossible.

Competing with these requirements is a desire that the body have significant rotational inertia to help passive stability. The pendulum frequency of the body hanging from the hip needs to be significantly slower than the slowest hopping rate (i.e., highest altitude hopping). The frequency is lowered by closer placement of the COM to the hip and increased rotational inertia. Along the same lines, the leg is not actually massless and so leg positioning during flight does rotate the body; higher body rotational inertia reduces the disturbance.

So the body should be compact: short for COM position, high for ground clearance, wide for inertia, but not so wide it will interfere with obstacles.

Rigidity of the body and component mounting is essential to preclude energy dissipation through relative displacements and vibrations during stance. This could become a serious issue if payloads are to be carried, since they would need to be internally rigid and securely mounted to the body.

The prototype body is a frame constructed of two parallel plates on which the motors are mounted. (See Figure 1.2.) This was easy to fabricate although replacing any individual component generally required a fair amount of disassembly. The ballast consists of stacks of steel washers fastened on the end of adjustable aluminum bars that are clamped between the body plate and the boom mount. The ballast bars extend out and down and are necessary to lower the center of mass below the hip since the body by itself is top heavy. Ideally, battery and payload mass would used to locate the COM and ballast would be unnecessary.

2.2.4 Boom Design

The prototype hopper is a "2D" hopper constrained to three degrees of freedom. The constraint is a tubular boom that extends from a universal joint mounted on the floor to a pivot at the attachment to the hopper body. The boom serves as kinematic constraint, position sensor, and gravity compensator. The universal joint allows the hopper to move in "X" and "Z" directions and the pivot allows the θ body rotation; the pivot is aligned with hip axis so the boom mass acts as an extra point mass at the hip. The kinematics are actually two degrees of freedom moving on the surface of a sphere and rotating about an axis at a fixed offset from the normal, but are otherwise treated as planar motion. The two axes in the universal joint have optical encoders, and the pivot in the end has a potentiometer. These position sensors are used for control and data collection.

The "gravity compensation" is performed by a piece of elastic rubber tubing attached between a point on the ceiling directly over the first boom joint and a point on the boom about a third of the way out from the base. The tubing acts as a spring that supports both the boom mass and some body mass. This reduces the effective gravity and lowers the hopping frequency; the prototype typically operates in about $1/3$ normal gravity, which increases the flight time by a factor of $\sqrt{3}$. This is primarily an experimental convenience to facilitate observation.

The design issues for the boom are toe tangency, gravity linearity, and boom stiffness. If the radial component of the toe velocity is zero when the hopper touches down, the toe can make firm contact without scuffing sideways. The boom is designed for this behavior by placing the base joint at ground level so the vertical tangent to the sphere that the toe traverses is at ground level. Of course, the hopper body moves vertically during stance so the body will always have some side motion during ground contact. We chose to have the hopper vertical at impact, so the theta pivot at the end of the boom is angled relative to the boom axis in order to be level at the typical impact altitude.

The effective gravity on the hopper varies as $\cos(y/\mathrm{radius})$ as it moves vertically. Using a linear spring as the gravity compensation spring helps flatten this a bit; as the hopper rises and and effective gravity diminishes, the compensation force also diminishes. A spring was chosen over a counterweight to avoid inertial effects; however, the spring does have transverse vibration modes which dissipate some small energy. In practice, both these effects are negligible and the measured flight trajectories look like parabolas with uniform acceleration.

More significant is the effect of compensation force on passive pitch stability. The hopper isn't truly in a low gravity environment, it has a constant force pulling up on the pivot as gravity pulls down on the mass. This produces a constant restoring torque around the pitch axis. It is small compared to the magnitude of the torque produced by the ground force but of much longer duration. This significantly enhances the passive pitch stability. Also, the boom pitch pivot is a ball bearing but does have damping friction that also helps stability. It is yet to be seen how passive attitude stability will perform in an untethered 3D hopper without these compensation forces.

During stance the leg force is approximately 150 N applied to the body for about 70 msec. The boom does vibrate under this shock load and dissipates energy. In general, higher boom stiffness is more efficient and desirable.

2.2.5 Electronics Design

The boom prototype is controlled by a 133 MHz Pentium PC. The interface to the robot consists of a 2 channel quadrature decoder card and a multifunction I/O card with many analog and digital inputs, some general purpose digital outputs, and several timer controlled digital outputs. The boom has a 512 line optical encoder mounted on each of the two base joints to measure XY position and a potentiometer in the end that measures body attitude. The timer outputs are used to generate two PWM position control signals for the servomotors. The sensors are sampled at 200 Hz by a software interrupt and velocities are estimated using digital filters. The servo control signal is a 200 Hz positive-going pulse train with widths between 600 and 2200 μsec. The hopper is connected to the interface through an umbilical cord that runs along the floor and out along the boom. Electrical power for the hobby servo actuators is provided by batteries on board. Also connected through this I/O interface is a lap box with four analog sliders and several momentary contact switches; this is the physical "user interface" for operator control.

The hopper control needs only one state estimation per flight so the system gets away with low-resolution encoders by fitting a parabola to 61 samples taken straddling the peak; this yields peak position, velocity, and time at resolution higher than the sampling rate or the encoder resolution.

The general lessons of this design are: use off-the-shelf interface cards; use encoders rather than potentiometers; hobby servos seem to work more smoothly at the higher-than-normal signal rates.

2.3 Specifications and Performance

The specifications for the boom-mounted prototype are as follows: leg length is 25 cm and the running circle is 1.5 m in radius. The machine mass is about 2.5

kg, including 730 g in the hopper mechanism itself, 200 g of batteries, 500 g ballast, 1150 g-wt of boom weight (measured at toe, since the boom pivot supports some of the boom weight). The leg itself weighs only 30 g excluding the hip bearing. It is noteworthy that the hopper mechanism comprises only 30% of the total mass; the batteries 8%; and the leg 1.2%. A full 60% of the mass is in the "dead weight" of the ballast and boom.

The effective gravity is generally about 0.35 G (3.4 $\mathrm{m/s^2}$). Bouncing passively, the hopper loses only about 20-30% of its energy each hop. However, the thrust servo is sized a little too small to maintain altitude in full gravity. The machine has hopped as high as 50 cm; 80 cm is theoretically possible based on leg elastic energy capacity, with the present machine mass and reduced gravity.

The highest running speed has been a little higher than a meter per second, although high speed has not been the focus of the work. The passive pitch stabilization effectively damps pitch oscillations up to about 0.5 radians amplitude; the chief limit is leg sweep travel, since at steep body pitch the joint limits are rotated until the leg can no longer be placed near the ground surface normal, and the hopper loses control.

Energy consumption is surprisingly low: the machine runs for 45 minutes on a single charge (approximately 5 W-hour) of the four sub-C cell nickel cadmium batteries. Most of the power is consumed by the thrust motor. The worst efficiency comes from holding the thrust motor at partial wind; the servo can consume 5 W staying in place. With full winds the holding power is close to zero.

The first Bow Leg logged between approximately 5 and 10 hours of operation before beginning to delaminate, at which point the efficiency decreased and it was replaced. The hopper body has had no major mechanical problems.

The precision of the hopper motion is limited by the inaccuracies and uncertainties in the flight and stance models, and also by the precision of actuator control. The motion is very sensitive to errors in the leg angle at touchdown: for example, a 0.04 radian error in leg angle (1.0 cm lateral error in foot position) translates to a 17 cm error in lateral position at the next touchdown, based on typical hopping conditions (0.3 m hopping height and 0.2 m/s forward speed).

Some re-engineering would likely improve basic performance. If the ballast weights were replaced with batteries, the hopping time would increase by at least a factor of three without any other mechanical changes or increased mass. A lower mass foot would decrease collision losses at touchdown and liftoff. Other refinements could include higher precision sensing of body attitude and leg position, more rigid positioning drives, and a more precise positioning servo.

2.4 Early Prototypes

The boom-mounted hopper is actually about version six. For those interested in project history, here's a brief review of the earlier versions, all designed and built by Ben Brown.

The first four hoppers were designed to float on an inclined air hockey table which provided a low-friction air bearing. The effective gravity was about 10%, directed toward the "ground surface," a piece of plywood resting on the low side of the table. Each hopper was constructed on a flat fiberglass disc, with a Bow Leg made of piano wire. The first two were passive test beds, one just to see if the leg would bounce, and one with a manually set one-shot thrust mechanism (a bead on a Bow String, supported by a hook) to see if the leg compression would relax the string. The next prototype had a leg positioner but no thrust; the Bow String was simply attached to the end of a control arm mounted on a hobby servo shaft. The servo was wired to an off-board receiver and the machine could be remotely controlled for a number of bounces until it lost too much energy. The fourth air table prototype had an offset pulley thrust mechanism not unlike the final version, except the timing was controlled with a relay and two switches so it would fully wind after each bounce.

Prototype three and four were both flown with human remote control and a linear controller. For the computer, a colored rectangular target was attached to the top and the position was tracked by an overhead vision system. The controller was a simple proportional position feedback that controlled the leg angle. The vision system was very noisy and produced poor orientation estimates, so leg positioning wasn't very accurate. The system wasn't engineered much beyond a marginally stable demonstration of hopping.

Although these prototypes carried their batteries on board, the servo control signals were sent along a tether of several 30 gauge wires. With the low gravity and low mass of these machines, the tether was a significant disturbance force, or if placed in the correct place, stabilizing force. Overall, the rotational damping was very low and they were difficult to control, but they did display an encouraging degree of passive pitch stability.

A difficulty of the air table was that the hopper needed to be low-mass and low-profile in order to float without excessive out-of-plane rotation. The decision was made to move to a boom, and version five was a test bed for the new fiberglass leg that consisted of a leg mounted on a piece of wood, attached to the end of a metal pipe with a simple universal joint. This motivated design of the present boom-mounted prototype. Various components of version six have been replaced, but the basic frame is the same.

2.5 Discussion

The ultimate goal of this line of research is the development of fully autonomous running machines that can cross rugged, natural and man-made terrains. While walking machines are bounded by their kinematic limits, running machines are bounded only by dynamic limits. A high strength composite spring can have a specific energy of 100 meters or more, thus a machine having 5% of its mass in the leg could theoretically hop 5 meters vertically or more. Parabolic flight means the widest crossable hole would be about twice that distance.

The Bow Leg has the efficiency and low power requirements that make self-contained, electrically-powered machines feasible. The design takes advantage of the dynamics to separate the motor forces from the impact event by displacing them in time, and enables the hopper to handle high mechanical power without requiring high control power. The high energy storage capacity of the leg may permit vertical and horizontal hopping distances on the order of meters, allowing mobility on very rugged terrain. While the current research focuses on single-leg machines, the Bow Leg is equally applicable to multi-leg machines. We anticipate that Bow Leg hopping and running machines will be capable of practical operation on real terrains, including small footholds spaced irregularly and separated by large horizontal and vertical distances.

The key to the Bow Leg idea is the freely pivoting hip. It is literally the idea about which everything else revolves. It rejects body disturbance torques by design, performing well enough to be the sole mechanism stabilizing body attitude in the planar prototype. This allows the leg to sweep freely and the leg to bend freely, generally eliminating degrees of freedom that must be controlled. Of course it wouldn't work nearly so well if the leg weren't low-mass to begin with or didn't automatically decouple from the actuators. But the selective use of uncontrolled joints in locomoting machines beyond simple pivoting toes suggests that robotic balance may be as much a mechanical creation as a control phenomenon.

In all fairness, there are advantages to designs that place the center of mass above the hip, namely that the low and wide machines suffer from a lack of payload space. In the upright design the freely pivoting hip still has merit for reducing leg to body coupling, but an additional inverted pendulum problem must be addressed. The Bow Leg itself though is still admirably well-suited: high efficiency, high capacity, direct energy control, well-defined force vectors, rugged materials, and low mass are suited to solve the problems of efficiently sweeping the leg, rapidly changing direction, and carrying an energy supply.

The hip design effectively solves the pitch problem for the prototype. It also greatly simplifies physical modelling and the control of the machine since the dynamic behavior is quite repeatable and predictable. This predictability stems primarily from the passivity—a mass-spring system is easy to model—but also from the efficiency, since the sources of loss are concentrated and consistent.

The managerial comment is that adopting different constraints can lead to interesting solutions. The project got started because there was an air table and hobby servos available with which to build a hopping robot. The constraints of low mass and low power led Ben Brown to the idea of the Bow Leg.

Another idea explored is the programmable mechanism. It is an idea familiar to designers of low-cost devices like answering machines, in which complex cams are frequently use to multiplex one motor to several tasks, using very small actuators to guide the state at decision points. But the same notion can be applied to much more dynamic systems like fast-moving running robots or manipulators. The general scheme might be similar to something like the Bow Leg: build a mechanical oscillator (body and leg) whose state can be governed at key points (impacts) using small actuators (e.g. leg positioner) with forces orthogonal to the loads, either in space or in time. Use mechanically triggered freedoms (e.g. thrust pulley) to deliver forces at precise times. Tension elements (e.g. Bow String) are very useful as "inequalities" to decouple forces. This is not a formalization but rather a suggestion of how to apply these ideas in general robotics practice.

2.6 Future Work

The basic future work is improvement of the designs outlined in this chapter. More precise leg positioning could lead to substantial improvements in repeatability. A more powerful thrust motor could operate continuously in full gravity. Adding the hip translation axis would allow direct control of body attitude and gymnastic moves.

The most exciting direction in which future work may move is the development of unconstrained "3D" Bow Leg Hoppers. Some notes specifically related to the mechanical design may be found in Appendix C. Conceptually, the Bow Leg idea seems to generalize well to 3D. The pivoting hip becomes a 2-axis pivoting hip, the Bow Leg needs a foot that handles 2-axis roll but otherwise is the same, and the thrust mechanism is exactly the same. Many system problems such as sensing the body position and orientation become much harder, and practical machines will need some form of local terrain sensing.

The passive pitch stability of the planar prototypes has been a striking simplification over previous legged machines. There are reasons to doubt it will be as successful in 3D however because the frictional damping of boom pitch bearing will be absent. Our closest examples are the air table machines in which the passive stability was apparent but less reliable. The clearest strategy for augmenting the passive stabilization is to introduce a translation stage that positions the body with respect to the hip. This can be another mechanical reconfiguration that occurs during flight for no energy cost, but which determines the lever arm for the ground force to produce a controlled torque impulse on the body. This would enable active control for body torques in a fashion very similar to the control of body forces. It would also provide a means for transfer of energy between rotation and translation, which might be useful for dramatic body rotations such as are involved in gymnastic motions.

We expect that lateral velocity control will decompose readily into two independent processes, much like the Raibert machines [Rai86]. Controlling yaw, however, is difficult with a single, small foot that cannot generate substantial torques. One intriguing possibility is the construction of a machine symmetric about the vertical axis. Although animals have a sagittal plane and a well-defined "front", this machine would have a symmetric hip and no preferred direction: it would bounce along any desired direction in the world, freely rotating about the vertical, yet changing course at will. Practically speaking, many designs do have a preferred travel direction and some other possibilities are as follows: using multiple legs; direct torquing with a yaw actuator and oversized foot; coordination of eccentric stepping and thrust control to generate yaw impulses; and creating stabilizing torques with momentum wheels or gyroscopes.

Chapter 3

Bow Leg Physics

The Bow Leg Hopper is a locomoting machine designed for predictable behavior. For most purposes, the continuous time dynamics of the machine may be represented by a discrete time dynamic model that predicts the result of ground contact using closed-form functions. This is possible because the state transition from one trajectory to the next is described by the passive physics of the bounce, as governed by the initial conditions set by the mechanical control freedoms.

The first idealized models presented are intended to illustrate the hopper physics. These form the basis for empirical models used by the planner for predicting motion and by the controller for interpreting the continuous time sensors and computing control feedback. The essential processes that are modelled are the impact against the ground and the flight phase. The machine is designed so the internal processes may be largely neglected, e.g. string forces, actuator motion, and structural dynamics. The bulk of the modelling is concerned with impact. The discrete model is further divided into an abstract model used for planning and a mechanical model used to translate plans into the actual hardware controls.

This chapter is divided into several analyses. First is considered the ideal hopper with a movable hip and instantaneous stance. Next is the impact model of the idealized hopper without movable hip, followed by a practical model of the actual hopper, with discussion of the calibration procedure. This discussion is generally independent of the controller in use; a specific discussion of controller state appears in Section 4.1.

3.1 Ideal Hopper Model

This section presents an analysis of the impact physics of an ideal Bow Leg Hopper. The result is a set of closed-form functions that map the impact velocity to the takeoff

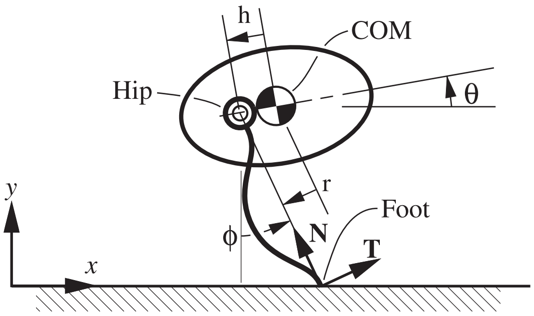


Figure 3.1: Schematic of an idealized planar hopper. The leg is a massless spring with perfect restitution. The frictionless hip pivot can move distance h along an axis during flight and clamp rigidly during stance. Body attitude is θ, measured with respect to the world x axis, and is illustrated with a positive value. Leg angle is ϕ, measured with respect to the world $-y$ axis, also illustrated positive. Gravity is assumed to point in the $-y$ direction. The center of mass position is defined by (x, y), measured from the world origin. The hip offset h and ground force moment arm r illustrated have negative values. Not illustrated is ξ, the energy stored in the thrust mechanism. The impact is analyzed in the leg frame $[\,\hat{\mathbf{T}} \quad \hat{\mathbf{N}}\,]$.

velocity given three mechanical control parameters.

The model differs from the actual prototype by assuming that the hopper has a hip translation control axis and that the leg is infinitely stiff. The Bow Leg mechanism is designed to approximate an instantaneous bounce, so this idealization is not just a illustrative example, it forms the core of a usable model. The actual machine does have significant stance time, which is taken up later in the practical analysis.

3.1.1 Preliminaries

The body of the hopper is a rigid structure attached to a Bow Leg with a frictionless pin joint at the hip. The assumptions are that the leg is massless, the foot connects to the ground with a pin joint during stance, and the ground is rigid. The leg is assumed to be stiff enough that impact is instantaneous, which means the body orientation and leg angle remain constant during the infinitesimally long stance.

It is assumed that the leg position is actuated during flight but moves freely during the impact. The hip is assumed to be mounted on a lead screw that positions the hip along an axis on the body during flight and locks rigidly in position during stance. The leg also assumes a mechanism to add energy on each bounce.

This machine hops like an elastic bouncing ball. It falls, impacts, reverses vertical direction, flies, and falls again. Kinetic energy is stored in the leg during the impact

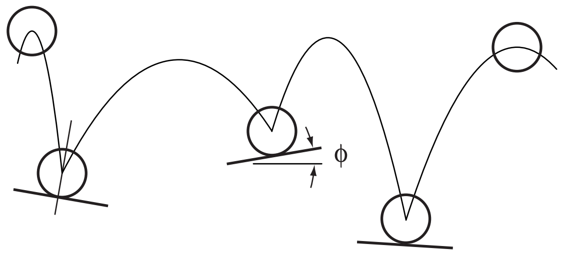


Figure 3.2: The hopper moves much like a bouncing ball; the hopper leg angle ϕ is analogous to the ground slope ϕ for a ball.

and released during takeoff.

Unlike a ball, the hopper has a leg. The leg angle is analogous to the ground slope under the ball, and the angle of restitution is proportional to the angle of impact when measured relative to the leg. The hopper controls its speed and direction on each step by positioning the leg before touchdown. The takeoff trajectory is determined by the impact velocity, the leg position, the energy loss from leg dissipation, and the energy gain from the leg energy mechanism.

Figure 3.1 illustrates the coordinate conventions, and Appendix appendix-symbols lists all the symbols. The center of mass position is described by the vector $[\,x \quad y\,]^{\mathrm{T}}$ and the body angle with respect to the world by θ. The leg angle ϕ is measured with respect to the $-\mathbf{y}$ axis in the world frame. The angles θ and ϕ follow the right hand rule in the plane.

The leg is massless with a pin joint at the hip and an effective pin joint where the foot makes point contact with the ground. With no inertia, the free body equilibrium dictates that the force applied to the pin joints must lie along the axis of the leg and be of equal magnitude and opposite direction. The force from the ground to the body therefore is directed along the leg axis. The force on the body is the sum of gravity and the leg force. Since the hip transmits no moment the only body torque is due to leg force applied along an offset from the center of mass, denoted as the moment arm r.

The leg coordinate frame $[\,\hat{\mathbf{T}} \quad \hat{\mathbf{N}}\,]$ is defined at the point of contact and specifies the tangent and normal directions of the leg at impact. The ground force is in the direction $\hat{\mathbf{N}}$. The spring has restitution ϵ that defines the ratio of impulse released to impulse absorbed. The value of ϵ is assumed to be constant.

Since the pivots at the foot and hip guarantee that the leg only exerts force along the leg axis, the tangent velocity of the body is unchanged in an instantaneous bounce. With no thrust, the natural leg dissipation leads to a change in body velocity that is purely restitutive:

$$\mathbf{v_r} = \begin{bmatrix} 1 & 0 \\ 0 & -\epsilon \end{bmatrix} \cdot \begin{bmatrix} v_{ti} \\ v_{ni} \end{bmatrix} \tag{3.1}$$

The velocities are expressed in the leg coordinate frame; the subscript i indicates "impact" and the subscript r indicates "rise," synonymous with liftoff or takeoff. This equation states that the tangent velocity is unchanged and the normal velocity is reversed with a momentum loss. As shown in the top of Figure 3.5, if the restitution ϵ equals 1, then the angle of restitution equals the angle of impact as measured about the leg position.

3.1.2 Instantaneous Bounce

The previous section sketches the general assumptions. Now is considered the instantaneous hopper bounce given the three mechanical control inputs: position of the leg at impact, energy released during stance, and the relative position of the hip and body center of mass. The leg position is treated implicitly since it determines the orientation of the normal and tangent frame.

The position of the hip relative to the center of mass controls the body torque impulse produced by the ground impact. Since the hip is a free pivot this is the only torque acting on the body. Moving the hip during flight is a kinematic reconfiguration of a massless leg that requires zero force, and the hip position rigidly clamps during stance to carry the high ground forces passively. In this fashion the hip actuation is a conservative mechanism that involves no loss or negative work.

The following analysis of the body torque is formulated as an impulsive analysis of two rigid bodies colliding. The control variables are the leg angle ϕ, the hip translation h, and the thrust energy ξ.

The body torque is the cross product of the foot vector and the ground force (illustrated in Figure 1.4):

$$\tau = \mathbf{r} \times \mathbf{F} \tag{3.2}$$

The ground force lies in the normal direction:

$$\mathbf{F} = F_n \hat{\mathbf{N}} \tag{3.3}$$

The hip translation h and the body torque τ are scalar quantities. The moment arm r defines the body torque:

$$r \equiv h\cos(\theta - \phi) \tag{3.4}$$
$$\tau = rF_n \tag{3.5}$$

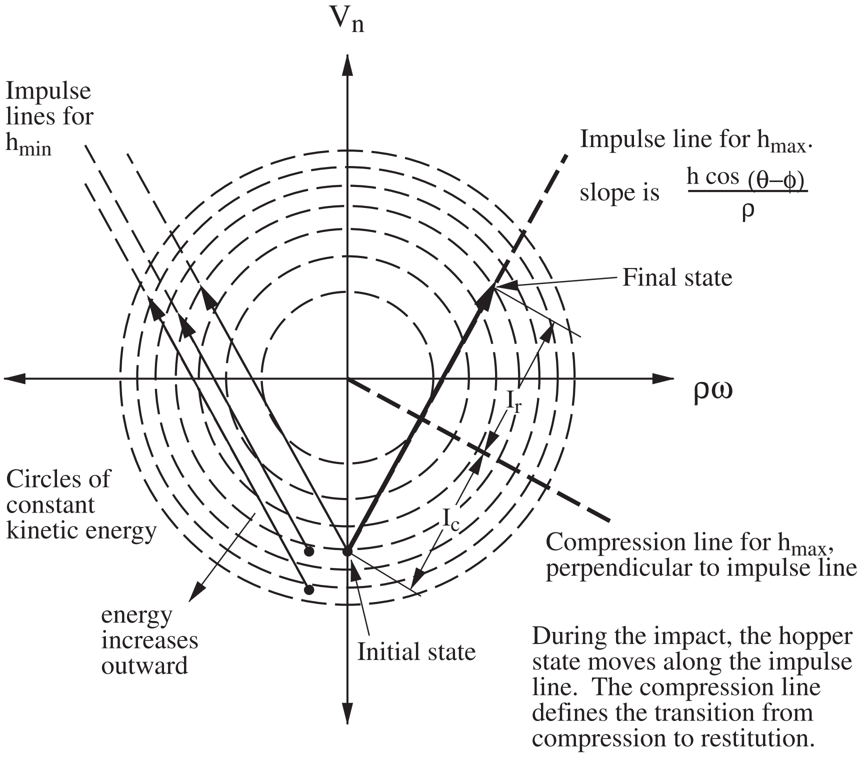


Figure 3.3: Illustration of impact physics in velocity space for a translating-hip hopper. The horizontal axis is scaled by the radius of gyration ρ so the units are m/s on both axes. The bold arrow represents an impact with the hip offset set at maximum. At the beginning, the normal velocity is negative and rotational velocity is zero. During impact, the hopper velocity moves from initial to final state along the impulse line. Assuming elastic impact, the compression impulse I_c equals the restitution impulse I_r. The final state is on the same energy circle with a positive rotational velocity. Several transitions from different starts are also shown for the rear most hip position $h_{\min}$.

The normal impulse I_n resulting from the ground force defines the change in both translational and angular velocities:

$$m\Delta v_n = I_n \tag{3.6}$$
$$m\rho^2\Delta\dot{\theta} = I_n r \tag{3.7}$$

The impact analysis is phrased like a Poisson impact analysis, graphically shown for the hopper in Figure 3.3. In this form the velocity state of an object moves along the "impulse line" in velocity space during the impact, accumulating impulse I_c until it reaches the "compression line." At this point the motion changes direction and the spring releases impulse, continuing ϵI_c further to the final state. The location and orientation of the impulse and compression lines are determined by the kinematics of the contact.

The velocity space is plotted with the angular axis scaled by the radius of gyration ρ to make the units equal. The impulse line in $[\, v_n \quad \rho\dot{\theta}\,]$ space has the following slope:

$$\frac{\rho\Delta\dot{\theta}}{\Delta v_n} = \frac{r}{\rho} \tag{3.8}$$

The compression impulse I_c defines the state at maximum compression:

$$v_{nc} = v_{ni} + \frac{1}{m}I_c \tag{3.9}$$
$$\dot{\theta}_c = \omega_i + \frac{r}{m\rho^2}I_c \tag{3.10}$$

The compression line is reached when the normal hip velocity reaches zero:

$$v_{nc} + r\dot{\theta}_c = 0 \tag{3.11}$$
$$\dot{\theta}_c = -\frac{v_{nc}}{r} \tag{3.12}$$

This relates the previous two expressions to find I_c:

$$-\frac{v_{ni}}{r} - \frac{1}{rm}I_c = \omega_i + \frac{r}{m\rho^2}I_c \tag{3.13}$$
$$I_c\left(-\frac{1}{rm} - \frac{r}{m\rho^2}\right) = \omega_i + \frac{v_{ni}}{r} \tag{3.14}$$

$$
\begin{aligned}
I_c &= \left(\frac{r\omega_i + v_{ni}}{r}\right)\left(\frac{rm\rho^2}{-\rho^2 - r^2}\right) && (3.15)\\
I_c &= -\frac{m\rho^2(r\omega_i + v_{ni})}{\rho^2 + r^2} && (3.16)
\end{aligned}
$$

Applying Poisson restitution:

$$
I_n = (1+\epsilon)I_c = -(1+\epsilon)\frac{m\rho^2(r\omega_i + v_{ni})}{\rho^2 + r^2} \tag{3.17}
$$

Substituting this yields the takeoff velocity:

$$
\begin{bmatrix} v_{tr} \\ v_{nr} \\ \omega_r \end{bmatrix} = \begin{bmatrix} v_{ti} \\ v_{ni} - \frac{(1+\epsilon)\rho^2(r\omega_i + v_{ni})}{\rho^2 + r^2} \\ \omega_i - \frac{(1+\epsilon)r(r\omega_i + v_{ni})}{\rho^2 + r^2} \end{bmatrix} \tag{3.18}
$$

Since the altitude doesn't change during stance, the energy dissipated in the impact is the change in kinetic energy:

$$
\begin{aligned}
E_{\text{diss}} &= E_r - E_i && (3.19)\\
&= \frac{m}{2}({v_{tr}}^2 + {v_{nr}}^2 + \rho^2{\omega_r}^2) - \frac{m}{2}({v_{ti}}^2 + {v_{ni}}^2 + \rho^2{\omega_i}^2)
\end{aligned}
$$

Which may be simplified with equation 3.18 to the following:

$$
E_{\text{diss}} = (\epsilon^2 - 1) \cdot \frac{m\rho^2(v_{ni} + r\omega_i)^2}{2(\rho^2 + r^2)} \tag{3.20}
$$

The previous derivation solves the case of an idealized leg with restitution ϵ but no thrust energy. However, the hopper may increase its kinetic energy by compressing the leg using the Bow String during flight, then releasing it to full length during stance. This stores a well-defined energy ξ in the leg that is a function of the leg compression and may be determined by measuring the leg potential.

However, the velocity change and the dissipation were derived from a pure restitutive loss without regard to whatever impulse is produced by the thrust mechanism. This may be resolved by assuming that the gain and loss processes are independent: in the instant of impact, the spring dissipates energy as a function of the impact state, and the thrust process produces an extra impulse that does work on the body. This assumption is motivated by the actual hopper in which the spring loss is uniform whether it is flexed around a low-energy state or a high-energy state.

With this assumption, the liftoff state may be derived by rewriting equation 3.18 with an additional fictional restitution term ϵ_t that represents the effect of thrust:

$$\begin{bmatrix} v_{tr} \\ v_{nr} \\ \omega_r \end{bmatrix} = \begin{bmatrix} v_{ti} \\ v_{ni} - \frac{(1+\epsilon+\epsilon_t)\rho^2(r\omega_i+v_{ni})}{\rho^2+r^2} \\ \omega_i - \frac{(1+\epsilon+\epsilon_t)r(r\omega_i+v_{ni})}{\rho^2+r^2} \end{bmatrix} \tag{3.21}$$

This may solved by satisfying the following constraint on the kinetic energy at liftoff; the net energy change is written as ΔE:

$$\Delta E = \xi + E_{\text{diss}} \tag{3.22}$$

$$\frac{1}{2}m{v_{nr}}^2 + \frac{1}{2}m\rho^2{\omega_r}^2 = \frac{1}{2}m{v_{ni}}^2 + \frac{1}{2}m\rho^2{\omega_i}^2 + \Delta E \tag{3.23}$$

The inclusion of ϵ_t may be justified by considering that the solution represented by equation 3.18 is a locus of solutions along a line in velocity space parameterized by the restitution ϵ. The kinematics are fixed in an instantaneous bounce so the impulse directions are fixed but the magnitudes depend upon the restitution. Since the thrust mechanism modifies the magnitude of the leg impulse, it will produce the same locus of solutions.

The energy constraint of equation 3.23 is an ellipse in the same velocity space, and the solution of the takeoff velocity is the intersection of the line and the ellipse. This is not presented symbolically but is illustrated in a velocity space impulse diagram in Figure 3.3.

Storing thrust energy in the leg effectively makes it a spring with a variable restitution that may be greater than one. This is true because the impact assumes an instantaneous bounce: in practice the parts are moving and the time history of the leg and thrust forces affects the outcome. But although both the leg potential and thrust forces are unknown, the fact that the body position doesn't change during stance means the relationship between angular and translational acceleration depends only upon the kinematics. This is reflected in the definition of the impulse line: no matter how far the state moves, the slope is fixed.

3.2 Idealized Two-Control Model

The previous section analyzed an idealized Bow Leg Hopper with a translating hip. This section presents another idealized model with a fixed hip. This is in preparation for the empirical model based on the actual prototypes presented in the following section.

In this model the body mass is placed exactly on the hip axis. The leg force always passes through the center of mass and no torques at all are exerted on the body. This leads to a straightforward reduction of the previous equations with h and r set to zero in which the body orientation no longer matters:

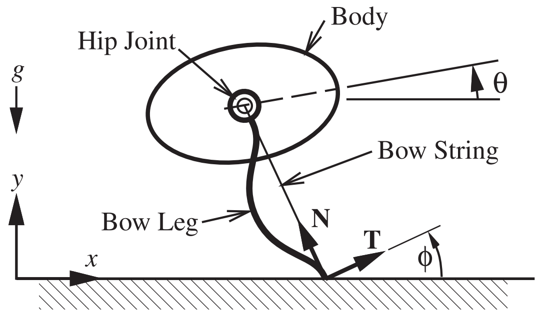


Figure 3.4: Cartoon of the planar Bow Leg Hopper. The leg is a lightweight fiberglass spring with high restitution. The hip is a ball bearing pivot that exerts minimal body torque. The thrust actuator and leg angle actuator are located on the body but not illustrated. The Bow String attaches the thrust actuator to the toe and can be retracted to compress the leg. The center of mass (not illustrated) is located just below the hip for natural pitch stability. The hopper position is defined by the body center of mass position and orientation (x, y, θ) and the leg angle ϕ, which is measured with respect to the world -Y axis. Gravity points in the -Y direction. The impact is analyzed in the leg frame $[\,\hat{\mathbf{T}} \quad \hat{\mathbf{N}}\,]$.

$$\begin{bmatrix} v_{tr} \\ v_{nr} \end{bmatrix} = \begin{bmatrix} v_{ti} \\ -(\epsilon + \epsilon_t) v_{ni} \end{bmatrix} \tag{3.24}$$

$$E_{\text{diss}} = (\epsilon^2 - 1) \cdot \frac{m {v_{ni}}^2}{2} \tag{3.25}$$

$$\frac{1}{2} m {v_{nr}}^2 = \frac{1}{2} m {v_{ni}}^2 + \Delta E \tag{3.26}$$

The controls are leg angle ϕ, appearing as the orientation of the analysis coordinate frame, and thrust energy ξ, a component of ΔE.

These may be combined to find the reduced form of the general solution that was presented graphically in Figure 3.3:

$$v_{nr} = \sqrt{\frac{2 \cdot \xi}{m} + \epsilon^2 \cdot {v_{ni}}^2} \tag{3.27}$$

The transform from the impact velocity to the leg coordinate frame is a rotation matrix using the leg angle:

$$\begin{bmatrix} v_{ti} \\ v_{ni} \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} \tag{3.28}$$

With basic falling body dynamics may be derived some spring limits. The vertical velocity at impact:

$$\dot{y}_i = \sqrt{2g\Delta y} \tag{3.29}$$

The maximum energy stored in the spring during stance:

$$E_{s\max} = \frac{m}{2}(-\dot{x}\sin\phi + \sqrt{2g\Delta y}\cos\phi)^2 \tag{3.30}$$

The energy dissipated by the spring:

$$E_{\text{diss}} = (\epsilon^2 - 1)\frac{m}{2}(-\dot{x}\sin\phi + \sqrt{2g\Delta y}\cos\phi)^2 \tag{3.31}$$

3.3 Empirical Modelling

Models are approximations created with a purpose in mind. This section presents the model of the physics used by the controller and planner.

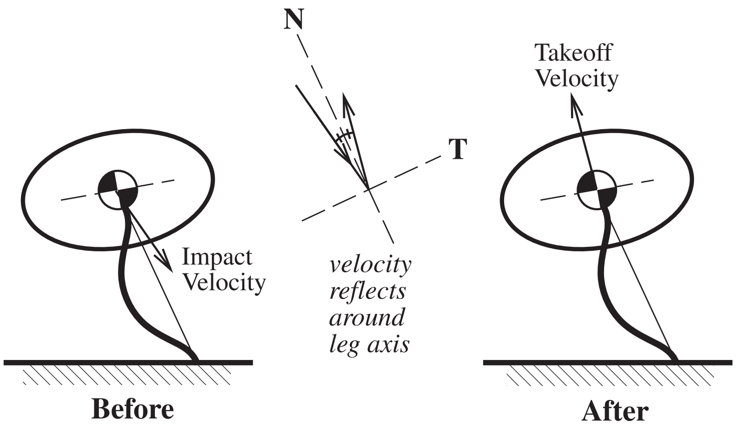
The idealized models addressed in the previous two sections are obviously unrealistic: no hopper can bounce off the ground instantaneously. Nevertheless, the models are closed-form, which is highly desirable for planning purposes. The strategy adopted was to consider the likely ways the physical hopper deviates from the ideal, measure the differences, and form an empirical model that explains the behavior *accurately enough* for control purposes. The appropriate level of accuracy was determined subjectively through experimentation to judge which candidate empirical models over-fit typical data and which left residual error attributable to experimental noise.

Another issue is calibration. The models include both parameters and calibration functions; a few examples are gravity, leg restitution, and the function mapping desired thrust energy to thrust mechanism angle. On a finer scale are functions to map servo angle to servo command pulse widths. A few of these are convenient to calibrate with direct measurement, but most are determined by fitting parameters based on test data. The data is acquired from the hopper as it performs a special-purpose calibration procedure.

3.3.1 Modelling Physics for Planning

The physical model defines the world for the planner; it defines the available actions and their results. The planner is treated in detail in Chapter 5, and the more abstract definition of a controller in Section 4.1, but the basic philosophy behind the development of the model used for planning is as follows: it needs to be accurate enough to

Instantaneous Impact Model



Empirical Leg Sweep Model

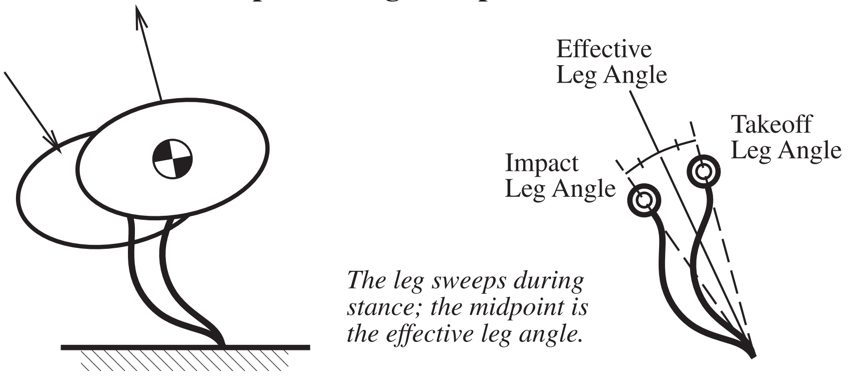


Figure 3.5: The top half illustrates hopping with a perfectly elastic instantaneous impact. The angle of restitution with respect to the leg angle equals the angle of impact, and the body position does not change during impact. The bottom half illustrates the empirical model that assumes a finite stance time. The effective leg angle is the midpoint of the sweep arc.

produce a plausible plan; modelling error is tolerated by use of feedback control during plan execution. It should be fast to evaluate, ideally closed-form. It is desirable for it to have a computable inverse to allow planning paths backwards. It is desirable for it to be abstract enough to represent multiple mechanisms so the planner could be easily generalized.

An abstract view of the hopper is as a point particle moving between discrete points in the state space. Each point represents one flight trajectory and each transition between points represents one bounce. The transition made from each state is controlled by the leg angle and a generalized thrust that includes negative values. Each possible transition may be categorized as colliding with terrain or not. Each state defines limits on the controls; the leg angle is limited to foot placements compatible with the terrain and the thrust ranges between natural dissipation (zero thrust mechanism angle, but negative energy change) and the maximum energy gain.

This particular generalization is closely related to the Bow Leg; it may safely ignore the body orientation and details of the stance because the Bow Leg mechanism is so simple that the hardware behaves close to an abstract model in the first place. Nevertheless, this model is universal enough to be usable by any terrain crossing hopping robot that doesn't care about incorporating attitude change into its plans.

3.3.2 Differences from Idealization

The instantaneous analysis assumes that the leg angle doesn't change during stance. If the leg is aligned with the impact velocity, this assumption is close to the truth since the tangent velocity will be close to zero. If the leg is not aligned, the leg will sweep: this means the body force direction changes during stance and the takeoff position is different from the impact position. Another alignment is with gravity; if the leg is not vertical, then gravity will accelerate the body about the foot. Both of these effects are more pronounced with longer stance duration. In general, the instantaneous model becomes less accurate as the hopper travels faster or bounces higher.

Another assumption is that the leg is massless. In practice, the leg inertia means the reaction torques from rotating the leg during flight will counter-rotate the body. Inertia also leads to energy losses during impact due to the inelastic foot collision. And the leg length is not generally constant from impact to takeoff since the thrust mechanism shortens the leg an amount proportional to the energy stored. This means the hopper makes ground contact at a lower potential energy state than when it breaks contact.

Generally solving for these effects requires solution of the equations of motion during stance. Since the leg potential is a non-linear function that must be measured from the hardware, this would lead to a solution based on numerical integration from the impact conditions. With certain leg potential functions this may be addressable analytically [SK95] [SK97], but for this project the pragmatic need for a efficient model led to the approximations sketched in the following paragraphs.

Leg Sweep

The leg sweep ϕ_{sweep} is the angle swept by the leg during stance. Since the foot is stationary this represents body motion. Typically the hopper stance time is short and the motion during stance is relatively symmetric. The model assumes that the integral of the stance forces over time is symmetric about the midpoint of the sweep arc. The midpoint of the sweep arc is approximately the "effective leg angle" $\phi_{\text{effective}}$, which is the leg angle for the idealized model that produces the same change in velocity.

In general, the leg sweep angle is a function of the state and depends on the tangential velocity and the stance time. In practice, the stance time is relatively constant. Experimentation with a variety of forms led to the simple approximation of a sweep angle as a constant multiplied by the tangential velocity:

$$\begin{aligned} \phi_{\text{sweep}} &= k_{\text{sweep}} \cdot -v_{ti} \\ \phi_r &= \phi_i + \phi_{\text{sweep}} \\ \phi_{\text{effective}} &= \phi_i + 1/2 \cdot \phi_{\text{sweep}} \end{aligned} \tag{3.32}$$

If the effective leg angle $\phi_{\text{effective}}$ is used as the value for ϕ in the ideal model, the result is good enough to be a useful predictor of takeoff velocity within the typical Bow Leg operating limits. The takeoff position is computed using the takeoff leg angle ϕ_r and the location of the foot contact.

Thrust Calibration

The leg angle model in the previous section approximates the dynamics of stance. The thrust model is concerned with calibrating thrust mechanism, approximating the relationship between the thrust motor angle and the energy stored in the leg. In practice the energy dissipated by the leg was observed to follow the ideal restitution model closely, so no approximation was required.

The energy stored in the leg is a function of thrust motor angle and is independent of the impact state. The following approximation includes the kinematics of the thrust mechanism, leg potential, and any losses in transfer to kinetic energy:

$$\xi = p_1 u_e + p_2 {u_e}^2 \tag{3.33}$$

The two terms involving the thrust motor angle u_e form a quadratic approximation of the energy stored in the leg. In practice energy is represented in units of "velocity-squared" so that the machine mass is eliminated from all calculations.

Flight Model

The flight model computes the trajectory resulting from a position and velocity at takeoff. The model assumes constant gravity and a constant lateral friction force; the effective gravity produced by the constraint boom and gravity compensation spring varies slightly with altitude, but the effect is negligible. The measurable but low horizontal deceleration is presumably due to bearing friction and tether drag.

The impact occurs when the foot touches the ground. Given the position and velocity of the hopper at the flight apex $[\,x \quad y \quad \dot{x}\,]$, the leg length l, the leg angle ϕ, and the altitude of the ground under the foot y_{gnd}, the following finds the state at impact:

$$
\begin{aligned}
y_i &= y_{\mathrm{gnd}} + l\cos\phi \\
t_{\mathrm{fall}} &= \sqrt{2(y_i - y)/g} \\
x_i &= x + \dot{x}t_{\mathrm{fall}} + 1/2 \cdot \ddot{x}_{\mathrm{lat}}{t_{\mathrm{fall}}}^2 \\
\dot{x}_i &= \dot{x} + \ddot{x}_{\mathrm{lat}}t_{\mathrm{fall}} \\
\dot{y}_i &= gt_{\mathrm{fall}}
\end{aligned}
\tag{3.34}
$$

The effective gravity is g and the lateral force is modeled by the acceleration constant $\ddot{x}_{\mathrm{lat}}$.

At takeoff, the hopper rises from the liftoff state $[\,x_r \quad y_r \quad \dot{x}_r\,]$ to the next apex:

$$
\begin{aligned}
t_r &= -\dot{y}_r/g \\
x &= x_r + \dot{x}_r t_r + 1/2 \cdot \ddot{x}_{\mathrm{lat}}{t_r}^2 \\
y &= y_r + \dot{y}_r t_r + 1/2 \cdot g{t_r}^2 \\
\dot{x} &= \dot{x}_r + \ddot{x}_{\mathrm{lat}}t_r
\end{aligned}
\tag{3.35}
$$

3.3.3 Calibration Procedure

Most of the model coefficients and physical parameters are determined by fitting the model predictions to observed data gathered while the hopper bounced back and forth in a special test procedure. The fitting is broken down into a progression of low-dimensional least-squares optimizations using Powell's Method. Part of the work of developing the calibration was involved with identifying a sequence with no forward dependencies.

The same sensors and actuators that are used for control are used to build the model. In fact, some of the leg calibration that might be measured manifestly is instead calibrated by measuring the effect on the velocity prediction.

Exploration

The exploration procedure is typically run for about 400 bounces over level ground. The purpose is to sample the state space and the control space over the full operating domain. The thrust progressively advances through four discrete values intermixed with zero thrust. This allows the hopper to reach different altitude equilibria, measuring both the passive bounce performance and the thrust performance.

While the hopper is cycling through altitudes, it is hopping back and forth across a point on the ground. At each direction switch, it chooses a random velocity; it may moves one or several bounces in each direction. On each bounce the hopper computes a leg angle using normal linear velocity feedback but discretizes the output. Leg angles near zero are clamped to zero to increase the number of samples usable for estimating leg restitution.

The data collected include the normal sensor readings, the control outputs and trajectory data. The sensor readings include the state and time at the trajectory peak and body attitude before impact. The trajectory is the position and velocity recorded at every tick during flight.

Fitting

The flight parameters are estimated first. A quadratic is fit to each trajectory to estimate the time and altitude of the peak and the vertical acceleration. This is used to cull potentially erroneous trajectories by checking the individual gravity parameters against the distribution, and the fitted peak time with that recorded by the controller. This culling removes trajectories that were not free flight, i.e., the start trajectory and collisions.

After culling, the gravity parameter is estimated by a global minimization of position error over all valid flights. The reference parabola is computed using the newly estimated peak time and position. After finding the vertical acceleration, a similar fit computes the (typically small) lateral flight accelerations.

The thrust mechanism is calibrated next. Estimating restitution and thrust involves comparing the energy of successive trajectories, hence it depends on the newly fitted flight model. The restitution is estimated first by considering all bounces with no thrust and zero leg angle. Since the position data is more precise than the velocity estimate, only cases with zero leg angle (hence no nominal change in lateral velocity) are included: the energy restitution for these cases is the following ratio:

$$\epsilon^2 \simeq (y_{n+1} - l)/(y_n - l) \tag{3.36}$$

The values y_n and y_{n+1} are the peaks of successive bounces. The estimated energy restitution is the average of the ϵ^2 values.

Parameters Computed from Training Set 98-02-21		
Parameter	Value	Definition
g	$-2.43\ \mathrm{m/sec}^2$	effective gravity
ϵ	0.82	restitution
ϵ^2	0.68	energy restitution
k_{sweep}	0.16	sweep angle coefficient
p_1	0.45	ξ vs. thrust angle, linear term
p_2	-0.07	ξ vs. thrust angle, quadratic term
Error Statistics on Training Set 98-02-21		
Statistic	Value	Definition
σ_x	$8\ \mathrm{mm}$	std. dev. of x error
σ_y	$7\ \mathrm{mm}$	std. dev. of y error
$\sigma_{\dot{x}}$	$17\ \mathrm{mm/sec}$	std. dev. of $\dot{x}$ error
N	442	samples in training set

Table 3.1: Model Parameters and Fitting Statistics

Next the quadratic thrust calibration curve is estimated using the newly fit leg restitution. Again only cases with zero leg angle are considered. The predicted energy is as follows:

$$E_{n+1} = mg((y_{n+1} - l) - \epsilon^2(y_n - l)) \tag{3.37}$$

The error is the difference between E_{n+1} and the prediction of the quadratic model, and the model parameters are tuned to minimize the sum of squared errors. A final step is to compute the range of permissible thrust energies by evaluating the thrust position limits in the model.

Having generated a model to compute actual thrust energy from the thrust control value, the impact model may be fit. It is actually performed twice, once to estimate outliers and cull them, and again to retune parameters based on the remaining data. This fitting computes the two remaining parameters, a leg angle offset and the leg sweep coefficient. The leg offset is nominally set by directly measuring the position of the leg for various control inputs when calibrating the hardware, but an additional tuning freedom is allowed so the origin of the leg position is estimated from the effect on the motion.

A sample set of parameter values and statistics is shown in Table 3.1. The errors listed are the residual; i.e., the distribution of the differences between the predicted and actual trajectory parameters on the same data set with which the model was fitted.

3.3.4 Terrain Model

The "terrain" used in the experiments consisted of flat concrete floor with bricks laid out as footholds or obstacles. This project did not address the problem of sensing terrain; in all cases the terrain was designed in advance, compiled into the controller, and then the pieces laid out on the floor. Holes lower than floor level were not physically practical but were simulated by labelling particular floor regions as "invalid footholds."

The chief function of the terrain model is to compute the ground contact given a flight state and impact leg angle. The terrain is modelled as connected line segments, each labelled with a friction coefficient. Horizontal segments are deemed "safe footholds."

3.4 Discussion

Those readers patient enough to read all the way through Section 3.3 may well be wondering: "why all the fuss over a few quadratic fudge factors?" There are a few answers: one is that the empirical models *were* considerably more elaborate until it became apparent that the performance of the simplest models was just as good. But there are two particular lessons here.

The first is that the Bow Leg Hopper has such simple dynamic behavior that the idealized model is almost good enough for planning. This is especially true if the hopper is picking its way slowly amidst obstacles, in which case the lateral velocity is low and the leg sweep has little effect.

The second is that the hopper has enough sensor noise and actuation error that the residual masks the subtle model error. This is a case of making it "good enough," albeit the standard was defined by error designed-in elsewhere. Certainly improving the state estimation and making the leg positioner more consistent would put more pressure on the modelling.

In the future, the control model needs to include the change of pitch velocity during stance. This might be controlled mechanically by implementing a hip translation axis. This would put the analysis of Section 3.1 fully to work and introduce additional calibration needs. At the minimum the radius of gyration and the translation servo calibration would need to be measured. And since this mechanism would involve potentially large body torques, the change of body attitude during stance might differ substantially from the ideal case.

Another mechanical development could be a leg equipped to capture impact energy by clamping the Bow String before the leg reaches full length. This involves only a modest change in modelling since the forces will still lie along the leg axis. The big differences will be representing asymmetry in the stance time and taking off with lower potential energy.

Chapter 4

Hopper Control

The Bow Leg Hopper is a natural hopping oscillator. Its efficiency stems from the use of springs and bearings to conservatively carry the forces of locomotion. For the same reason, it is discretely controlled by configuring the leg mechanism during flight. This is primarily motivated by a desire for high efficiency and small actuators, but a side benefit is that the controller need only compute one set of controls per bounce. By the same token, the controller *may only apply* one control per bounce, so disturbances that occur after sensing or during the bounce may only be rejected on successive impacts. If the ground crumbles beneath the foot, the hopper can only helplessly wait until it is in the air again.

Every possible controller for the planar hopper takes the form of a function mapping three trajectory parameters to two control outputs once every hopping cycle. The simplest controllers are linear feedback rules not unlike Raibert three part control [Rai86]. These only work on level ground; the planning controller for crossing "terrain" uses a combination of graph-search planning and feedback control. The planner searches to find sequences of foot placements compatible with the terrain and the physics. Feedback control is computed once per bounce to reject local errors; when the plan is exhausted or accumulated error becomes too great, a new plan is generated.

This chapter defines the control problem implied by this mechanism, presents the linear controllers and details of the real-time control implementation, and graphical methods for deriving state space constraints from terrain constraints. The planner is presented in Chapter 5. The discussion is limited to the case of a planar hopper, although much of the argument also applies to future 3D hoppers.

The first section defines the relationship between the degrees of freedom and the relevant control freedoms in order to justify the control model.

4.1 Controller Formulation

The Bow Leg mechanical design permits only one control cycle per bounce and this defines the properties of the controller. The controller function takes the following form:

$$\begin{bmatrix} \phi_{n+1} \\ \xi_{n+1} \end{bmatrix} = f(x_n, y_n, \dot{x}_n) \tag{4.1}$$

In this function the variables (ϕ_{n+1}, ξ_{n+1}) are the leg angle and stored leg energy at impact and $[\, x_n \quad y_n \quad \dot{x}_n \,]$ defines the trajectory preceding the impact. This function summarizes the control and comprises the physical model used for feedforward, terrain geometric constraints, the task being performed, and error feedback. The discrete form can be justified by examining the effect of each actuator and the definition of state.

The leg servomotor determines the angle of the leg prior to impact. During flight, the leg carries no load and can be positioned quickly. This motion only slightly affects body pitch since the leg mass is approximately 1% of the body mass. During stance, the leg positioning motor is physically decoupled from the leg. It is conceivable the leg servo could be repositioned during stance in order to exert horizontal ground forces as the Bow String regains tension at liftoff, but this analysis neglects this possibility. Thus the leg motion can be entirely described as ϕ_n, the leg angle in world coordinates at impact n.

The thrust motor determines the energy stored in leg tension prior to impact. The leg always begins flight at its lowest energy state. During flight, the motor performs positive work on the leg spring. It is conceivable it might be immediately reversed to dissipate some stored energy but the *net work* during flight is always non-negative. During stance, the thrust motor becomes physically decoupled from the leg as the now-slack string is released. The leg then extends to full length, and all stored energy is released. The thrust can be entirely described as ξ, a non-negative potential energy added to the kinetic energy. As described in Section 3.1.2, the sum of thrust ξ and dissipation E_{diss} is the net energy change ΔE. The range of ΔE available to the controller includes negative values and may even be entirely negative in states with high dissipation.

The full physical state nominally has ten dimensions: three body DOF, two actuator DOF, and the corresponding velocities. We make several assumptions to define a trajectory using only three dimensions. First, we may neglect pitch and pitch velocity since the body is designed to passively stabilize orientation and rotates like a slow pendulum. This axis is decoupled from the other coordinates since body rotations only slightly affect the direction of leg forces, and the leg position is independently defined in world coordinates. Second, the actuators have insignificant dynamics on the time scale of the hopping cycle and may be treated simply as outputs.

The hopper may thus be treated as a point particle with four state variables $[\,x \quad y \quad \dot{x} \quad \dot{y}\,]$. However, we assume that all constraints are time-invariant and thus only the geometry of the trajectory matters. Since the free flight physics is known, each trajectory can be described by only *three* parameters; we use the vector $[\,x_n \quad y_n \quad \dot{x}_n\,]$ which specifies the position and velocity at the apex of the trajectory.

Note that the leg and thrust values are a function of time during flight $(\phi(t), \xi(t))$, but only the final values (ϕ_n, ξ_n) affect the impact. The abstract control problem may be described with discrete functions but the implementation does require control over time. The abstract control values closely correspond to the mechanical freedoms: the stored energy is a monotonic function of the thrust servo angle, and the leg angle ϕ is the sum of the body attitude θ and the leg servo angle.

The low power of the motors does impose timing constraints. The minimum time required to store leg energy depends on the magnitude of ξ and the maximum motor power. In practice, the entire flight time is required to store a large impulse, so energy storage for impact n must typically begin immediately after takeoff $n-1$; this energy will affect the trajectory following impact n. In contrast, the leg servo can typically position the leg shortly before impact since it is moving an unloaded low-mass leg.

4.2 Linear Controller

The simplest controllers of this form are linear feedback rules. For example, the linear position controller takes the following form:

$$\begin{aligned} q_{\mathrm{te}} &= \dot{x}^2 - 2gy \\ \bar{\dot{x}} &= k_{\mathrm{xdx}}(\bar{x} - x) \\ \\ \phi &= k_{\mathrm{xdphi}}(\bar{\dot{x}} - \dot{x}) \\ \xi &= k_{\mathrm{thrust}}(\bar{q}_{\mathrm{te}} - q_{\mathrm{te}}) \end{aligned} \tag{4.2}$$

The vector $[\,x \quad y \quad \dot{x}\,]$ specifies the apex of the current trajectory. The first line computes the total energy q_{te} in units of velocity-squared (i.e, the $1/2M$ term is factored out). The second line is a proportional gain to choose the desired lateral velocity $\bar{\dot{x}}$ as a function of position error. The last two lines implement independent proportional feedback that computes the leg angle and thrust energy to regulate the velocity and total energy to the desired values. This leg angle must be translated to body coordinates; both controls are then translated to motor angles using calibration functions. The velocity-squared energy unit is used for all energy values (including ξ) so the machine mass doesn't need to be explicitly known (it is implicitly part of the thrust calibration).

The linear control is related to the Raibert three part control [Rai86]: the touchdown leg angle is similar to foot placement and the leg retraction at impact analogous to thrust during stance. The leg angle controls forward speed much like the Raibert model, and the thrust controls total energy, roughly equivalent to hopping height. The "third part," body attitude, is passively controlled so there is no counterpart here. So although the controller is discrete in time and exerts no actuator forces during stance, the physics of legs leads to similar control relationships.

This is a very simple controller but is sufficient for holding a position or hopping at constant velocity (by skipping the second line of equation 4.2). Next is examined the details of putting this controller in operation. The more elaborate planning controller is presented in Chapter 5.

4.3 Real-time Implementation

The hopper control may be discrete in time but the motion is continuous. This section documents the details of connecting the abstract control to the actual mechanism.

The real time component of the hopper control reads the sensors; estimates the state; continuously issues servo position commands; manages a state machine representing ascent, descent, and stance; and schedules the control computations. At the lowest level, the hobby servos use internal position feedback to move to commanded positions encoded as PWM (pulse-width-modulated) signals from the control computer.

4.3.1 State Machine

Scheduling is controlled by a state machine that cycles though four phases: ascent, descent, pre-impact, and stance. At liftoff, the onset of the ascent phase, the thrust command for the following impact is translated to a thrust motor angle and sent to the servo. Shortly after the apex is the onset of the descent phase, at which point the trajectory state is estimated by fitting parabolas to the raw x and y sensor data. The discrete control is then computed, which includes calculating error estimates, adjusting the plan, and possibly initiating replanning. During descent, the state machine continuously updates the leg servo angle using the the commanded leg position and the body attitude estimate. At the pre-impact altitude, the leg position is frozen to allow settling. During stance, the motion evolves according to the spring-mass physics of the machine.

The state machine update rate is 200Hz. On each cycle the sensors for (x, y, θ) position and the string tension switch are read. The raw position measurements are filtered to estimate position and velocity, and the filtered values are used to trigger state machine transitions.

4.3.2 Sensor Processing

Discrete control only needs discrete sensing. For the hopper, sensing takes place on two time scales; a high-quality state estimation for the discrete controller is performed once during flight, and a continuous state estimation is generated for the real time system. The high-quality estimate is generated by fitting quadratic polynomials to the 61 sample points centered on the apex of the trajectory, and the continuous estimates are computed with a simple estimator using Euler integration.

The continuous estimator computes body state (position and velocity) by independently processing sampled data from the x, y, and θ position sensors (Figure 3.4). The $(x, \dot{x})$ estimator takes the following form:

$$\begin{aligned} \tilde{x}_i &= x_{\mathrm{raw}i} - x_{i-1} \\ x_i &= x_{i-1} + (\dot{x}_{i-1} + k_1 \tilde{x}_i) \cdot \Delta t \\ \dot{x}_i &= \dot{x}_{i-1} + k_2 \tilde{x}_i \cdot \Delta t \end{aligned} \tag{4.3}$$

The variable $x_{\mathrm{raw}i}$ is the latest position sensor value scaled into real world units; $(x_i, \dot{x}_i)$ is the new state estimate, $\tilde{x}_i$ is the computed position error, and Δt is the tick interval (5 msec). The gains (k_1, k_2) are hand-tuned; typical values for the x and y estimators are $(40, 400)$.

The estimator for body orientation and angular velocity $(\theta, \dot{\theta})$ has the same form. The estimator for $(y, \dot{y})$ is similar but has one difference; the velocity update includes a feedforward acceleration $\ddot{y}_{\mathrm{est}}$ to model the effect of stance forces:

$$\dot{y}_i = \dot{y}_{i-1} + (\ddot{y}_{\mathrm{est}} + k_2 \tilde{y}_i) \cdot \Delta t \tag{4.4}$$

In practice the feedforward term was chosen to be zero except during stance (despite gravity). During stance it took a positive value (typically around 20) that was hand-tuned to minimize the error in the $\dot{y}$ estimation after stance, as judged by the y estimator error during ascent.

The encoders used on the x and y axes are inexpensive 512 line optical encoders that are decoded to 2048 position values. The hopping circle is 9.29 m in circumference, so the x encoder resolution is approximately 4.5 mm of lateral travel. At very low speeds this granularity does lead to a slight cogging in the sensor data. The body rotation axis uses a potentiometer for historical reasons; the analog noise is a source of error and requires lower filter gains (e.g. $(20, 100)$) for smoothing. This causes a noticeable lag, which can lead to leg positioning error.

Each quadratic fit is computed using a fitting matrix multiplied by the vector of 61 sensor readings $\vec{x}_{\mathrm{raw}}$ to find polynomial coefficients:

$$\begin{aligned} \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix}^{\mathrm{T}} &= \mathbf{A} \cdot \vec{x}_{\mathrm{raw}} \\ \begin{bmatrix} y_0 & y_1 & y_2 \end{bmatrix}^{\mathrm{T}} &= \mathbf{A} \cdot \vec{y}_{\mathrm{raw}} \end{aligned} \tag{4.5}$$

The fitting matrix $\mathbf{A}$ is a constant matrix generated off-line with a singular value decomposition and compiled into the controller. The idea is that an approximation to a function may be generated at a discrete number of points by multiplying basis functions sampled in time by a set of coefficients. The basis functions are the terms of a quadratic evaluated at the times of interest:

$$\mathbf{B} = \begin{bmatrix} 1 & t & t^2 \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{4.6}$$

The matrix $\mathbf{B}$ has as many rows as time samples; in this case, 61 rows generated with values of t ranging from $[\,-30\Delta t \quad 30\Delta t\,]$. Since the sampling interval is 5 msec this represents 300 msec of flight time. Running in higher effective gravity would require a smaller sample length.

The fitting matrix can be computed by considering the relation between the fitted coefficients x_n and the raw sensor data:

$$\vec{x}_{\mathrm{raw}} = \mathbf{B} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \tag{4.7}$$

The previous equation may be solved for the coefficients using the SVD to generate a pseudoinverse of the basis matrix; this pseudoinverse called $\mathbf{A}$ has the property that multiplication by discretely sampled data computes a least-square estimate of the coefficients.

$$\begin{aligned} (U, \Sigma, V) &= \mathrm{SVD}(\mathbf{B}) \\ \mathbf{A} &= V \cdot (1/\Sigma)^{\mathrm{T}} \cdot U^{\mathrm{T}} \end{aligned} \tag{4.8}$$

The term $1/\Sigma$ represents a square matrix in which each element of the diagonal is the reciprocal of the corresponding diagonal element of Σ if that element is non-zero; zero values remain zero.

In operation, the rough time of the apex is found by observing the sign change of vertical velocity computed by the y estimator. After 30 more samples are collected, the coefficients are computed and the precision estimate of the state is formed:

$$\begin{aligned} t_p &= -y_1/(2y_2) \\ x &= x_0 + x_1 t_p + x_2 {t_p}^2 \\ y &= y_0 + y_1 t_p + y_2 {t_p}^2 \\ \dot{x} &= x_1 + 2x_2 t_p \end{aligned} \tag{4.9}$$

Since the basis function matrix was generated with symmetric time values, the apex time t_p is relative to the center of the filter data. The actual time of the peak is as follows:

$$t_{\text{peak}} = t_p + (t - 30\Delta t) \tag{4.10}$$

In this expression, t is the time when the fit is performed, i.e., the time of the last of the 61 samples.

4.3.3 Control Output

The controller computes values for thrust energy and leg angle that must be translated into servomotor commands. The desired leg angle ϕ, measured with respect to the vertical, is transformed to an angle in body coordinates u_ϕ using the body orientation θ and estimated velocity $\dot{\theta}$:

$$u_\phi = \phi - \theta - b_\omega \cdot \dot{\theta} \tag{4.11}$$

The servo command angle u_ϕ is then linearly scaled using measured scale and offset to a servo PWM command value. The desired thrust ξ is specified in terms of energy and is transformed to a servo value u_e using the inverse of the thrust model presented in Section 3.3.2.

The two motors are hobby servos that perform position control internally when supplied a PWM position command signal. The PWM is generated with a timer chip configured as a positive going one-shot that is software triggered on the 200 Hz sensor sampling cycle. The servo command can range between 600 and 2200 μsec before hitting internal limit stops; the timer circuit resolution is 0.5 μsec. The servo only supplies motor current when given a command pulse; when the hopper goes to sleep the pulses are no longer sent and the motors become backdrivable. Different command rates were evaluated; our operating rate of 200Hz has noticeably smoother motion than the typical 20–50 Hz.

4.3.4 Other Real Time Issues

A considerable effort was spent ensuring that the 200 Hz control cycle operates in hard real-time. The controller runs under the QNX operating system, a POSIX-compliant real-time UNIX variant for standard PC hardware. The controller runs as a separate process on a timer interrupt and only interacts with I/O cards and memory buffers to avoid filesystem and graphics delays. A separate process acts as a FIFO for buffering real-time debugging information and writing log files. The user interface is a third process that allows modification of most of the controller state using a semaphore-controlled shared memory; the user sees a complete readout of controller state and an animation of the motion and may interactively modify gains and issue commands. Note that this effort was necessary to ensure a consistent sensor sampling interval; the discrete control that only runs once during flight has considerable real-time latitude.

4.4 State Space Diagrams

The rest of this chapter discusses graphical methods for analyzing hopper trajectories. This is something of a shift from the previous subject but has this in common: this is a continuation of exploring how physics motivates a control approach. In this case, the discrete control model is well suited to diagrammatic analysis. The new element is consideration of terrain geometry: geometric representations of state can naturally include terrain constraints.

For the following few sections, the hopper is assumed to be an idealized hopper with an instantaneous bounce and perfect efficiency. For clarity it is assumed that at ground contact the hopper body is always distance l above the ground; normally that altitude is $l\cos\phi$. The perfect efficiency is an idealization of constant total energy hopping, with the graphical advantage that the hopper state is reduced to just two dimensions. The different diagrams choose different coordinate pairs to illustrate various situations, but the constraint is always as follows:

$$1/2 \cdot m\dot{x}^2 - mgy = E \tag{4.12}$$

In practice, most of the thrust energy is used to make up for leg losses and the dynamic range of the thrust mechanism is small compared to the combined kinetic and potential energy (the "total energy"). This means the hopper needs to accumulate energy over time to climb slopes and cross high obstacles. But for short term analysis, constant total energy is a reasonable local policy and easy to implement.

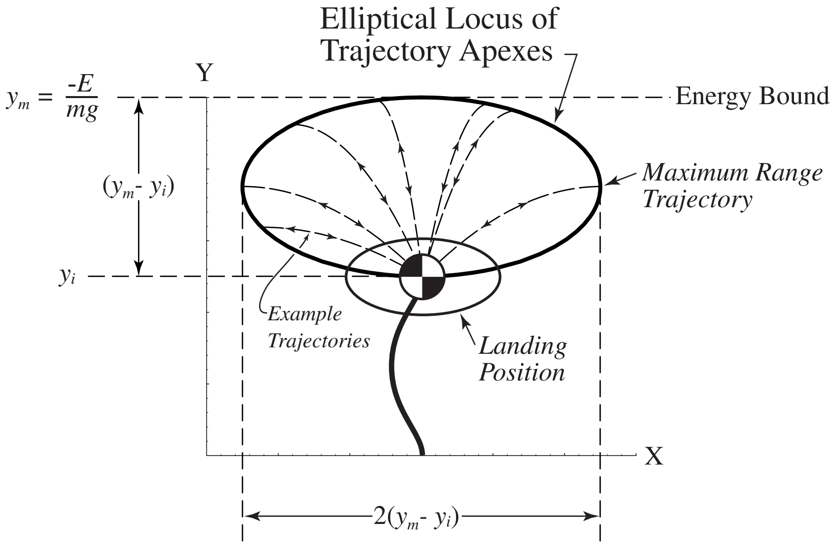


Figure 4.1: The elliptical curve is the locus of apexes of trajectories that land at the specified point; also the locus of trajectories reached taking off from the given point.

4.4.1 Umbrella Diagrams

An "umbrella diagram" is an illustration of a set of hopper trajectories plotted as the (x, y) locations of the apexes. They are useful as pictorial representations, but are ambiguous since each point represents a parabolic path but not the direction travelled along the path. The implicit constraint is the total energy constraint, which also imposes an upper bound on the allowable states at $y = -E/(mg)$; these states have lateral velocity of zero.

Figure 4.1 illustrates the set of states that land at a given point. The locus is an ellipse centered between the impact point and the maximum possible altitude and is twice as wide as tall. The same ellipse is also the set of states that are reachable from a given point; in this guise it is related to the solution to the artillery range problem. The full curve is shown but will sometimes be truncated on the lower side at the point where the trajectory lands or takes off at 45 degrees. On level ground these points are the maximum range points; in general the angles in the upper half are preferred since the hopper works much better "hopping" than "skittering." Since the energy is specified on an absolute scale, at lower landing elevations this curve is broader but the center point remains fixed at the upper energy bound; this can be seen in Figure 4.4. The illustration is symmetric and the directional ambiguity conveniently summarizes behavior for both directions. In the asymmetric cases to come the direction of travel will be explicitly defined.

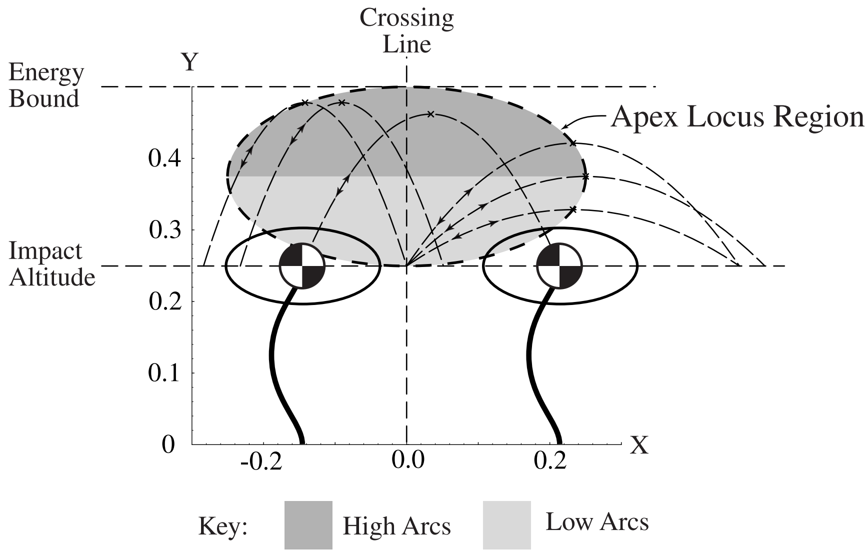


Figure 4.2: The shaded region is the locus of apexes of trajectories that cross the vertical line. Crossing is defined as a takeoff and landing on opposite sides of the line.

The curve may be derived as the simultaneous solution of the energy constraint and the lateral displacement as a function of the velocity and the falling time:

$$\begin{aligned} E &= 1/2 \cdot m\dot{x}^2 - mgy \\ \Delta x &= \dot{x} \cdot t_{\text{fall}} = \dot{x} \cdot \sqrt{2(y_i - y)/g} \end{aligned} \tag{4.13}$$

The solution may be succinctly expressed as a function of y; the sign ambiguity selects the right or left half:

$$\Delta x = \pm 2 \sqrt{\frac{(y_i - y)(E + mgy)}{mg}} \tag{4.14}$$

This may also be written in terms of the maximum altitude y_m; from this form may be derived the equation for an ellipse with the center at $(y_m + y_i)/2$, halfway between the maximum altitude and the impact position:

$$\begin{aligned} E &= -mgy_m \\ \Delta x &= \pm 2 \sqrt{(y_i - y)(y - y_m)} \end{aligned} \tag{4.15}$$

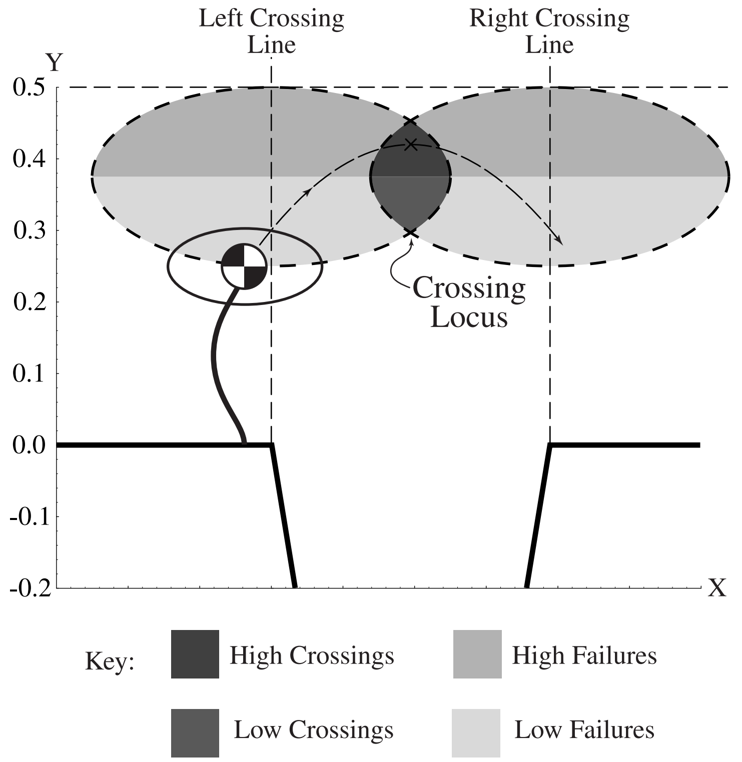


Figure 4.3: The dark region is the locus of apexes of trajectories that cross the hole. Crossing is defined as taking off from the left side of the left edge and landing on the right side of the right edge, or vice versa.

$$
\begin{aligned}
\Delta x^2 &= 4(y_i - y)(y - y_m) \qquad (4.16)\\
&= ((y_i - y_m) - (2y - y_m - y_i)) \cdot ((y_i - y_m) + (2y - y_m - y_i))\\
&= (y_i - y_m)^2 - (2y - y_m - y_i)^2
\end{aligned}
$$

$$
\frac{\Delta x^2}{(y_i - y_m)^2} + \frac{(y - (y_m + y_i)/2)^2}{((y_i - y_m)/2)^2} = 1 \qquad (4.17)
$$

The ellipse in equation 4.17 has a minor axis of length $(y_i - y_m)/2$ and a major axis twice the length of the minor axis. Both depend only on the geometry; i.e, the total energy is expressed geometrically and the gravity parameter doesn't affect the geometry.

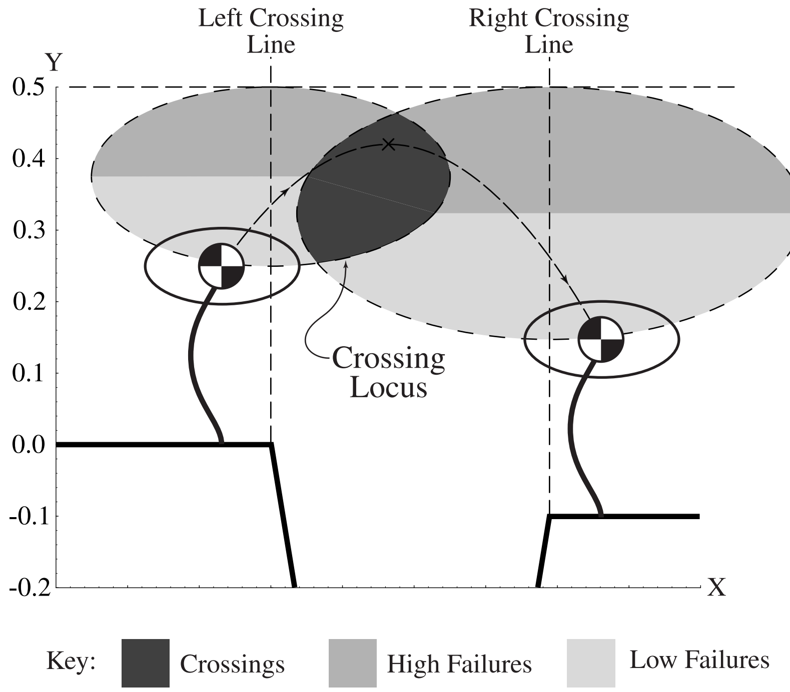


Figure 4.4: The dark region is the locus of apexes of trajectories that cross the hole. Crossing is defined as taking off from the left side of the left edge and landing on the right side of the right edge, or vice versa.

The first use of this diagram is Figure 4.2 which illustrates the set of states that cross a given vertical line. This is defined as the set of states for which the takeoff and landing body positions are on opposite sides of the line. This particular diagram is symmetric and may represent a crossing in either direction. A few particular solutions in the region are illustrated, but the region may be derived by considering the upper boundary, which represents states that lie exactly on the given line. The left portion might represent states that land moving toward the right; this segment is an open boundary of the region since all states between that curve and the vertical line are guaranteed to land on the right side of the line, and they must have taken off from the left side. Similarly, the right portion of the umbrella curve might represent states that took off from the line moving toward the right; this segment is an open boundary of the region since all states between that curve and the vertical line are guaranteed to have taken off from the left side of the line, and clearly land on the right side. The total region is the union of these two regions. The argument is symmetric for a left-moving hopper.

This result is applied in Figures 4.3 and 4.4 to the problem of determining which

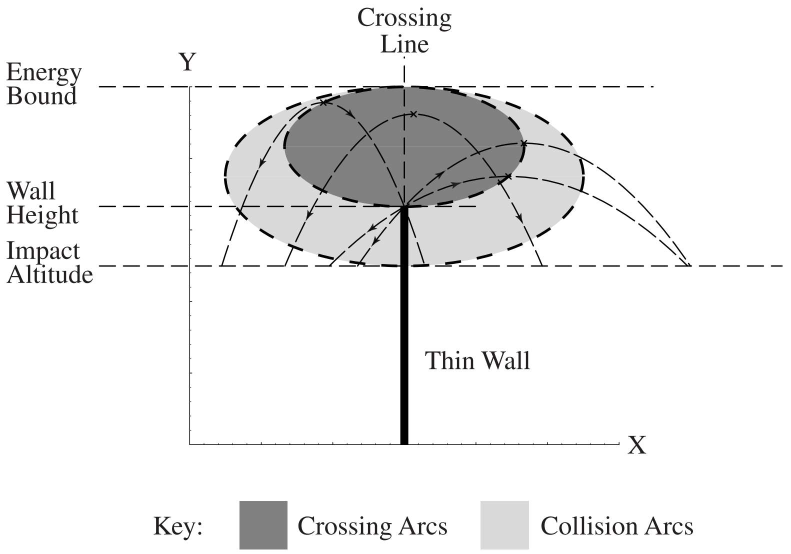


Figure 4.5: The shaded regions are the locus of apexes of trajectories that succeed or fail in crossing a thin wall. The hopper is treated as a point and the wall has zero thickness. Three arcs are illustrated that graze the top of the wall; these correspond to points on the boundary of the success locus.

states (at the specified energy level) may successfully cross a hole. Since the hopper shouldn't land in the hole, this may be reduced to the set of states that take off on the left side of the left edge and land on the right side of the right edge, or vice versa. In other words, the solution is the set of states that cross both edges in one flight, which is the intersection of the crossing regions for the two edges. This sense of "crossing" only considers body position; the hole in actuality defines a constraint on *foot* position. For example, there are extremal states with body positions *over* the hole not considered by this definition of "crossing." In general, the foot placement constraint involves not just the geometry but also the previous state: for the purposes of these diagrams landings and takeoffs are always assumed to have the foot placed straight down, since the diagrams are meant to illustrate the trajectories that satisfy the terrain constraint without illustrating whether those trajectories are achievable from other states.

Figure 4.5 extends the line crossing case to a thin wall. In the case illustrated the wall is higher than the impact height and there exist states guaranteed to collide with the wall. The body is assumed to be a point; i.e., this is a configuration space obstacle and the usual finite sized body requires growing the obstacles with the usual C-space methods. The crossing region shown may be derived by considering the states that pass through the point at the top of the wall: the physics is the same as the case where

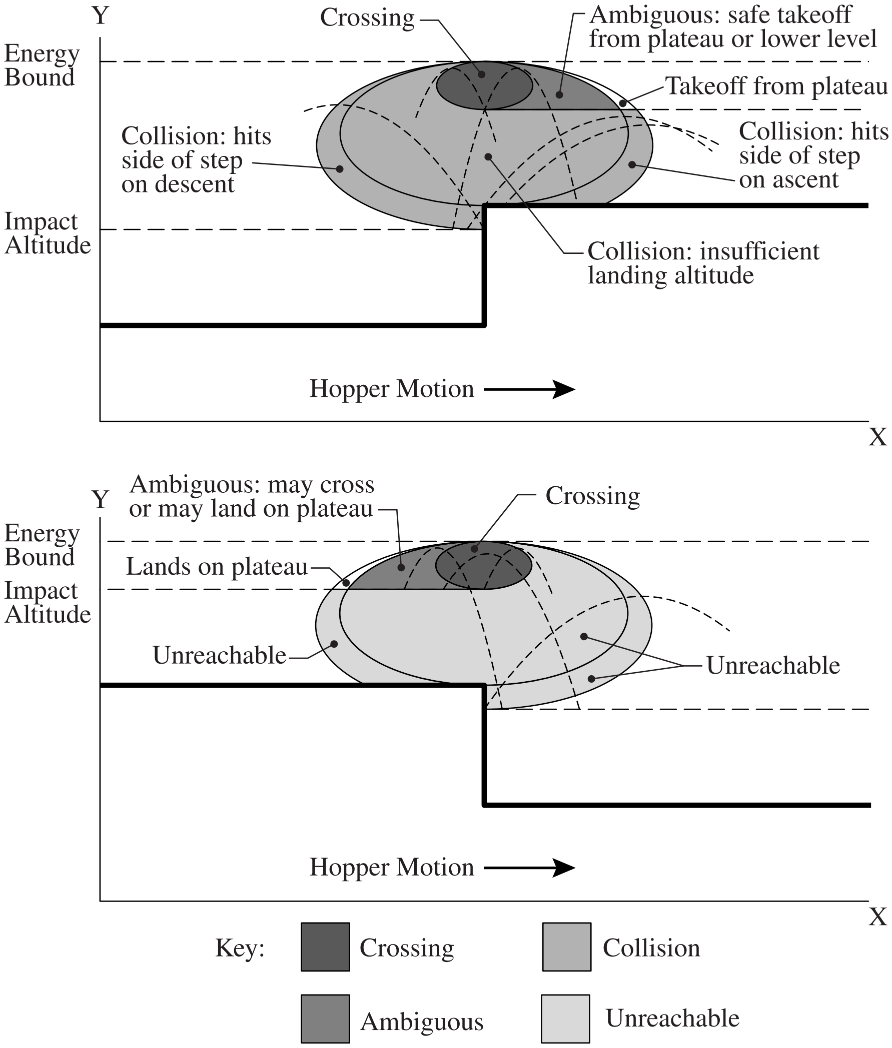


Figure 4.6: The shaded regions are the locus of apexes of trajectories that succeed or fail in crossing the step. Each diagram specifies the result for left to right motion, although the geometry is symmetric. The states outside the ellipses are not labelled but could be similarly categorized.

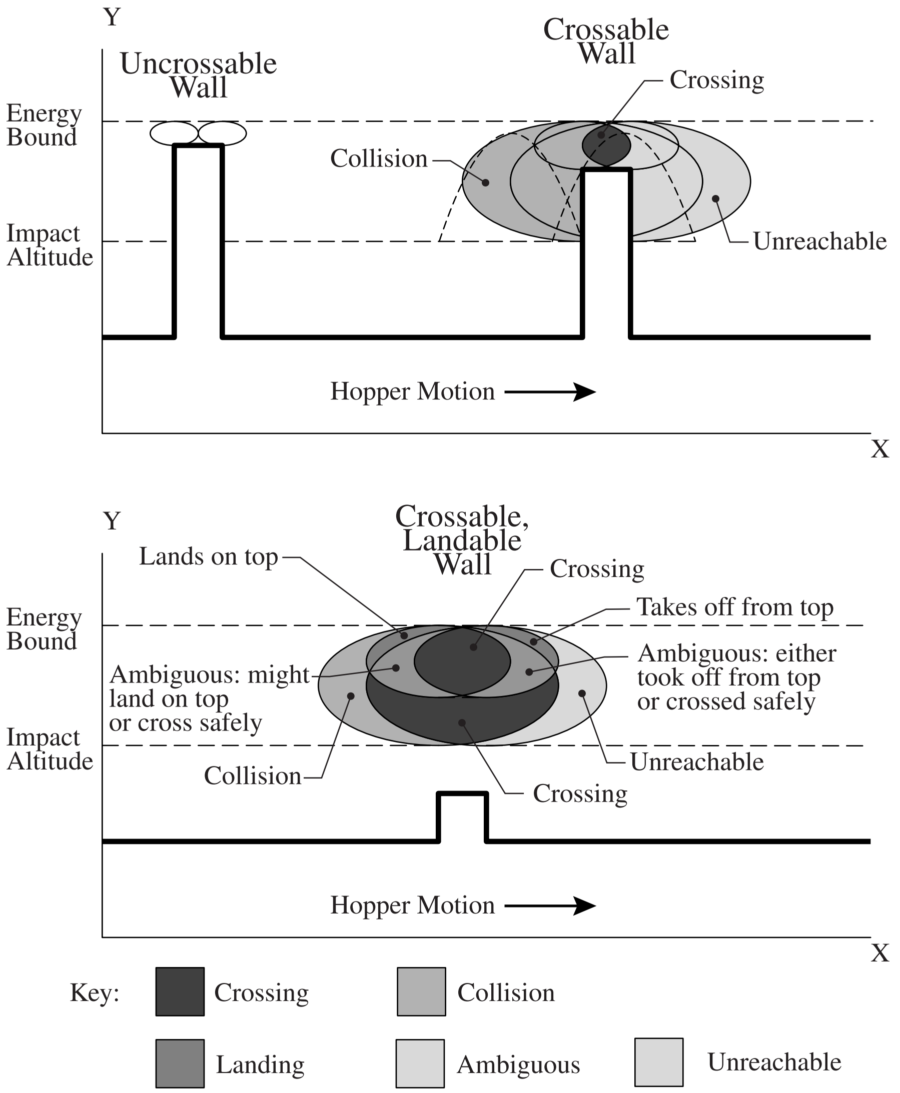


Figure 4.7: The shaded regions are the locus of apexes of trajectories that success or fail in crossing each wall. The top left wall is too high to cross, the top right may be crossed, and the bottom wall is low enough to cross or land on top. The hopper is treated as a point, and the walls have the thickness illustrated.

the body would land at that point, and so the same ellipse may be constructed. The ellipse for a ground landing at the same position defines all states that must either cross or collide, and the set difference yields the regions of guaranteed collisions. This diagram is symmetric and describes a crossing in either direction.

Figure 4.6 illustrates the possibilities for a terrain step that is low enough the hopper may hop on the plateau. The three ellipses are formed by the dimensions of the lower level impact, the upper level impact, and the clearance of the step itself. The small ellipse contains the states that unambiguously succeed. Adjacent to it is a region in which each trajectory has two possibilities for ground contact. In the upper diagram, the ambiguity results because the parabola crosses close enough to the step that the state could be achieved by a bounce from the lower level or by a bounce off the plateau itself. This is possible because the construction rules assume that the leg collision is not a factor: that particular parabola involves the body crossing the step within a leg length, and it is assumed the hopper may simply move the leg out of the way. In the lower diagram the ambiguity is symmetric; the parabola could be landed on either the plateau or the lower level, depending on whether the leg were placed down or swung out of the way during the descent. Also, in the lower diagram many states are labelled unreachable; in fact, any apex above the energy limit or below the impact altitude is unreachable so most unlabelled states are also in this category.

One fact this diagram does not make clear is the range of leg angles that may achieve a particular region from a particular takeoff. However, if one draws the reachable ellipse around any takeoff from the plateau, the crossing region presents a larger target that from the bottom, in the sense that a greater fraction of the possible trajectories result in a state in the region of success. In this sense it is easier to hop down than up: hopping up requires more precise control of the takeoff trajectory to land in the window between landing short and not achieving sufficient altitude to land on top.

The last example illustrated in Figure 4.7 shows three cases of finite sized walls; the first is uncrossable, the second crossable, and the third is low enough the hopper may either cross or land on top. Each of these diagrams is essentially a superimposed pair of step climb and step descend diagrams, and the ambiguities result for similar reasons. As before, "landing" is defined only with respect to a leg pointing straight down, and it is assumed the leg may swing clear of obstacles during flight.

4.4.2 Discrete Phase Space Diagrams

The directional ambiguity of the umbrella diagram may be avoided by instead plotting the locus of trajectory apexes on $(x, \dot{x})$ axes. Normally, this would be a phase plot, but unlike a phase plot each line on this plot does not represent a trajectory, but represents a set of trajectories, one for each point on the line. Like the umbrella

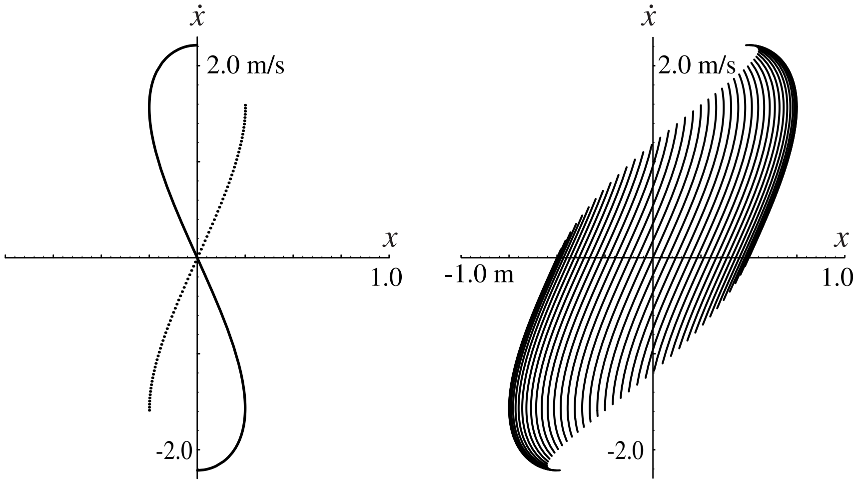


Figure 4.8: A pair of discrete "phase" diagrams illustrating states reachable after one and two bounces. The left diagram illustrates the states that land at the origin as a solid line, and the states reachable from the origin as a dotted line. The dots are spaced 0.01 radians of ϕ apart, but not every final state is reachable from every initial state. The right diagram illustrates the region of states reachable from the origin after two bounces.

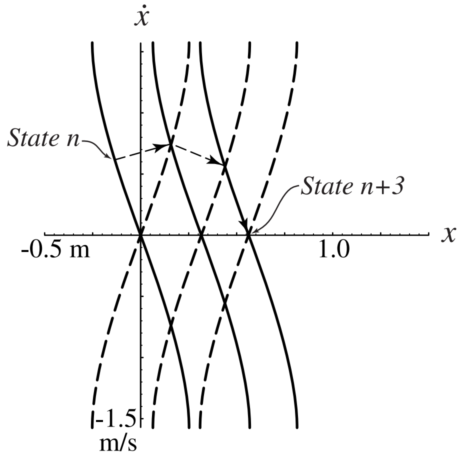


Figure 4.9: A discrete "phase" diagram illustrating a three bounce trajectory ending with a vertical takeoff. The trajectory is represented by transition arrows. State n lies on the landing curve for the origin; the corresponding takeoff curve represents a superset of reachable states. The hopper takes off with slightly higher lateral velocity, and the second landing point is determined by intersecting the new state and the landing curves. This process continues until State $n + 3$, which takes off at the zero lateral velocity point on the takeoff curve and will land at the same point.

diagram, the total energy bounds the area of admissible trajectories.

The locus of trajectory apexes that land at a given point is derived from the same constraints of equation 4.13, but eliminating y instead of $\dot{x}$:

$$\Delta x = \dot{x}\sqrt{\frac{2E - m\dot{x}^2 + 2mgy_i}{mg^2}} \tag{4.18}$$

This may be recast without reference to the y axis by stating the energy in terms of the maximum lateral velocity possible from the takeoff:

$$\begin{aligned} E &= 1/2m\dot{x}_m^2 - mgy_i \\ \Delta x &= \sqrt{\frac{\dot{x}_m^2\dot{x}^2 - \dot{x}^4}{g^2}} \end{aligned} \tag{4.19}$$

The left half of Figure 4.8 illustrates the pre-image and post-image of a landing point. The pre-image is the set of trajectories that land at that point, and the post-image is the set of trajectories reachable from that point. In the figure, the complete pre-image is shown but only a segment of the post-image in order to emphasize that the set of states available after takeoff is constrained by the range of possible leg angles. Each point on the landing curve is associated with a segment of the takeoff curve; the extent of that segment is a function of the impact velocity and the foot friction at that landing. The points on the takeoff curve are plotted assuming a vertical fall ($\dot{x} = 0$) and are plotted for values of the leg angle spaced 0.01 radians apart.

The right half of Figure 4.8 extends that case to two bounces. Each curve represents the choices available on the second bounce; the entire region is the set of states reachable after two bounces. The first leg angle is plotted at discrete 0.02 radian intervals and the second bounce leg angle plotted continuously.

Figure 4.9 represents a three bounce sequence over level ground. In this case only the trajectories near vertical have been plotted. The diagram may be traced as follows: State n is on a curve that intersects the x axis at the landing position. From the landing position is plotted another curve illustrating all possible reachable states; it is assumed a leg angle exists to put the hopper on the trajectory represented by the point at the end of the first arrow. This chain is iterated for two more bounces. State $n+3$ represents a trajectory with a vertical takeoff (i.e., zero lateral velocity $\dot{x}$) and returns the hopper to the same position.

This diagram offers a different view of the state transition process, but does not illustrate terrain geometry as literally as the umbrella diagrams. As shown in equation 4.19, the terrain altitude does enter into the curves via the impact altitude y_i; the lower the altitude, the greater the range of possible lateral velocities and the taller the takeoff and landing curves become.

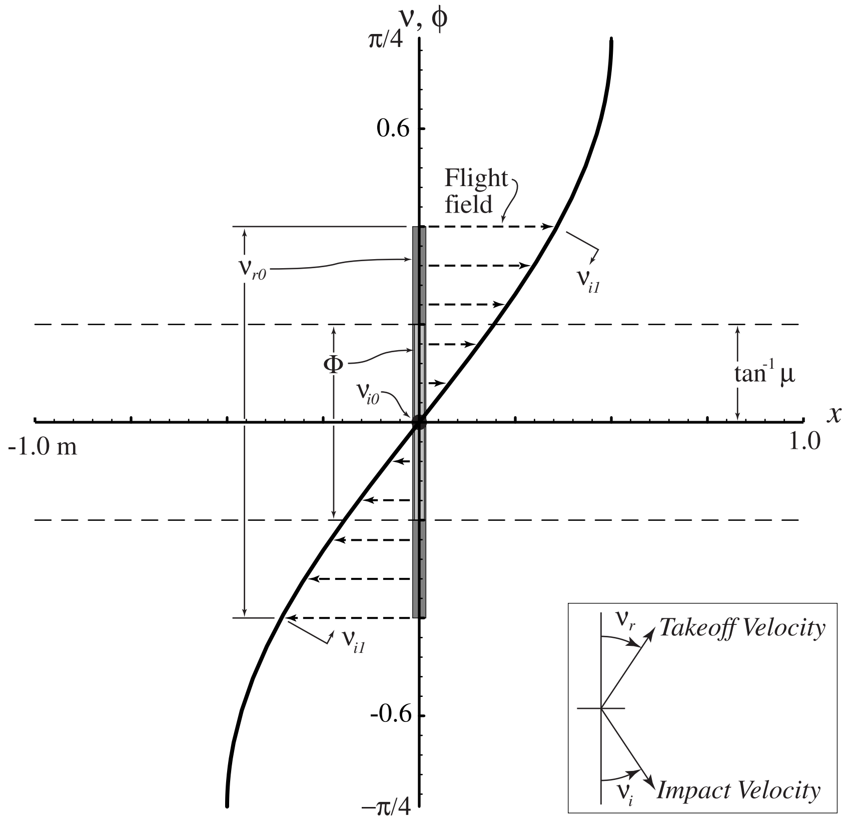


Figure 4.10: A reflection angle diagram illustrating states reachable from initial state ν_{i0}, given the set of leg angles Φ. The set of possible takeoff states is ν_{r0}, which is mapped by flight to the impact states ν_{i1} on the curve. The flight map illustrated assumes the landing altitude is the same as the takeoff altitude.

4.4.3 Reflection Angle Diagrams

The diagrams of the previous section illustrate the physics but fail to show the control choices clearly: the range of leg angle at each impact is not represented in the diagram. An alternative is the *reflection angle* diagram which parameterizes the impact rather than the trajectory. Given constant total energy, a trajectory may be described by a takeoff position and takeoff angle; this may be easily graphically transformed to the successive impact, and the leg control range plotted on the same axes. These diagrams are best suited for describing paths over level ground with holes.

The angle convention is illustrated in the inset in Figure 4.10. The impact angle ν_i is measured with respect to the $-y$ axis; a vertical fall has a value of zero, and a fall moving to the right has a positive value. For convenience, the rise angle (liftoff angle) ν_r is measured in a left-hand sense with respect to the $+y$ axis. A vertical liftoff has a value of zero, and a liftoff moving to the right has a positive value. Given the instantaneous bounce, the bounce function is as follows:

$$\nu_r = \nu_i - 2\phi \tag{4.20}$$

The flight function is considerably messier:

$$\begin{aligned}\nu_i \;=\; \operatorname{atan2} \Big(& \sin\nu_r \sqrt{2E/m + 2gy_r}\;, \\ & \sqrt{2g(y_i - y_r) + 2\cos^2\nu_r(E/m + gy_r)}\;\Big)\end{aligned} \tag{4.21}$$

$$\begin{aligned}\Delta x \;=\; & 2\sin\nu_r \sqrt{(E/m + gy_r)\left(\frac{y_i - y_r}{g} + \frac{(E + mgy_r)\cos^2\nu_r}{mg^2}\right)} \\ & -(E/mg + y_r)\sin 2\nu_r\end{aligned} \tag{4.22}$$

Figure 4.10 presents a reflection angle diagram for a vertical fall and a flight. The vertical axis indicates angle and the horizontal axis is the lateral position of the body at impact. The hopper impacts with ν_i equal to zero. The range of leg angles within the friction cone $\pm\phi_\mu$ is illustrated; after the bounce, the takeoff range spans twice that range: $\mp 2\phi_\mu$.

In this diagram a takeoff range is always a vertical line at the same x coordinate as the impact point. The effect of flight is to "shear" this line into a curve representing how different takeoff angles land at different x positions. On level ground, the impact angle is equal to the takeoff angle for a given flight ($\nu_i = \nu_r$), so this curve maps points on the vertical line directly horizontally to a corresponding impact point.

4.5 Discussion

This chapter is something of a collection of odds and ends; the hopper is so simple it was easy to put together a linear controller, so most of the interesting control work falls in the planning chapter. But the theme of the chapter is that discrete physics leads to discrete solutions, whether in control or in analytic method. This chapter is intended as a prelude to the planning chapter to motivate the development of search based planning and to illustrate the underlying search space. The diagrammatic methods are not useful to the machine per se but might help the reader to understand the control problem.

Chapter 5

Planning

The Bow Leg Hopper was conceived with real terrain in mind. It has the efficiency to use on-board power and enough energy storage to hop several times its own height, important attributes for jumping up and down steps and bounding between rocks. We believe the simplicity of the design will mean that the reliability in the lab will translate to ruggedness in an outdoor environment.

The big question is whether the device can be controlled with enough agility. Ideally, a controller would always place the foot on a secure foothold and always direct itself to a trajectory that leads to other footholds and eventually to the goal, working from noisy sensor data about the shape of the terrain and the state of the robot. What this thesis represents is work toward this ideal.

Chapter 4 discusses a simple linear controller suitable for level ground. However, for crossing terrain the "control" for the hopper looks more like "planning" than typical closed loop feedback. It uses graph search to consider sequences of footfalls that satisfy terrain, friction, and task constraints. The control outputs for each step are computed using the closed form model of the physics presented in Chapter 3. The planner computes paths in real time while the machine is in motion and can produce a partial plan in the few hundred milliseconds available during flight. As the hopper approaches the ground, it configures the mechanism using the best available plan. After the impact, the trajectory is measured again and the planner decides whether it may apply feedback control to stay on the plan or must replan for the next step. The thesis doesn't include the problem of sensing terrain from a moving vehicle, so the planner has an accurate model of the terrain available and high-quality position information from the constraint boom.

The following sections are organized from general abstract issues to the specifics of the implemented system. At the end is a discussion and conclusion section and suggestions for future work. Readers who wish to skip the pedantic course may examine the following road map:

- Section 5.1 develops the formal specification for a hopper planner.
- Section 5.2 outlines general solutions and system design issues.
- Sections 5.3 and 5.4 presents the planning systems implemented and tested in the laboratory.
- Section 5.5 presents ideas for offline planning.
- Section 5.6 wraps up with a summary discussion.

5.1 Formulation

Section 4.1 developed the general form for any hopper controller. This is reproduced as follows:

$$(\phi_{n+1}, \Delta E_{n+1}) = f(x_n, y_n, \dot{x}_n) \tag{5.1}$$

This is graphically depicted in Figure 5.1. A more planner-centric view is to view the plan as a map from one trajectory to another:

$$(x_{n+1}, y_{n+1}, \dot{x}_{n+1}) = P(x_n, y_n, \dot{x}_n) \tag{5.2}$$

This formulation was inspired by [BRK99]. This map abstracts the details of the bounce itself and defines the local goal for every trajectory in the domain; successive trajectories presumably chain to the goal. The control action for each state is implicit in the successor state and may be computed by inverting the physical model. This map may be visualized as a vector associated with each point that represents a one bounce transition to another state. In the planning search, this map is discretized as a graph with nodes representing sample points and arcs representing these vectors.

For the purposes of this discussion, the procedure that generates a plan P is the planner, and the procedure that uses plan P to compute a specific transition vector is the controller. Together these form a real time control system that uses planning methods.

This is a very general formulation intended as a starting point for discussing the specific planners tested in this thesis. These planners all search out specific paths in state space, then use a linear controller to accommodate modest deviations from the path. The combination of a sequence of specific states (a single path) and a linear controller may be viewed as a map defined over isolated points. This map has a domain surrounding each point (each state) whose extent is the domain of the linear controller. As the hopper moves through states the linear feedback should converge

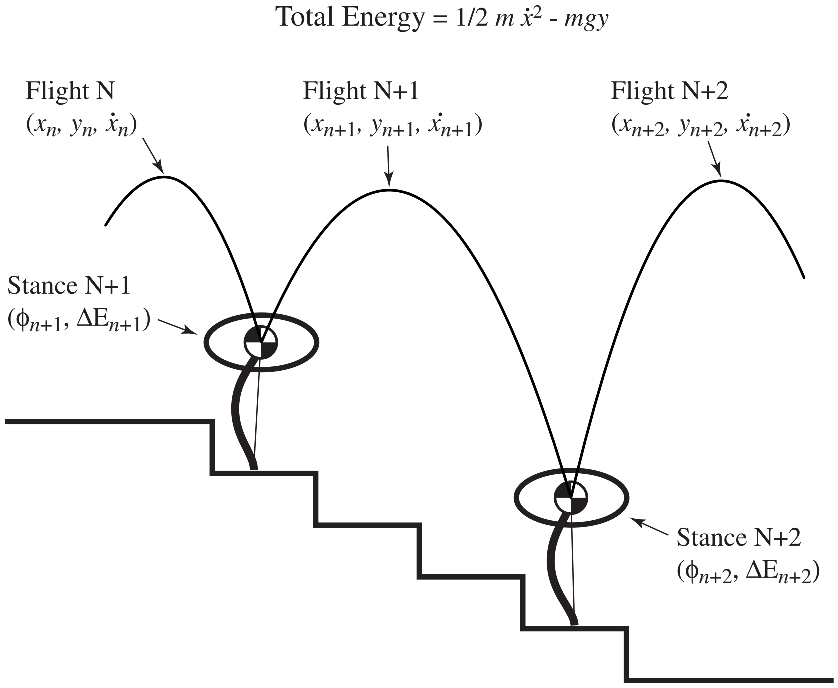


Figure 5.1: Illustration of the hopper state as seen by the planner. Each flight parabola is represented by three apex parameters. Each stance is a transition between flights controlled by the leg angle ϕ and energy change ΔE. The path shown has constant total energy so the hopper reaches approximately the same altitude even as the terrain drops away.

on the planned sequence. This particular description of the problem was inspired by [BRK99].

5.2 Planning Issues

This section enumerates the physical constraints on plans in order to motivate the planner design. The diagrams of Section 4.4 illustrate some of the physical constraints but are not sufficient to write a planner, and the planning formulation of the previous section offers little hint as to how such plans may be produced. This is a general discussion of issues; many of these thoughts were not directly tested by the actual planner, but all were considered.

The planner chooses a path in a state space in which each flight trajectory is represented by a point. The set of trajectories reachable from that point is a manifold parameterized by the two physical controls. The shape of that manifold is the combination of all the physics and constraints: leg kinematics, bounce transition function, flight dynamics (principally gravity), terrain shape, friction limits, and mechanical limits.

This is an open formulation, but is better thought of as a form of foot placement planning than route planning; some other approach would be better suited for choosing the long-term path across a large swath of terrain. So given that some other agent will hand down short term position goals the problem is finding a sequence of foot placements that get to a position and is tolerant of error during execution. This approach is open to arbitrary path constraints that might express auxiliary goals such as performing a particular gait during the sequence.

5.2.1 Foot Placement Constraints

The most basic constraint on a candidate path is the quality of each foot placement. The kinematic element is whether the terrain geometry and the joint limits allow contact, and the dynamic element is whether the leg forces can be supported by the strength and friction of the contact point.

One way to visualize the geometry is to draw a open-side-up semicircle with a radius equal to the leg length around every point on the flight trajectory, and record which positions intersect usable ground. On flat ground, every angle in the semicircle is geometrically feasible.

The planner measures the leg angle with respect to the world coordinate system. The joint limits on the leg, however, are with respect to the body — if the body pitch rotates enough, leg angles that are otherwise safe might be outside the limits of joint travel. If the hopper is successful at keeping the body pitch near zero, the leg limits are generally wider than the friction limits and don't enter into planning; this work assumes that the body pitch is not an issue and doesn't consider joint limits.

The friction constraint specifies which leg positions are likely to lead to a success bounce without the foot slipping. For the Bow Leg, this is especially simple: the freely pivoted hip means that the leg force is always directed along the axis between the foot and the hip. Assuming a Coulomb friction cone, the leg angle must always stay within $\arctan \mu$ radians of the normal to the surface. For level ground, the normal points vertically; for example, a friction coefficient of 0.5 means the leg angle must stay within about 27 degrees of vertical or the static friction force won't be enough to keep the foot from slipping.

The actual footfall occurs within a distribution that depends on the error in the model, the trajectory measurement, and the leg position control. The safety of a foot placement depends on this distribution and distance of the predicted landing from nearby hazards. Foot placement safety is an estimate of risk that is difficult to quantify. Furthermore, errors in the friction estimate are quickly catastrophic if the planner chooses overly aggressive velocity changes since the foot will slip. Without resorting to explicit measurement of failure probabilities, there are necessarily heuristic choices during planning to balance risk. A perfectly conservative hopper would be stuck hopping in place, since it must assume no foot friction is available (small leg angles) and that the physical model is wrong (minimize velocity).

In this planning formulation the most restrictive physical constraints are on foot placement. These actually constrain a combination of a "state" and a "control"; it might be convenient if the foot position could be used as a state variable, but the resulting state definition would not be of minimal dimension. Also, there exist physical states without a uniquely defined foot placement, such as the flight states of the ambiguous region of Figure 4.6.

5.2.2 Energy Constraints

The other control in our planning model is thrust, which in the Bow Leg design directly controls kinetic energy delivered to the body during stance. For the most part energy control and direction control may be treated independently. The energy may be planned as a function of terrain and only needs to be tightly coordinated to path planning during dramatic elevation changes. The advantage of this decoupling is that most of the time the planner need only consider two one-dimensional problems, which is generally more efficient than a single two-dimensional solution.

The planner lumps passive dissipation and control thrust together as an "energy control" whose value may be negative. The energy dissipation is a function of state, so the limits on this control reflect both mechanical limits and the state.

The "total energy" of a state is the sum of potential and kinetic energy at any point. Since gravity is conservative the total energy is constant for a given flight parabola. Total energy is a convenient planning metric because it can remain constant over a path, including those with lateral velocity changes. Paths at constant total energy have peaks at lower altitude as the lateral speed increases; a path hopping down a slope at constant lateral speed reaches the same altitude on every flight, so the flight time becomes progressively longer.

The mechanism limits both the maximum energy that may be added using thrust and the maximum energy that may be stored during impact; for the planning problems in the lab, the planner may safely ignore the storage limit, since the thrust limits the altitude that may be reached and there are no deep holes in the floor. Section B.2 includes a brief analysis of thrust motor power.

Each terrain feature defines minimum and maximum bounds on the total energy required to cross that point. As a base case, on level ground the minimum is the potential energy at standing height (with which the hopper would be taking infinitely small steps), and the maximum is the altitude with potential equal to the leg storage limit. The diagrams in Figure 4.7 illustrate examples of crossing walls for a specified constant energy; under the assumptions of the diagram the minimum crossing energy is that for which the crossing region shrinks to a point, and the maximum crossing energy that for which every state in the crossing region represents a path that exceeds the mechanical limits on impact.

However, the assumptions of the umbrella diagram do not take into account leg angle. For example, the lowest energy crossing for a hole in otherwise level ground is a 45 degree takeoff with the foot placed on the edge and the leg at the edge of the friction cone; the body is actually over the hole during stance, and lands with the foot extended to the opposite edge. A step is minimally crossed by a near-vertical takeoff close to the base of the wall that lands with the foot extended to the rim. Note that these energy bounds do not guarantee that the "success" states don't collide one bounce later, let alone achieve the goal.

5.2.3 The Clock

If the planner assumes the terrain to be static then the plans are time-independent. However, energy storage does impose a time constraint on plan execution. It is mechanically advantageous to store energy at low power, so the process continues through the entire flight. The energy released at bounce N must be stored starting immediately after bounce $N-1$. This requires some anticipation; during flight $N-2$ the measurement is made to determine the energy stored during flight $N-1$, which affects flight N. This is acceptable because the energy dissipated at each impact may be reasonably predicted. Note that the plan itself only needs to specify the energy released after a given state.

The more difficult time constraint is on the planning process itself. It takes some initial portion of flight to measure the state, and some final portion to move the leg; if the planner is to use the most current state when generating a new plan, it has only the short interval in between in which to come up with the first step of the plan. Computation may continue during stance and beyond, although by that point the first step of the plan will have been committed. If a planner can anticipate a future state and begin planning before it is reached, much more time is available.

If dissipation were zero then the idealized physics would be time-reversible. However, even with some dissipation paths may be time-reversible if the difference in energy between adjacent states is both less than the dissipation limit in one direction and less than the thrust limit in the opposite direction; this may be trivially satisfied by maintaining constant energy so the difference is zero. By reversing the start and goal a forward-time planner could be used to plan paths backwards from a subgoal.

Even with just forward-time planning, paths can be chosen that may be reversed at every bounce. This could be used as a strategy for hopping across unknown terrain: if at every bounce a hopper could guarantee that it may reverse direction at the next bounce, it would guarantee it could "back out" along the same path.

5.2.4 Terrain

The planning process is limited by the available terrain information. On the coarse scale, a limited range of perception limits the planning distance. On the fine scale, limited precision of perception limits the planning precision.

For this work the planner assumes the terrain geometry and friction coefficient are known. (In the experiments the terrain was configured to match the model.) In general practice either might be perceived incorrectly; in particular, remotely estimating friction requires perception of material type, a generally unsolved problem. However, only discrete points of the terrain ever come into contact with the foot so "false negatives" that incorrectly declare a foothold unusable are tolerable if footholds are otherwise sufficiently dense.

Another property that might be worth considering during planning is impact dissipation. The hopper performs best on rigid terrain; a segment of high-dissipation terrain reduces the dynamic range of the hopper and imposes a condition on the minimum initial energy that could successfully navigate the segment.

One minor point: every trajectory "collides" with the terrain eventually: there are no infinitely deep holes. On ascent, a trajectory may be evaluated to see if it collides, but on descent, it is the combination of a trajectory and a leg angle that determines whether it collides or makes safe contact. In other words if the time-until-bounce is less than the time-until-collision, that control is safe for that trajectory.

Most of this discussion of constraints could be applied to the case of a 3D hopper. However, planar geometry makes global proofs simpler. The 2D geometry implies that every x coordinate between the start and goal must be crossed. Barring two-level bifurcations (multi-platform geometry), this defines either a vertical ray or segment for each x coordinate that must be crossed, depending on whether there is an overhead obstacle in addition to the ground. The set of states that crosses each vertical ray is defined by Figure 4.2.

5.2.5 Graph Search Planning

Up to this point, the discussion of planning has been independent of any particular algorithm. This thesis seeks to demonstrate the possibilities and limitations of applying ordinary graph search techniques to the foot placement planning problem.

Graph search is hardly ideal. The problem does not define a symbol system; it has discrete transitions but all state quantities are continuous, so trajectory and control discretization is necessary. By the same token, graph search is amenable to heuristics and generates solutions incrementally. The continuous generation of valid partial solutions is well-suited to on-line planning in order to always have *something* to do when time runs out during descent.

Discretization

The problem is naturally discretized in time since each bounce represents a discrete choice. Some variations on the energy mechanism add energy in specific quanta, but otherwise both leg angle and thrust energy controls are continuous. A sampling of the continuous range of feasible leg angles and thrust must be selected at each bounce for expansion in the search; the number of samples is the search branching factor.

There are some natural discretizations: terrain can often be treated as patches for which any landing is roughly equivalent. This defines two dimensional patches in the state space with corresponding two dimensional patches in the control space.

If when near the goal the control discretization does not include a choice that reaches the goal state, then the plan will not reach the goal exactly.

Heuristics

The hopping task is specified through the search heuristics. It is the estimate of the cost to the goal (the cost-to-go) that determines which paths are explored and the cost of partial paths that determines the relative ranking. Since the basic task of hopping to a location is underconstrained, any gait choices or "style points" are expressed in the relative cost of different motions.

The fundamental search heuristic is the measurement of path cost and the associated estimate of cost-to-go. Ideally, cost would be measured in some real unit that could be measured and normalized. One possibility is measuring energy expense. However, the energy cost for falling down is infinite since the hopper cannot get up and hence would never reach the goal. Since some heuristics encode risk probabilities (e.g., chance of the foot slipping), the non-zero probabilities multiplied by infinite costs would lead to all paths having infinite cost. Even if the hopper could get up from a fall, any terrain requiring the crossing of a deep hole would again involve infinite path costs. Such a planner would be entirely risk-averse; encoding cost encodes a life philosophy.

5.3 A Planning Solution

This section discusses the final version of the planner "as-built." Following this is a discussion of its predecessor, which was more elaborate but illustrates some other useful ideas.

5.3.1 Algorithm

This following algorithm is a straightforward best-first search in which each node represents a path ending on a flight state, and each arc represents a bounce (i.e., a transition between flights). Much of the behavior is wrapped up in the heuristic functions.

To initialize the search, create a priority queue with a single node that represents the current flight state. Then iterate the following until termination:

1. Remove the highest-score partial solution N from the head of the priority queue. If N is null, return the best solution found so far. If N satisfies the goal, return N.

2. If the depth of N in the search tree is within a depth limit (i.e. the number of bounces below a threshold), expand the node as follows:

   (a) Test the result for each of five evenly spaced leg angles between ± 0.38 radians:

      i. Compute the landing position for N and ϕ.
      ii. If the foot lands on a non-level segment then return failure, else continue.
      iii. Compute the approximate result of the bounce based on half-max thrust.
      iv. Compute total energy of resulting flight, the difference from the desired total energy, and the energy to be added on this bounce (the thrust control value).
      v. Recompute the bounce using the newly computed thrust control.
      vi. If the takeoff is within 45 degrees of vertical, return success, else return failure.

   (b) If the bounce was successful, create a new search node representing the new flight state, compute the cost heuristics and a score, and add it to the queue of unexpanded nodes.

   (c) If the new node has a score higher than any found so far, record it as the new "best so far" solution.

3. If memory for tree storage is exhausted, return the best solution found so far, else iterate.

This algorithm was chosen because it is unbiased and simple. New nodes are back-linked to their parent nodes so many plans may be stored in limited memory. It is simple and brute force, however, and its performance is sensitive to the hand-tuning of the heuristic functions.

The evaluated cost of a partial path (a node) is the sum of costs for each bounce (each arc). The score for a node is the negation of the sum of the cost and the estimated cost-to-go. The cost of a bounce is the sum of a friction penalty, a "verticality" penalty, a foothold risk penalty, and a constant cost:

$$\begin{aligned} \text{score} &= -(\text{cost} + \text{cost to go}) \\ \text{cost} &= \text{cost}_f + \text{cost}_v + \text{cost}_r + 1.0 \end{aligned} \tag{5.3}$$

The friction penalty is a smooth function that rises quickly as the leg angle nears the edge of the friction cone for the terrain at a particular placement. The variable ϕ_{norm} is the leg angle normalized to $[0, 1]$ in which the value 0 is the center of the cone and 1 is the friction limit; the cost is as follows:

$$\text{cost}_f = 5 \cdot {\phi_{\text{norm}}}^6 \tag{5.4}$$

The numeric constants in each of these expressions were chosen through trial and error, evaluating the planner on a set of sample problems.

The "verticality" penalty favors bounces that take off closer to vertical. The motivation is that the more horizontal takeoffs involve more leg sweep and are harder to predict; this penalty favors paths for which the the models are more reliable. A side effect is to limit the lateral speed for a given hopping altitude: to go faster across the terrain the hopper also needs to go higher. The cost is a smooth function of the ratio between the tangent and normal velocity of the body expressed in the leg frame:

$$\text{cost}_v = 20 \cdot (v_t/v_n)^6 \tag{5.5}$$

The foothold risk penalizes foot placements near the edge of safe footholds. This prefers landing well inside a region to accommodate the inevitable trajectory error. The variable $\text{footx}_{\text{norm}}$ is the normalized position of the foot which takes the value 0 at the "safety limit" 6 cm from the edge of the terrain segment and the value 1 at the edge. Foot placements inside the safe region have a normalized value of 0.

$$\text{cost}_r = 20 \cdot {\text{footx}_{\text{norm}}}^2 \tag{5.6}$$

The primary factor in the estimate of the cost-to-go is the number of bounces remaining to the goal, under the assumption that there exists a locally safe path that minimizes the risk factors so the constant bounce cost dominates. An admissible heuristic in this case is one guaranteed to underestimate the cost and hence overestimate the average speed. A tunable compromise is to compute the maximum speed allowed by the desired total energy and scale it by a hand-tuned factor.

If the best path travels at uniform velocity, then the estimated number of bounces to the goal is the distance divided by the product of the maximum lateral velocity and the time of flight.

$$\text{cost to go} = k_{\text{xcost}} \cdot |x - \bar{x}|/(\dot{x}_m t_f) \tag{5.7}$$

The variable $\dot{x}_m$ is the maximum lateral velocity, computed from the desired total energy, t_f is the estimated time of flight, and k_{xcost} a heuristic scaling factor, typically around 4 or 5; this indicates the average lateral speed for the test planning cases generally involved about four times as many bounces as the maximum velocity across level ground.

5.3.2 Plan Execution

The plan is consistent with the model of the physics but is not naturally stable. The sources of uncertainty that lead the hopper off the plan include systematic error in the physical model, mechanical backlash in the leg servo, error in the state estimation, and friction and backlash in the constraint boom. After each impact the controller computes an adjustment to the plan for the next two impacts intended to return to the planned trajectory. If the error is too large, the controller abandons the plan and begins creating a new one from the measured state.

The leg angles $\phi_1..\phi_n$ at n successive impacts may be considered a vector that defines the reachable trajectories. In general, a trajectory is defined by three parameters and three successive impacts may span the trajectory space. However, hopping at constant energy reduces the trajectory space to two dimensions. Thus a deviation from the path can be corrected by adjusting two successive leg angles to reattain the planned trajectory. The correction combines linear feedback and feedforward computed using the physical model.

If the corrected foot placement falls outside a safe region defined around the planned foothold, the controller cannot guarantee the safety of that bounce and a new plan is generated. Planning occurs concurrently with execution; the planning system is an anytime planner and computes usable partial plans immediately. When starting from scratch, the best plan available before impact is used, but is then refined during the remainder of the hopping cycle. Once completed, the plan is used until accumulated error forces a replan or the plan is exhausted.

The energy of the hopper is regulated using a feedback loop that varies thrust to maintain a constant total energy. The hopper is designed so that the dissipation is relatively independent of forward speed. The planner estimates the operation of this controller so that initial energy ramp-up or ramp-down will be correctly treated.

Several hopping performances are presented in Chapter 6, and some statistical data on the performance of the planning controller are presented in Section 6.3.

5.3.3 Comments

The planner requires a limit on search depth since the goal may be arbitrarily far away. This planner just keeps searching until it runs out of time or memory; the "best" path with the lowest sum of cost and cost-to-go is the one returned. Usually this is a partial path headed toward the goal.

But how far may the path be trusted at all? As an open-loop plan the error accumulates so fast that more than a bounce or two is just fantasy. Closing the loop during execution means the plan is a reference path around which to control. This raises the question though, of how tightly a reference path needs to conform to details of the hopper physics. It is possible the dynamic limits of the machine could be abstracted to some simple constraints on path geometry. A purely geometric planner could then fit parabolas across the terrain, only approximately matching the end of one to the beginning of the next.

5.4 Other Solutions

During the course of its evolution the planner was continuously refined, but three distinct points in the progress are worth describing. The planner presented in the previous section was the third major revision that resulted from streamlining the more elaborate second version. However, the second version incorporated some features that might be useful under particular circumstances, which are described in the next section. Following that is brief description of the first revision for historical interest.

5.4.1 Second Revision Planner

The second planner revision is similar in most respects to the third. The chief difference is the algorithm for choosing the discrete set of leg angles to be evaluated when expanding a node. An example result is illustrated in Figure 5.2 in which node n is expanded to successors a, b, and c that will each land on a distinct terrain segment, i.e., the middle stone, the narrow right stone, and the safe region on the right. This was generated using the following ideas.

One feature of a terrain model based on line segments is that each foot placement region has an identity and is ordered with respect to the other segments. When

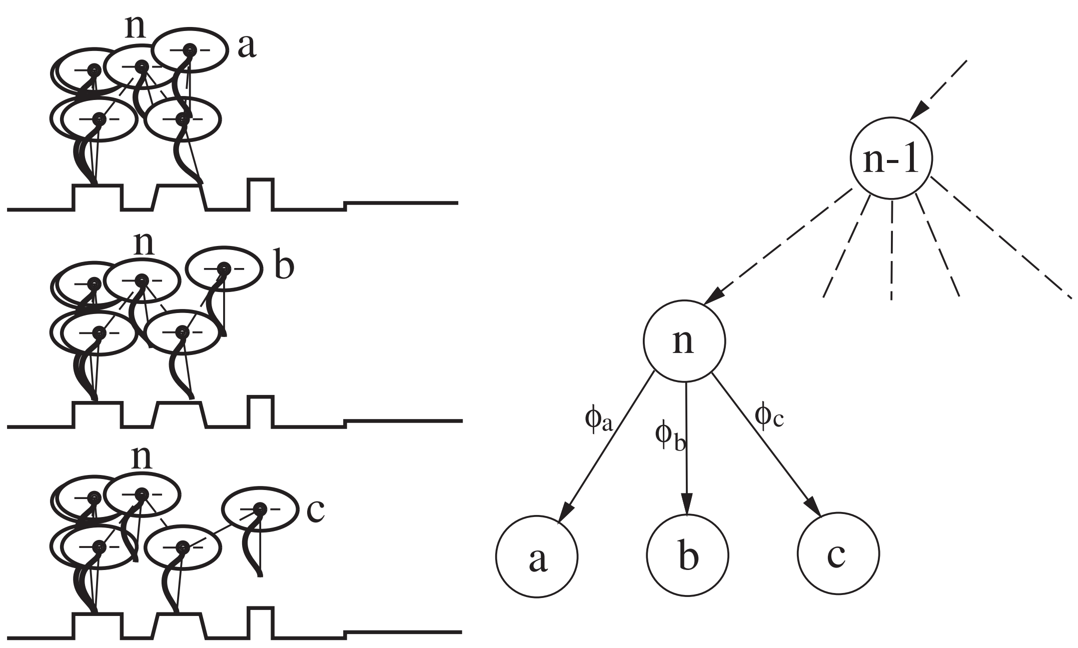


Figure 5.2: An example of the three successor states a, b, and c, heuristically chosen for plan state n. Each is associated with the illustrated leg angle.

evaluating the leg angles at a particular bounce, each resulting trajectory can be categorized according to the identity of the terrain segment on which that trajectory might land. In other words, from a particular landing state, each nearby terrain segment is associated with a range of leg angles, and the ranges do not overlap.

This selection algorithm recursively subdivides the range of allowable leg angles to identify the boundaries of the segments along the ϕ axis corresponding to the nearby terrain segments. Given a flight state and two points on the ϕ axis, the flight trajectory for each is computed and projected to the following landing. If the segment indices of the landings are the same, then all leg angles in the range land on the same segment and the procedure returns. Otherwise, it divides the segment into two parts and recurses on each. The result is a list of segments along the ϕ axis corresponding to terrain regions, possibly separated by leg angle ranges resulting in collisions. Given this list, the discretization routine selects a leg angle at the center of each segment. If a segment is wide enough, other angle are chosen by coarsely gridding across it.

The assumptions are that the trajectories that land on a particular segment are fairly similar, and that it is good to maximize the variety of paths searched by including all discrete landings. This method also aims to guarantee paths to individual

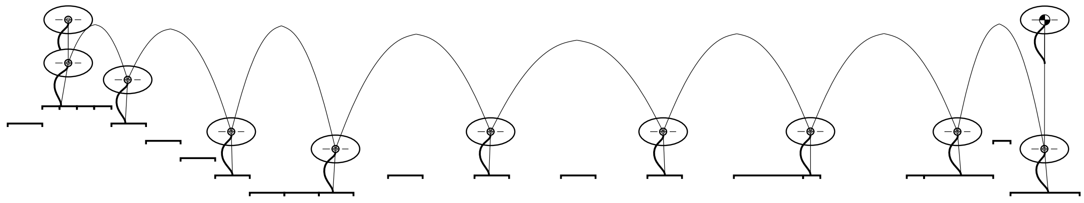


Figure 5.3: An automatically generated plan for traversing terrain composed of stepping stones. This was generated by the first planner revision that models terrain as discrete segments. In the initial state on the left the hopper is falling straight down. The goal is to reach the last state on the right with negligible lateral velocity.

narrow footholds by searching in the leg angle space.

The chief disadvantage of this procedure is the computational expense. The recursive segmentation of leg angles involves an evaluation of the bounce physics, flight path, and second landing point for each leg angle sampled. A second issue is the particular bias introduced by this sampling heuristic. It performs reasonably well with medium width stepping stones separated by gaps since it guarantees paths to the neighbors. But since the gridding routine always places the landing over the center of the stone, if the stones are too narrow, the paths might have no future since the leg choice will highly constrained. Note that the third revision planner doesn't solve this problem either, it just sidesteps the question with a unbiased uniform grid that often fails in the same case.

5.4.2 First Prototype Planner

The first major revision of the planner represents the culmination of the planner prototype presented in the thesis proposal. This planner uses a terrain model that described the ground as discrete horizontal "stepping stones," as illustrated in Figure 5.3. The advantage of the uniform gridding is that it makes for an easy search discretization: each evaluation of a landing generates trajectories to the center of nearby cells. The disadvantage is that only highly simplified terrain may be represented.

During expansion of a node, the planner computes the leg angle for the impact that will send the hopper to the center of a nearby stone. Figure 5.4 shows the set of trajectories considered from an initial state. The branching factor for each stone is the number of reachable stones. The leg angle is computed using a few iterations of Newton's method on a closed form expression that models a parabolic descent, impact, and parabolic trajectory to the nearby stone.

The expansion generates a state that places the body over the center of the stone. This means, however, that only bounces with a leg angle of zero will place the foot

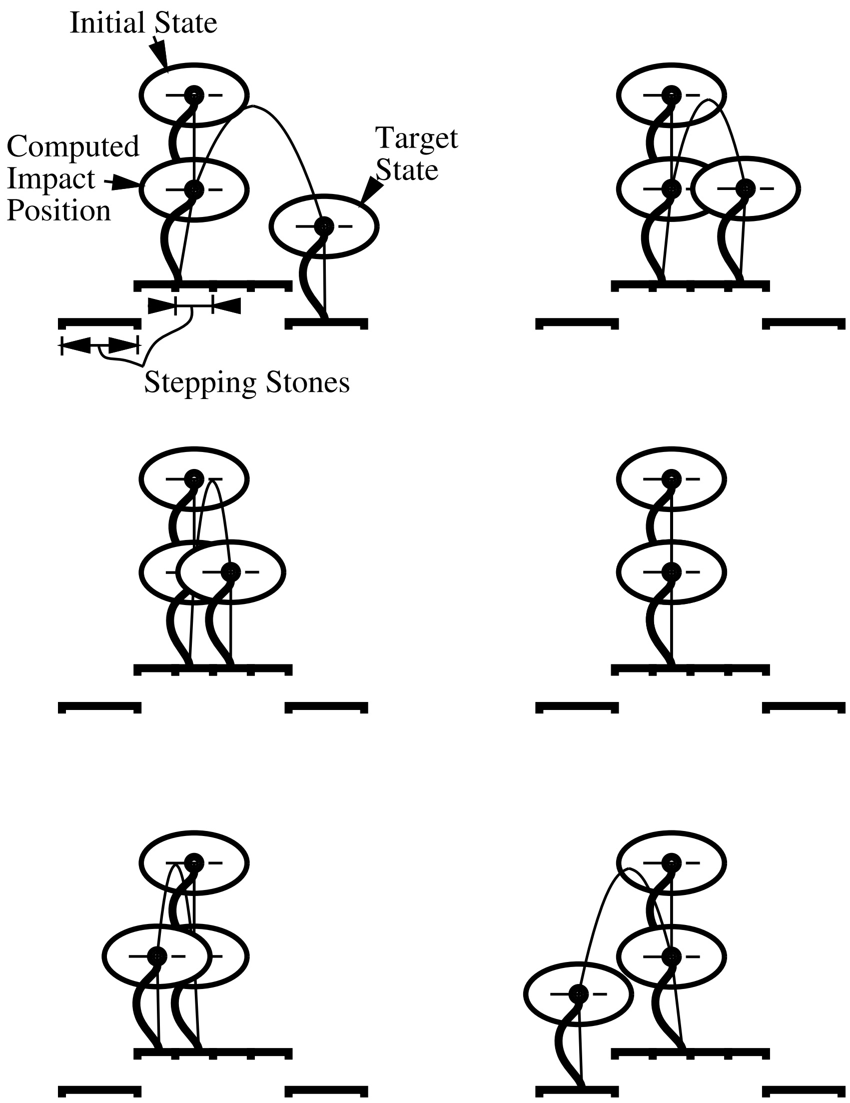


Figure 5.4: The set of trajectories considered by the first prototype planner for a given start state. This planner computes the desired leg angle that will reach each target state by numerically solving the impact function. The targets are chosen to be the center of each nearby stepping stone. The six trajectories shown are the complete set of choices that satisfy physical constraints and represent the search graph branches from this node. The terrain has been artificially divided into discrete stepping stones to reduce the branching factor.

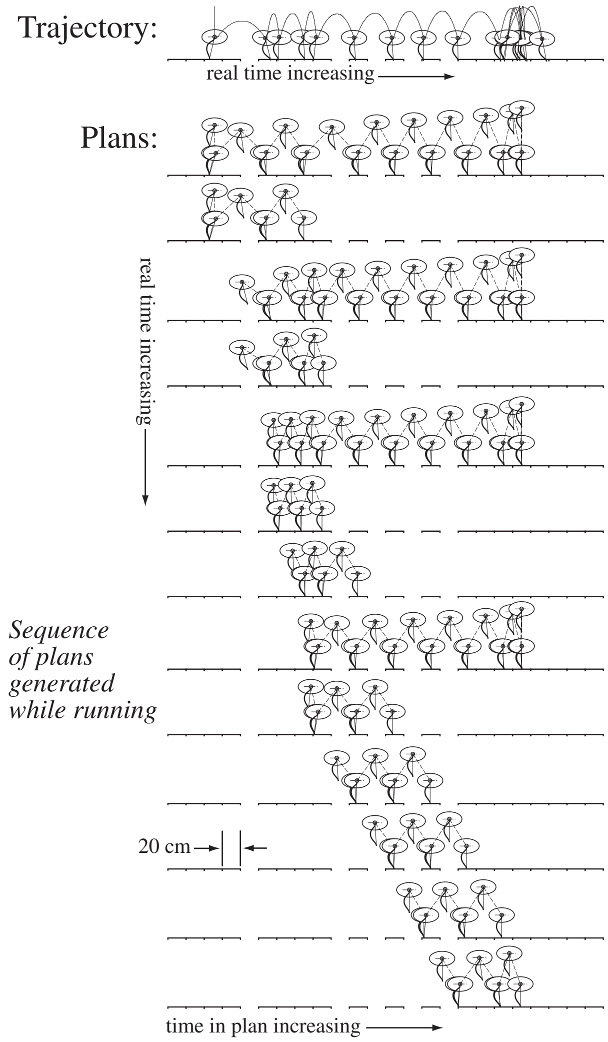


Figure 5.5: Experimental run and plans from October 7, 1997, run 7. The top plot illustrates the actual trajectory. Below are the succession of plans. Long ones are full plans, short ones are adjustments to correct errors. Real time increases moving down the figure, and planning time increases to the right.

at the center of the stone, even though that is the safest foot position. What is needed is to plan a trajectory that places the *foot* at the center of the stone, not the body. This is accomplished by performing a relaxation procedure after each node expansion that adjusts the previous leg angles to recenter the foot placements. At the beginning of expansion, the final leg angle of the partial plan is zero. During expansion, leg angles are computed to generate new nodes that reach each nearby stone. For each new partial path, the initial sequence is copied from the parent, the new leg angle inserted instead of the zero angle, and the new flight node is appended. The last foot placement now is no longer at the center of the stone; the relaxation procedure iteratively adjusts all leg angles to keep each foot placement at the center of their respective stones and the final flight state the same.

The advantage of this relaxation is that the plans are much more error-tolerant since the foot is always landing at the safest position. The disadvantage is the memory and computational expense; each new partial plan requires a new copy of all parent states in order to modify the leg angles, and a relaxation procedure is executed on each partial plan generated.

The cost heuristic (task function) for this planner generates a score for a candidate trajectory from the apex position x, the apex velocity $\dot{x}$, and the length of the path p:

$$
\begin{aligned}
\tilde{x} &= x_d - x \\
\dot{x}_d &= \begin{cases} \dot{x}_m, & \text{if } k_v\tilde{x} > \dot{x}_m \\ -\dot{x}_m, & \text{if } k_v\tilde{x} < \dot{x}_m \\ k_v\tilde{x}, & \text{otherwise} \end{cases} \\
\dot{x}_{\text{err}} &= \dot{x}_d - \dot{x} \\
\text{score} &= -|\tilde{x}| - |\dot{x}_{\text{err}}| - k_l p
\end{aligned}
$$

Two constraints are always applied to prune the search. First is friction: assuming Coulomb friction with coefficient μ, the leg angle must lie inside the friction cone within angle $\arctan \mu$ of vertical. The second is an artificial constraint that the leg angle be close to the impact velocity. The closer the angle, the less the leg angle will change during stance and the better the impact approximation. Leg angles are rejected that lie outside a cone defined by a threshold around the velocity vector.

5.5 Offline Planning

The goal of this thesis is development of on-line planning. However, a side experiment was implementation of an off-line planner that generates a policy for every state on a grid by planning backward from the goal. The planner assumes a constant total energy, so the grid covers a two-dimensional region of the $(x, \dot{x})$ plane that includes

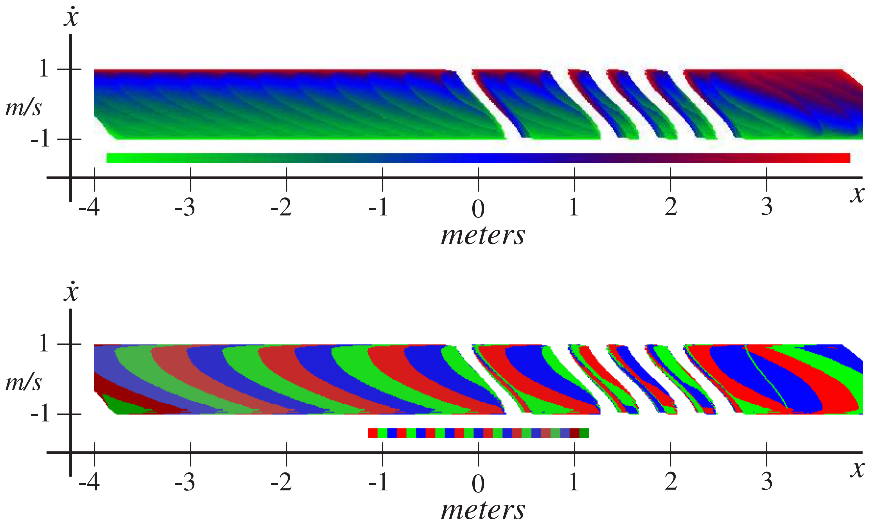


Figure 5.6: A policy for achieving a goal position from any state within a region of the $(x, \dot{x})$ plane at a specific total energy. The upper diagram illustrates the leg angle and the lower diagram the number of bounces to the goal. The horizontal axis is the x position spanning 8 meters on a 1 cm grid, and the vertical axis is the $\dot{x}$ velocity ranging from -1 m/s at the bottom to 1 m/s at the top. The goal is $(x = 3, \dot{x} = 0)$; the white diagonal bands are the pre-images of five obstacle holes. The narrow horizontal bars indicate the (unlabelled) range of values in the associated image.

the goal. The planner performs an exhaustive reverse breadth-first search using the states on the grid, an inverse model of the bounce physics, and the heuristics from the forward planner.

The search begins by placing the goal state into a priority queue. At each iteration, the cheapest path (lowest cost to the goal) is removed. The path is projected from that state to its takeoff point, and the leg angles at that contact are evaluated to find all trajectories on the grid that could possibly reach the state. The path cost for each predecessor state is computed using the heuristics and compared against any previously found from that state. It the new path is cheaper, it replaces the state in the grid and is added to the priority queue for possible expansion. This iterates until the queue is empty.

The experiment used an 800 by 100 grid at 1 cm by 2 cm/sec resolution. The procedure ran in about an hour on a 133 MHz Pentium processor, without any particular attention to code optimization. A sample result is shown in Figure 5.6.

The result was tested on the robot by using the policy as a look-up table. Energy was regulated to the constant level required by the policy, and on each flight the $(x, \dot{x})$

position was used to index the correct leg angle from the policy table. The hopper then successfully crossed the five holes from a variety of start positions. The chief limitation was imprecision in the energy regulation; the energy wandered enough around the nominal level to cause a few failures in marginal cases near the edges of the holes. Ideally, a 3D table would be generated to represent the policy for every state.

5.6 Discussion

To some degree, the opportunity to apply planning methods to locomotion is attributable to the advance in computer speed since the Raibert work. It is a trend that accounts for quite a few of the advances in the field of robotics. But the simplicity of the Bow Leg mechanism was a significant advantage. It has predictable physics that allowed the development of a physical model computationally cheap enough to use in a planner.

A important creative constraint on the planner development was a decision to require real time operation; this means no precomputation to analyze terrain, no off-line searches, and that paths must be found while moving. This assumption incorporates at least some of the real world constraint that terrain may be sensed at sufficient resolution only as it is traversed. This also limits the scope of algorithms and encourages the development of heuristics.

A general principle for planner design is to take full advantage of the mechanism; that is, exploit the discrete operation and the simplicity of the physics to allow all possible motions. Another is to aim toward natural terrain. Laboratory solutions that depend on a planar machine or on overly structured world might not generalize, so the work must embrace uncertainty, the eventual perception of the environment, and operation in real time. The planner should constantly be tested on the actual robot as a reality check. And last is to strive toward a principled solution that will outlive this hardware.

These principles suggested a state space graph search to me. It does need to discretize the controls and thereby loses full generality but is simple and unbiased and can work in real time. It may incorporate arbitrary constraints to allow for varied task expression. The incremental forward search satisfies real time by finding approximate solutions quickly. The simple terrain description can be modified on-line, allowing the eventual incorporation of on-line terrain perception.

Future Work

Eventually the Bow Leg Hopper will include a movable hip used to control body pitch by configuring the coupling between the ground impulse and the body torque impulse. The pitch could be simply regulated at zero during execution, or an explicit pitch path could be generated. It seems likely that the decoupled planning approach will still work fine. Since the energy in body rotation is much smaller that energy in translation, a pitch plan could be added to the body plan by a modest adjustment to the energy plan to account for energy transferred to and from rotation. The hopper will also include a Bow Leg with partial energy release that may retain stored energy after impact. A decoupled approach could generate an energy plan based on terrain contour. Such a plan would take the form of a function specifying total energy as a function of terrain position, and would include retention of energy in the leg on descending slopes. Using this function the body path could still be planned with a one-dimensional search.

The existing planner could be made more efficient by caching results. Since the planner is intended to continuously cross new terrain, such a cache would need to store results as templates that could be applied to similar terrain elsewhere. This could also take the form of a library of canonical terrain configurations and precomputed plans. It is conceivable that plans for one segment could be parametrically fit to another terrain segment.

A higher level problem is strategic planning. An example is planning with limited terrain data. If the terrain map always includes the goal, then local planning is sufficient. But a horizon on terrain perception requires strategies like "hop to the edge and look across" or "make sure you can back out." This may involve planning contingencies or accepting the possibility of catastrophic failure.

The planners explored in this thesis produce plans with a constant total energy. The chief advantage is that the planner need only plan the leg angle; the search space is one-dimensional. The chief disadvantage is that the hopper has an absolute limit on the altitude it can achieve, even if the terrain rises at a modest slope and the hopper could easily add energy to maintain constant fall distance. Also, terrain is measured locally, not globally, so absolute energy is unmeasurable. So it is desirable to plan in terms of total energy, but not necessarily *constant* total energy.

Here are some suggestions for better ways to accomplish energy planning:

- Use an explicit analysis of the minimum energy required to cross a terrain segment to choose a lower bound on the total energy, and heuristically choose a desired value somewhat greater. This would tend toward "minimal energy plans," but plan quality might suffer as the lowest energy states are also high risk.

- Vary the desired total energy as a function of position. It could be generated by a piecewise convex hull of the terrain or a local averaging. The smoothing factor is related to the limits of the thrust mechanism; ideally, a large step would affect the desired energy far enough away the hopper would start building up energy in anticipation and not have to slow down.

- The desired total energy could also be slowly varied as a dynamic value of an individual plan. In the simplest case, whenever the planner fails to find a solution it could increase the overall energy goal and try again from scratch. Alternatively, each search branch could be evaluated at several energy levels and the plan quality used to choose the appropriate energy goal. Similarly, the planner could plan at a constant level but propagate the energy goal back a few steps and replan when the trial plan encounters a big elevation change.

- The energy added at each bounce could be chosen from the entire control range. This raises the planning space to two DOF per bounce. This could fully exploit the mechanism and might solve some degenerate cases, but the planner would likely waste a lot of time evaluating plans with similar average energy but with minor variations in the energy level bounce by bounce.

Energy planning could also be incorporated into the cost functions; the energy consumed by a candidate energy plan is a reasonable basis for comparison. Unfortunately, very low energy plans are usually high risk plans. As discussed previously, incorporating the expected cost of failure in energy terms is difficult both because the prior probabilities of a plan failing is difficult to estimate and because the cost for failing can involve infinities.

Chapter 6

Experiments

The majority of "experiments" performed during development of the Bow Leg Hopper prototype took the form of test runs while debugging the controller software. This chapter presents a sampling of these performances along with a few more focussed experiments. Robotics is an engineering discipline; these experiments were performed in the spirit of engineering development rather than scientific discovery. As such, the work did not include any comprehensive side-by-side comparison of different methodologies.

Experiments with the machine include hopping in place, running at low velocities across level ground, and crossing obstacle fields composed of "stepping stones" separated by "holes" in which the hopper must not land. In many experiments the stepping stones were all the same height, and in others varied modestly or formed gentle stairs. The stepping stones were generally arrangements of patio bricks. Holes were simply designated regions on the floor with which contact must be avoided.

Overall, the prototype exhibited excellent mechanical reliability: on nearly any day over the past two years it could be turned on and demonstrated using the linear controller to hop in place or at moderate velocity back and forth. The planner, however, was quite finicky and depended upon accurate calibration of the effective gravity, motor calibration, and the tuning of the heuristics. It was also the most complex software component and required a great deal of debugging effort to keep running in conjunction with the real time system.

The first section presents a number of hopping tests. Following these are more specific experiments to document mechanical performance.

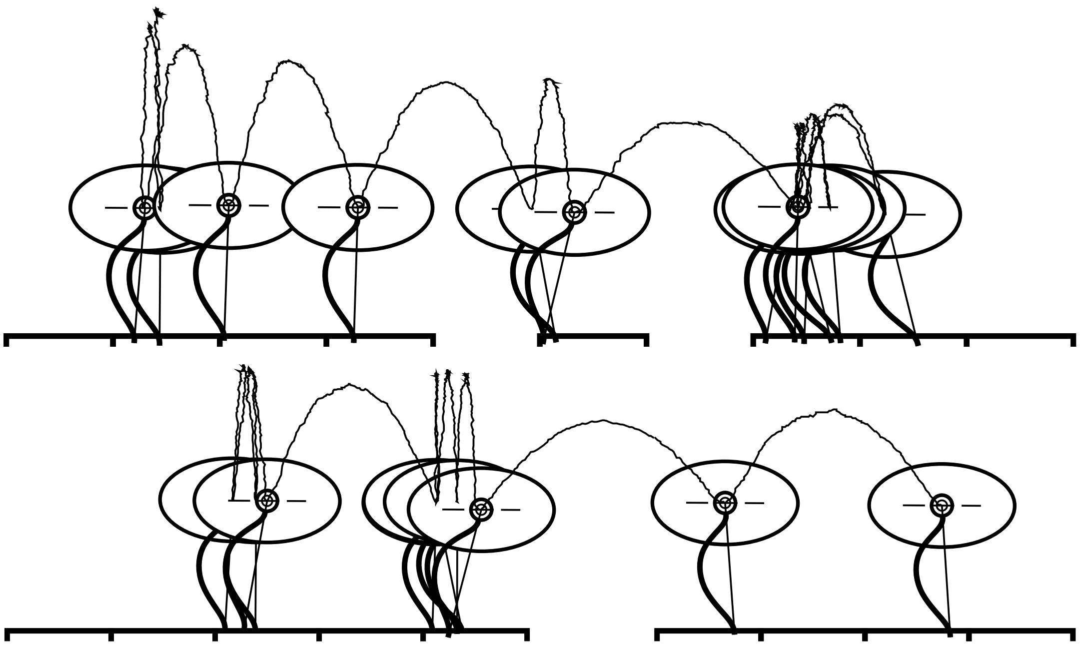


Figure 6.1: Early experimental data showing the center of mass trajectory and touchdown positions. The top plot is run 97-03-26.1 in which the hopper runs across stepping stones with two gaps to a goal position. The bottom plot is run 97-05-03.3 in which the hopper approaches a hole and then crosses it. In both cases the pitch axis is clamped so the hopper has only 2 DOF. The planner in use is the first revision with discretized terrain.

6.1 Performances

For the initial tests of the hopper and planner the pitch axis was clamped at the boom so the hopper had only 2 DOF. Two example runs are shown in Figure 6.1.

Figure 6.2 illustrates the first recorded run with the pitch axis unclamped. The top plot traces the body path; the remainder are plots versus time. The hopper was started by lifting it by hand and dropping it at about the 52 second mark. The body attitude (third plot) was stabilized entirely passively; the slight positive bias reflects a slight imbalance of the ballast weight. Note that the leg angle (bottom plot) is computed at the beginning of descent, then adjusted once before impact. This is a graphic depiction of the low control bandwidth of the Bow Leg design.

The controller state index (fourth plot down) illustrates the operation of the state machine used to track the transitions between ascent, descent, pre-impact, and stance. This is used to signal when trajectory data is available, when to commit to a leg angles, and when to begin tensioning the leg.

Figure 6.3 is another early run, in this case using the first revision planner to jump over a single hole. Note the “hunting” behavior at the end as the planner tries to stay in place at the goal by constantly replanning paths from nearby locations; the terrain discretization causes plan discretization that makes the hopper overshoot.

Figure 6.4 shows the hopper traversing four holes using the second revision planner. The holes are located in the intervals [1.0, 1.2], [1.4, 1.6], [1.8, 2.0], and [2.2, 2.4]. The hopper pauses for three bounces on the stone [1.6, 1.8] as the planner fails to find a plan for some unknown reason. Note the pitch was prone to slow oscillation in this run; the passive stability depends on the careful balance of the ballast weight and may fail dramatically if the pendulum frequency gets close to the hopping frequency.

Figure 6.5 shows the hopper traversing two glass blocks and a trapezoidal brick. The glass blocks were 8 cm tall, occupying the regions [91.500, 91.673] and [91.800, 91.973], and defined with a friction coefficient of 0.2. The top of the trapezoidal brick was 9.2 cm tall and occupied [92.150, 92.228], defined with the usual friction coefficient of 0.4. These friction coefficients were not precisely determined; for the glass block the value was simply reduced until the planner stopped using leg angles that caused the block to slip.

Figure 6.6 shows the hopper traversing a trapezoidal brick and a piece of wood 2x4 balanced on edge. The brick was 9.2 cm tall, positioned at [71.606, 71.684], and the piece of wood was 8.8 cm tall, positioned at [72.000, 72.038]. This was a dramatic demonstration, but only worked about a third of the time due to imprecise control of the hopper trajectory. However, even in cases where the foot landed on the edge of the wood, the wood would be knocked away but the foot would simply hit the floor instead and the hopper would keep running.

Figure 6.7 shows the hopper climbing three gentle steps. The first step is 5.8 cm tall at [61.410, 61.600], the second 11.5 cm tall at [61.710, 61.900], and the third 17.5 cm tall at [62.010, 62.200].

6.2 Mechanical Performance

Two different leg positioners were used in the hopper prototype. The first was an actuated tube surrounding the Bow String, and the second a separate harness of two strings in a triangular arrangement. Figure 6.8 shows the response of the leg to a series of control signal steps using each mechanism. Normally the hopper has no direct sensor for foot position but at the time a temporary sensor was in place using an independent pair of strings attached to a potentiometer. These plots illustrate the combined effects of the hobby servo step response, slop in the control strings, and any slop in the sensor strings. The reason for the redesign is that the tube approach has zero lateral stiffness at the goal point and was prone to rattling around the goal during flight. The triangular harness is much stiffer at the the goal point and this experiment suggested the positioning response was no worse.

The controller estimates body velocity using an estimator as described in Section 4.3.2. Figure 6.9 illustrates the performance of the estimator; the top two plots show a case with a well-tuned feedforward vertical acceleration and the bottom two a poorly-tuned case. In each case the actual x velocity is constant during flight; the step response of the `qd.x` lateral velocity curve primarily reflects the performance of the filter.

For reasons of hardware availability the body pitch potentiometer was never replaced with an encoder and hence is the noisiest sensor axis. As a result the estimator gains were much slower than the other axis; Figure 6.10 illustrates the operation of the theta estimator as it averages out the electronic noise.

6.3 Planning Performance Statistics

Figures 6.11 6.12, and 6.13 present some statistics on the performance of the hopper while using the planning control. Since the planner views the hopper as a discrete-time system, each datum for these plots represents once bounce or one flight phase. Figure 6.11 is a scatterplot of the difference between the expected and observed apex position. This includes residual modelling error as well as uncertainties in the mechanism. Figure 6.12 shows the performance of the constant energy regulation. Figure 6.13 is a histogram of the number of planned bounces successfully performed before requiring a replan, either due to excessive error or simply finishing the planned sequence. Bear in mind, however, that each unit represents one bounce sequence, so the bins to the right represent increasingly more bounces per unit. Thus the leftmost bin has the highest value but only represents 17% of the total number of bounces.

The following data sets were used to generate these statistics: `98-02-21.5`, `98-02-21.8`, `98-04-01.4`, `98-02-21.7`, `98-04-16.3`, `98-04-16.4`, and `98-04-16.5`.

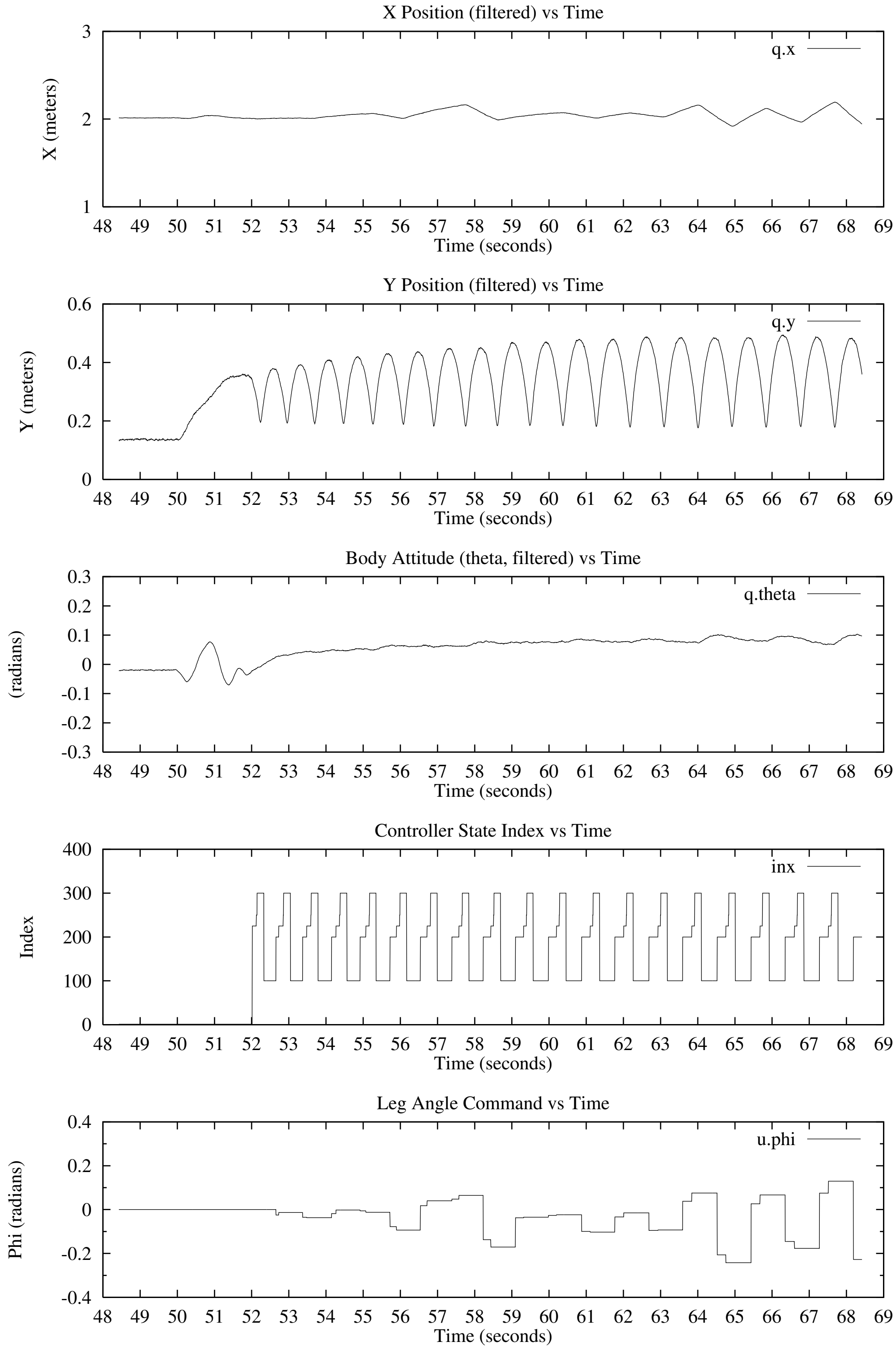


Figure 6.2: Data Set 97-05-07.1, hopping in place with a linear controller.

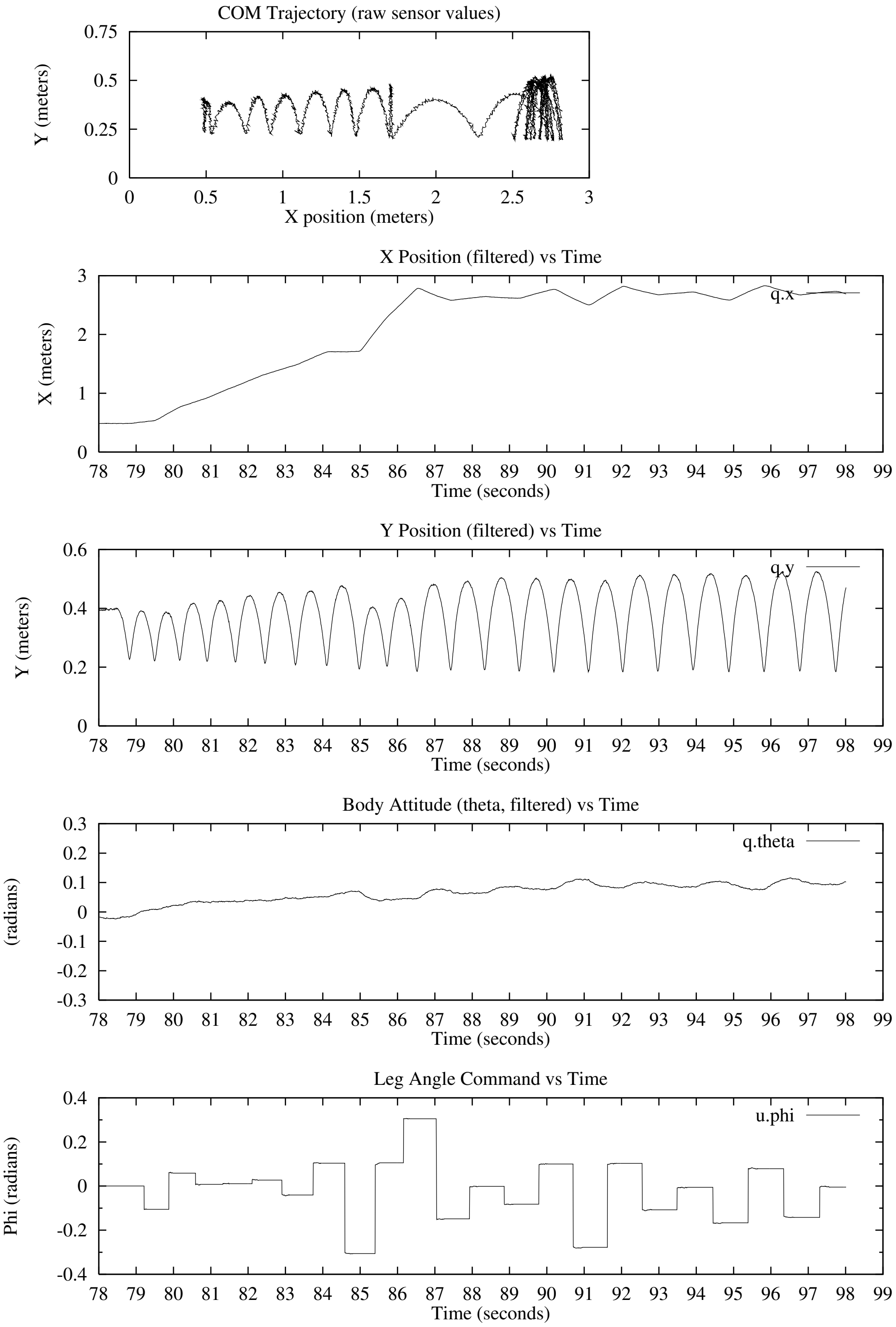


Figure 6.3: Data Set 97-05-07.3, jumping over hole using planner.

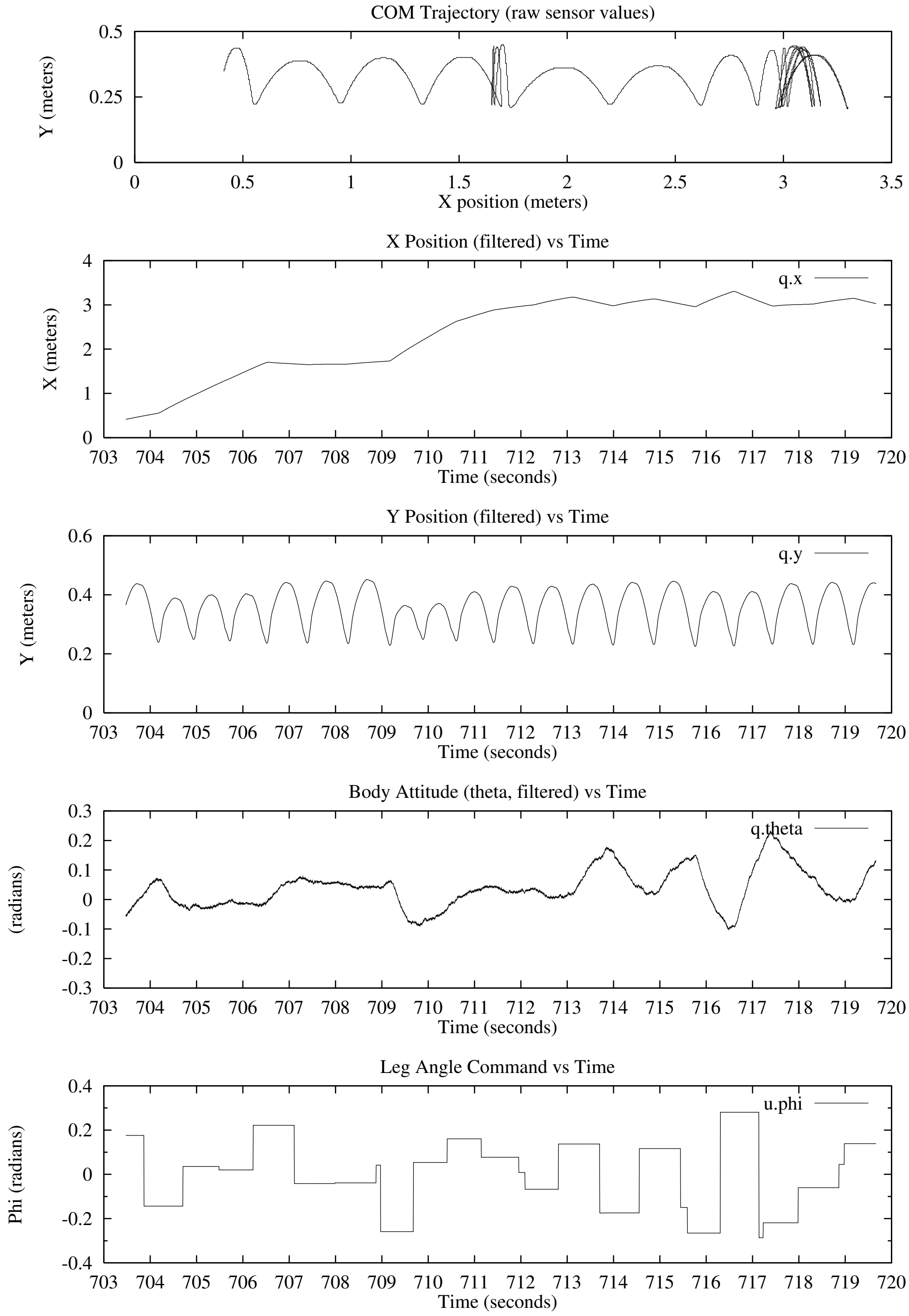


Figure 6.4: Data Set 98-02-21.7, jumping over series of four holes using planner. Between 707 and 709 it pauses on the stone between the second and third holes.

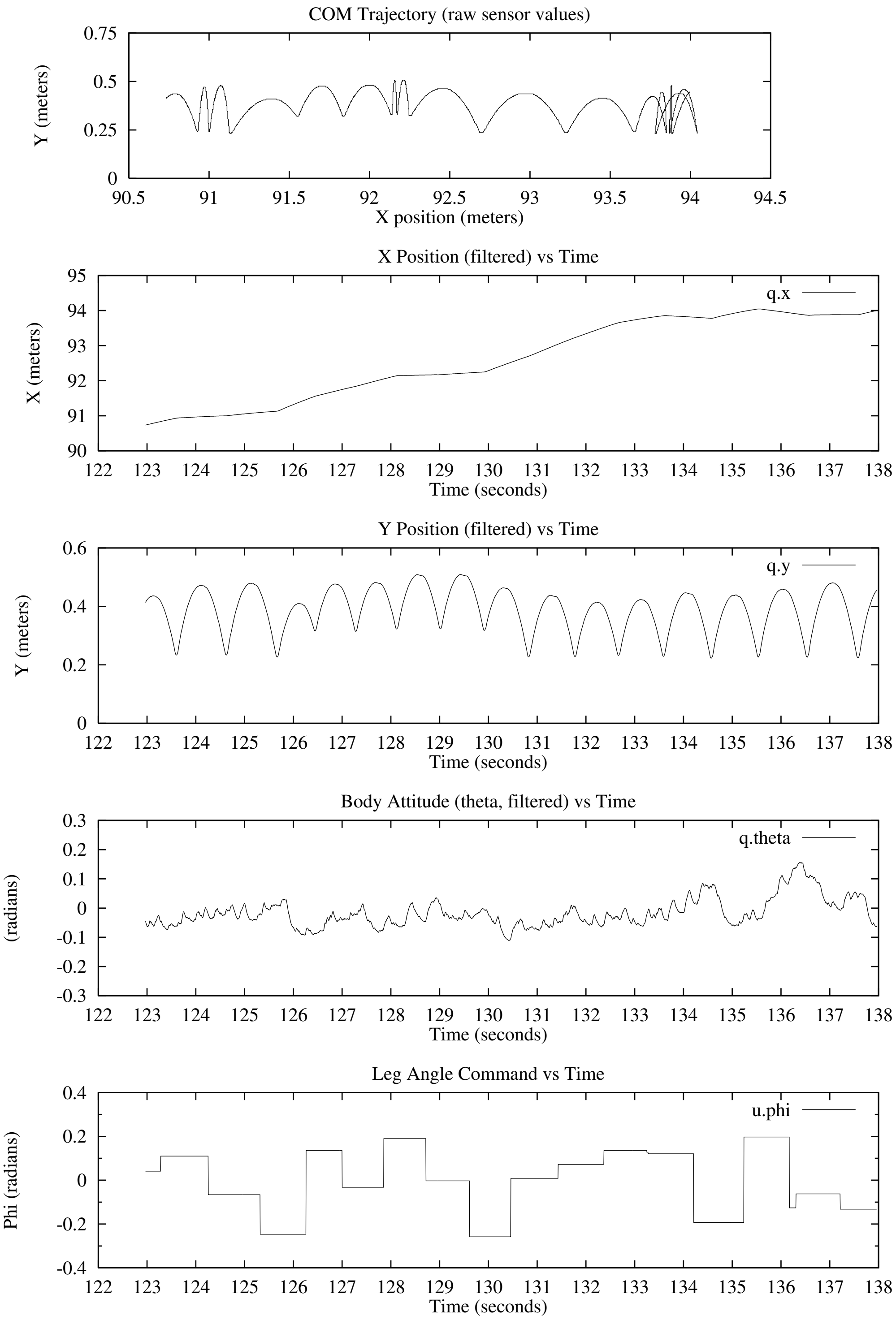


Figure 6.5: Data Set 98-04-16.3, hopping on two glass blocks and a trapezoidal brick, using planning.

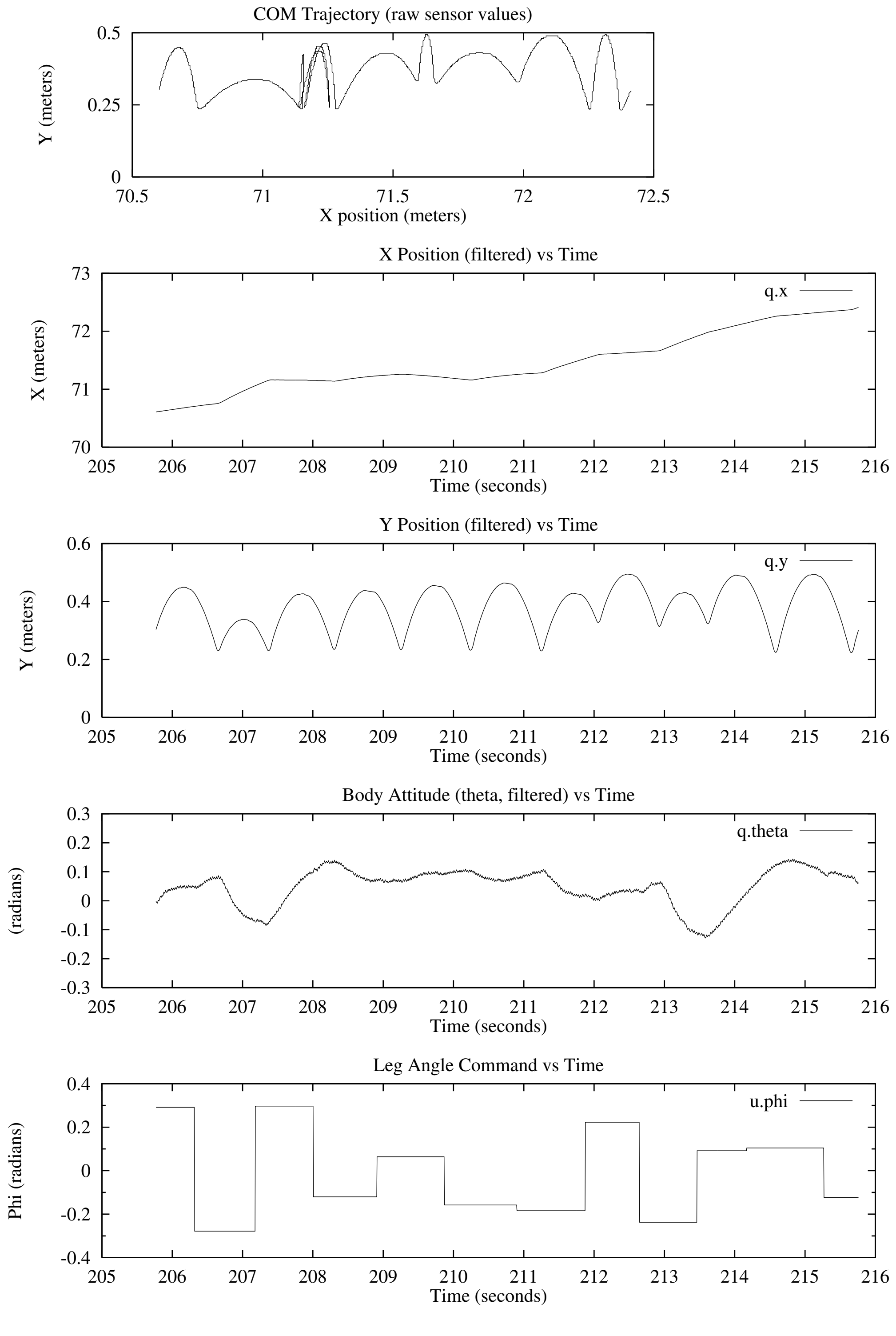


Figure 6.6: Data Set 98-04-16.4, hopping on a trapezoidal brick and a 2x4 on edge, using planning.

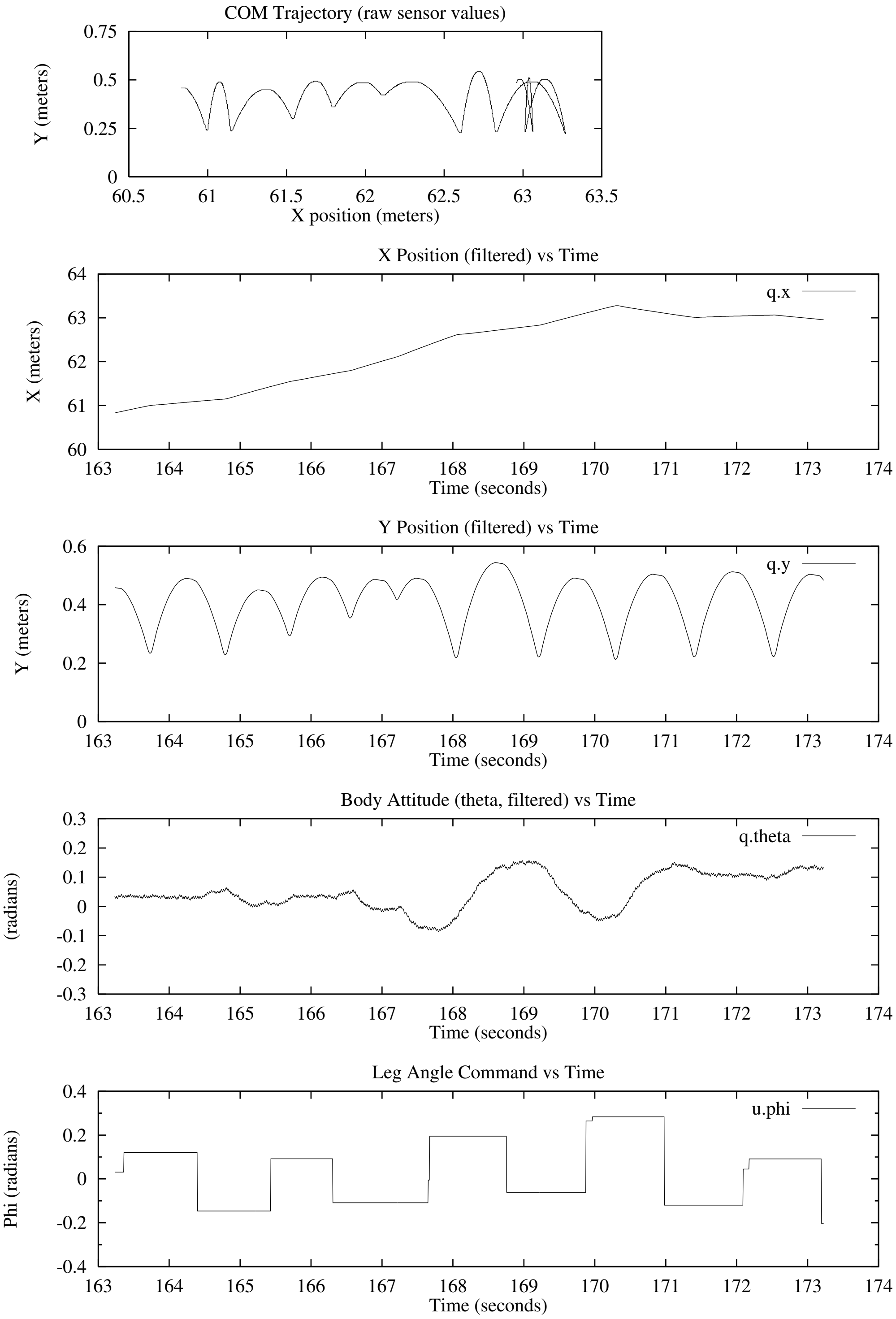


Figure 6.7: Data Set 98-04-16.5, climbing three gray brick steps, using planning.

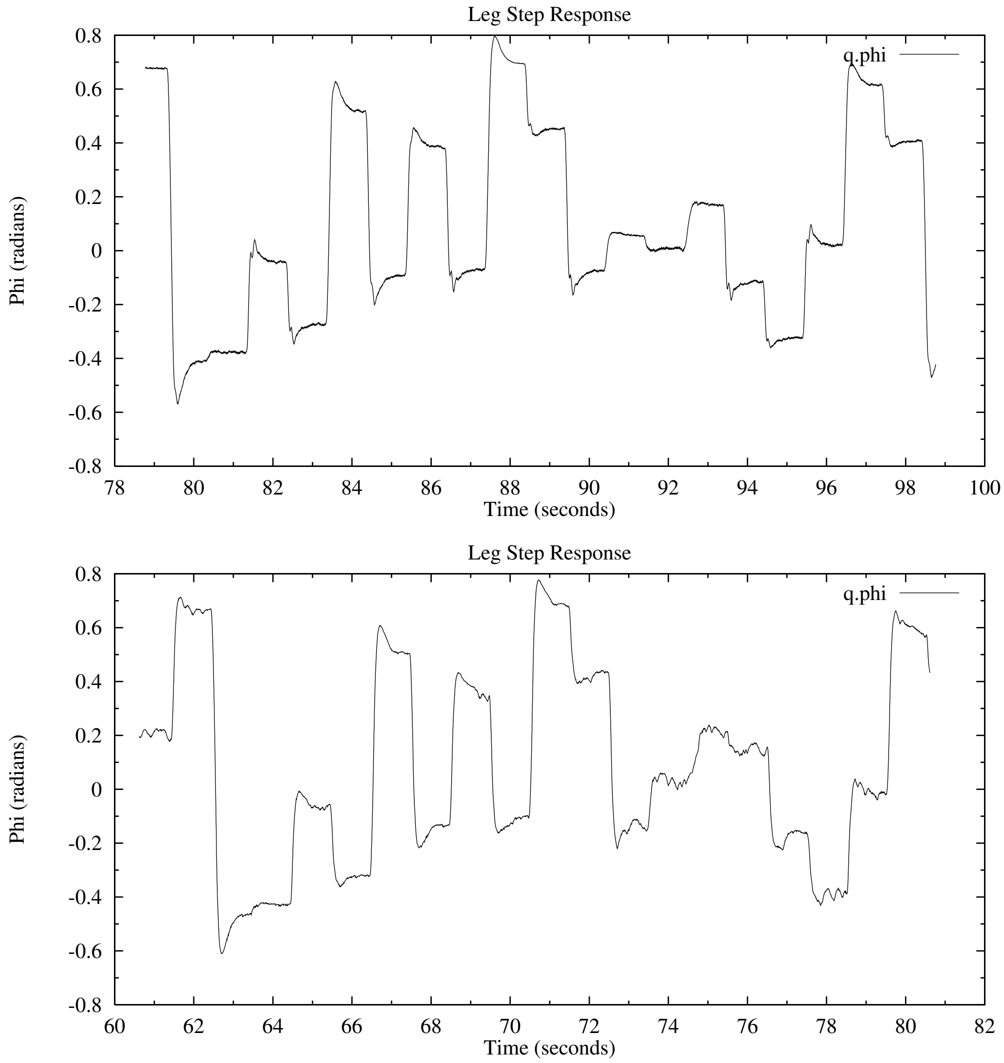


Figure 6.8: Data Sets 98-01-26.1 and 98-01-27.2, illustrating the leg step response using a temporary leg position sensor. The top plot uses the original positioning lever with a tube around the Bow String, the bottom plot the triangular harness with two strings. The same sequence of steps was used as the control signal for each test.

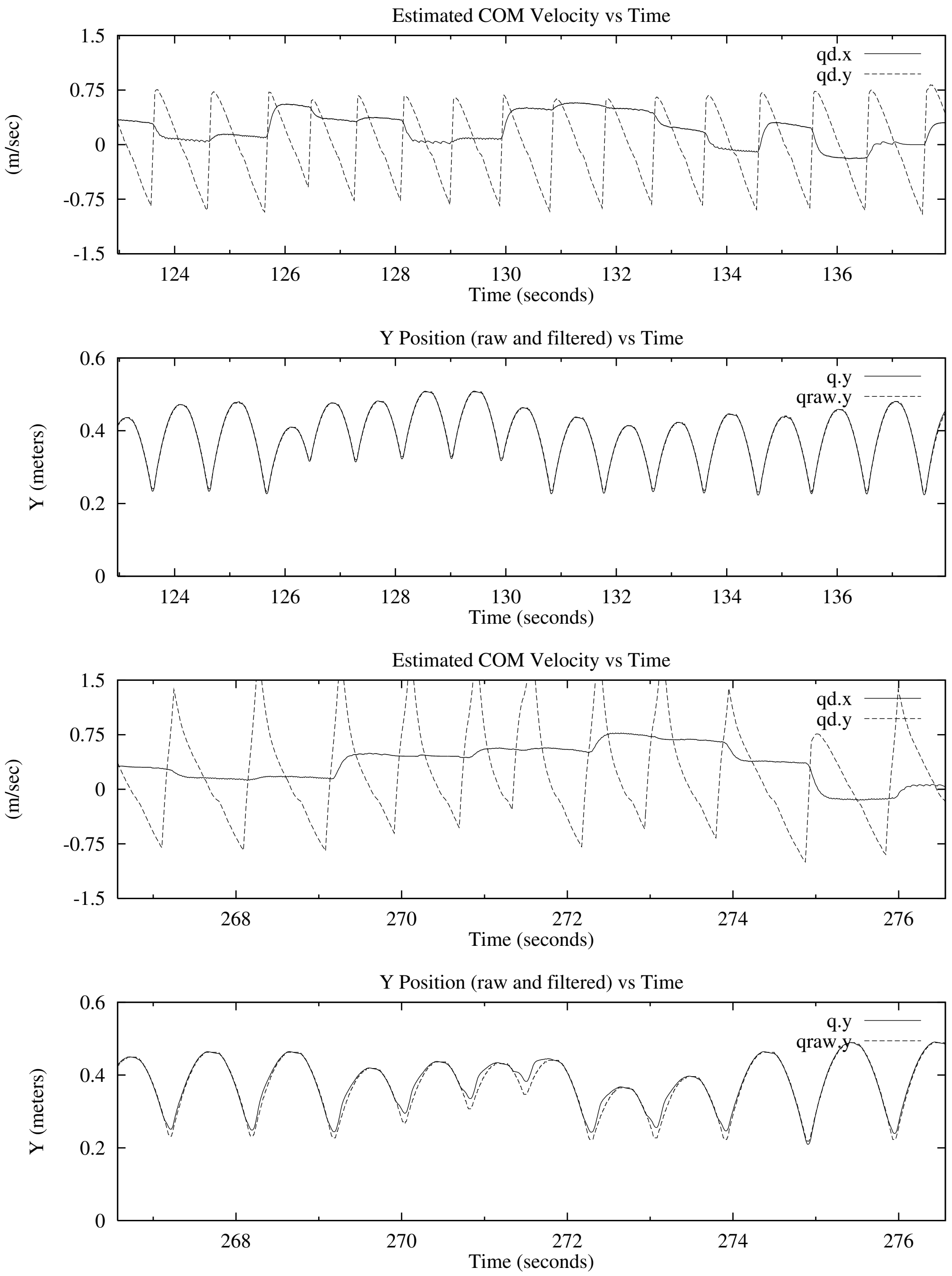


Figure 6.9: Data Sets 98-04-16.3 (top two plots) and 98-04-01.4 (bottom two plots), illustrating the velocity estimator in a well tuned and a poorly tuned state. The steep vertical in the Y velocity results from a feedforward estimate of stance acceleration; the downslope should be a straight line during flight.

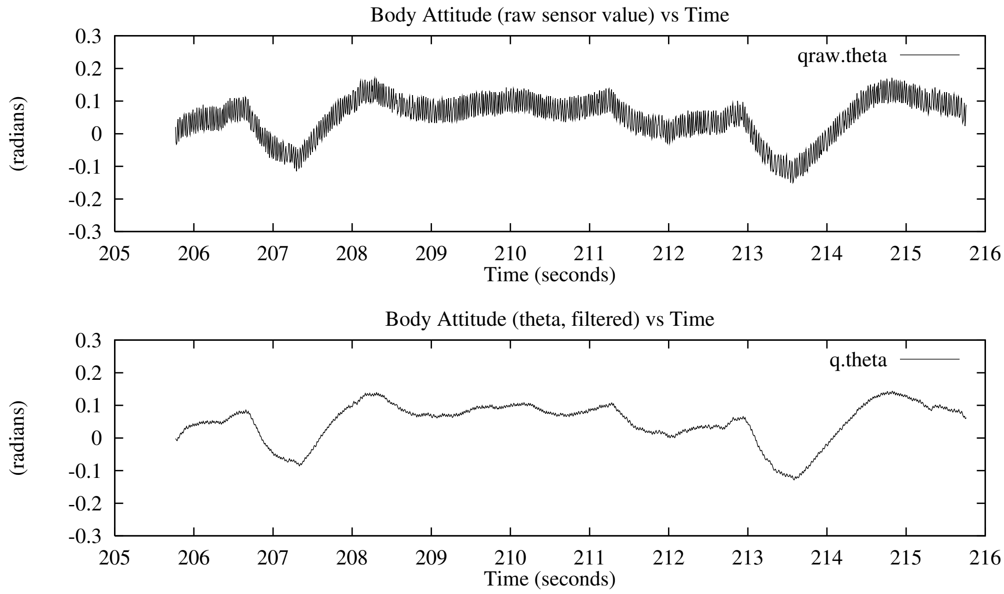


Figure 6.10: Data Set 98-04-16.4, illustrating operation of the theta filter. This axis is passively stabilized.

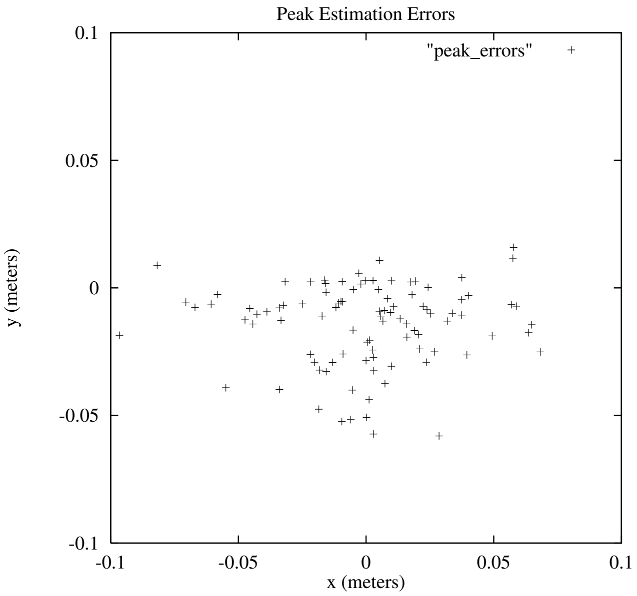


Figure 6.11: Apex position error over 101 bounces taken from seven experimental runs. The error is the difference between the expected peak position and the actual peak position, and represents the error in plan execution that must be rejected.

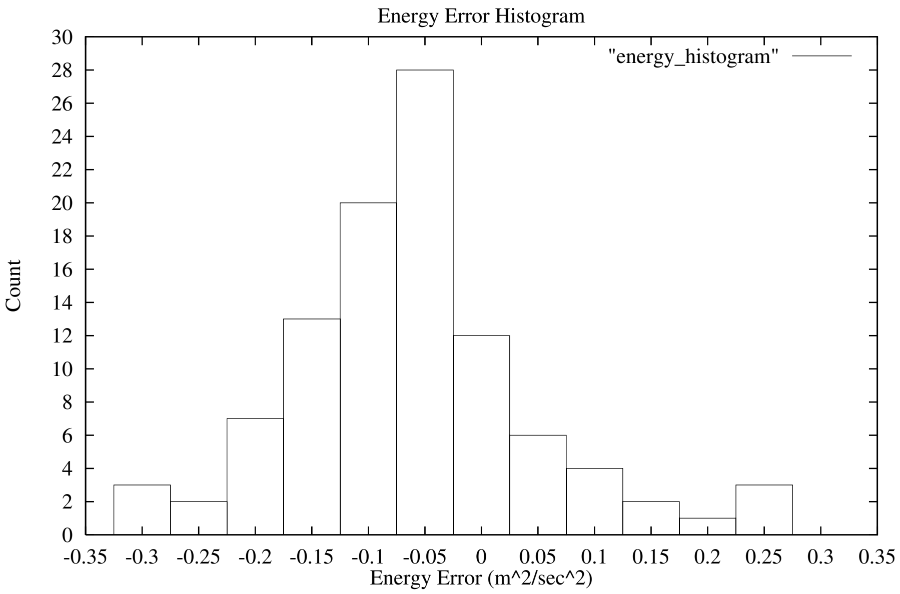


Figure 6.12: Histogram of energy regulation errors over 101 bounces taken from seven experimental runs. The units are m^2/sec^2 to normalize for machine mass; the typical energy goal is about 2.0.

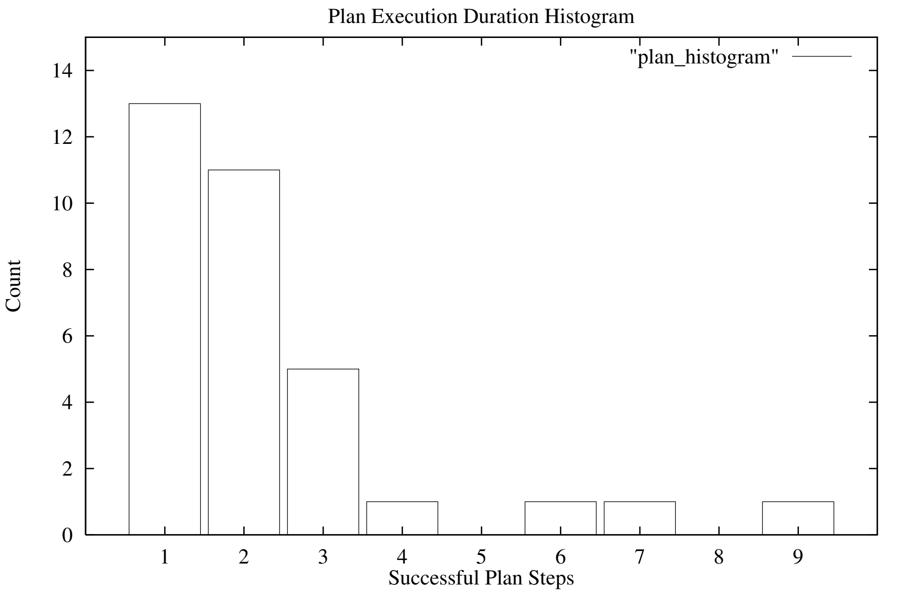


Figure 6.13: Histogram of the duration of plan execution using 33 subsequences totaling 76 bounces from seven experimental runs. Each subsequence ends when a replan was required; the histogram plots the distribution of subsequence lengths.

6.4 Accelerometer Measurements

This section documents an experiment to measure the accelerations experienced by the hopper body during stance. The intent was to validate some mechanical design assumptions central to the simplicity of the Bow Leg idea. These assumptions are that the freely pivoting hip locates the leg forces along the leg axis and the low-mass foot and leg minimize energy losses during the initial ground contact. Furthermore, it was expected that the leg would have relatively constant force, meaning stance time would be proportional to the fall distance. One unknown was whether body vibration was a cause of parasitic energy dissipation; this experiment did not measure the loss, but offered some evidence as to the magnitude of the vibration.

The following data were taken January 28, 1999. A pair of accelerometers were mounted on the hopper body and the output measured with a digital oscilloscope. The sensors were calibrated to approximately 0 Volt equal to 0 acceleration and 500 mV equal to one earth gravity. The measurements were made using a pair of IC Sensors 3021-020-P accelerometers amplified by a Measurements Group 2120 Strain Gage Conditioner instrumentation amplifier and recorded with a model TDS 410A Tektronix scope. For all but one case, the sensors were mounted on the constraint boom near the hopper.

The complete set of recordings included the following:

1. Series of recordings from dropping the hopper by hand, no battery installed, with the gravity compensation spring in place:

2. Short series of recordings from dropping the hopper by hand, no battery installed, with the gravity compensation spring slack.

3. Series of recordings from dropping the hopper by hand, battery not installed, the gravity compensation spring in place, recording only Y but at a finer time scale.

4. Series of recordings made at different time scales while the hopper was hopping in place using the linear control algorithm.

5. Series of recordings from dropping the hopper by hand, battery in place, with the gravity compensation spring in place, with the leg angle set to a value and the controller in calibration mode. The free-fall measurement (compensation spring force) was 394 millivolts.

6. Two recordings to illustrate impulse response of both the body and boom structures.

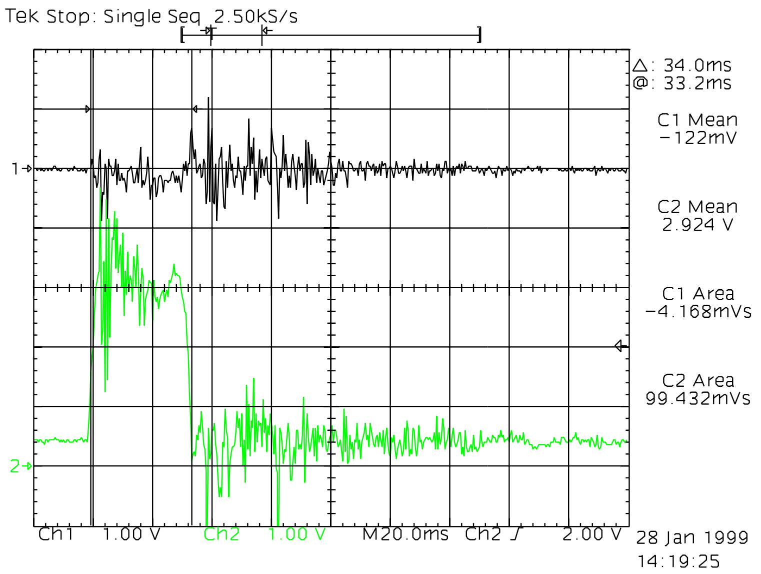


Figure 6.14: X and Y accelerometer outputs, falling from 40 cm in reduced gravity.

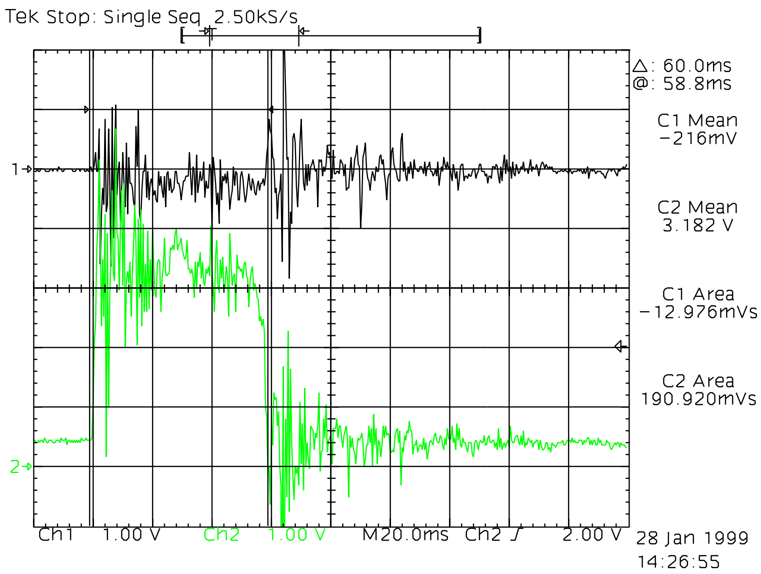


Figure 6.15: X and Y accelerometer outputs, falling from 80 cm in reduced gravity.

The first pair of recordings shown in Figures 6.14 and 6.15 were taken during impact after dropping the hopper by hand. The battery was not installed in order to measure the response independent of any battery mount vibration. The effective gravity was reduced by the compensation spring. The plot was generated by the oscilloscope from the screen display; the two cursors were manually positioned over the apparent beginning and end of the impact and the mean and area calculations performed by the device. Since the accelerometers were calibrated to approximately 500 mV equal to one earth gravity, the scaling is the 1 V is equal to 19.6 $\mathrm{m/sec}^2$ and 1 V-second is equal to a velocity change of 19.6 $\mathrm{m/sec}$.

In the two figures, the upper plot is the lateral acceleration and the lower plot the vertical. The leg was positioned straight down so the average lateral acceleration labelled "C1 Mean" is near zero as expected. Ringing is apparent in both dimensions during stance and extends 40–60 msec after takeoff. Since the leg is about 25 cm long, Figure 6.14 represents approximately a 15 cm fall; as shown by the average Y accelerometer value labelled "C2 Mean," the leg exerts about a 6 G force on the body. The value "C2 Area" reflects a net velocity change of about 2 $\mathrm{m/sec}$. Figure 6.15 represents a 55 cm fall; the average acceleration increases less than 10%, consistent with a nearly-constant leg force. Since the impact velocity is $-\sqrt{2g\Delta y}$, the expected velocity increase is $\sqrt{55/15}$ which is equal to 1.91, very consistent with the measured ratio of 1.92.

Figure 6.16 represents a passive fall with the gravity compensation spring held slack so the hopper falls with 1 G acceleration. The vertical acceleration ("C2 Mean") is lower than in Figure 6.14 since the leg force is more balanced by gravitational force. The total velocity change ("C2 Area") is about the same as the longer fall of Figure 6.15; considering the difference in altitudes this implies the compensated acceleration was 26% of G; unfortunately, that value was not otherwise measured. The initial lateral vibration is worse but the body ringing time is about the same, consistent with observing a free vibration of the structure.

Figure 6.17 was taken at finer time scale (5 msec/div) from a 50 cm altitude (25 cm fall). The stance time is about 43 msec. The leg forces apparently load and unload in just a few milliseconds.

The effect of the leg angle may be seen in Figure 6.18. The hopper was dropped by hand from 50 cm altitude with the controller holding the leg at the 0.30 radian position. For the motor to function the battery was installed so the body had slightly higher mass. The expected ratio of velocity impulses is $\tan(0.3) = 0.31$; the measured ratio is $0.044/0.101 = 0.44$ which is higher than expected.

Figure 6.19 illustrates several impacts while hopping in place using a linear controller. The scale is 200 msec/division and the measured cycle time is 756 msec. Figures 6.20 and 6.21 present a close-up view of a takeoff and landing at 2 msec/division.

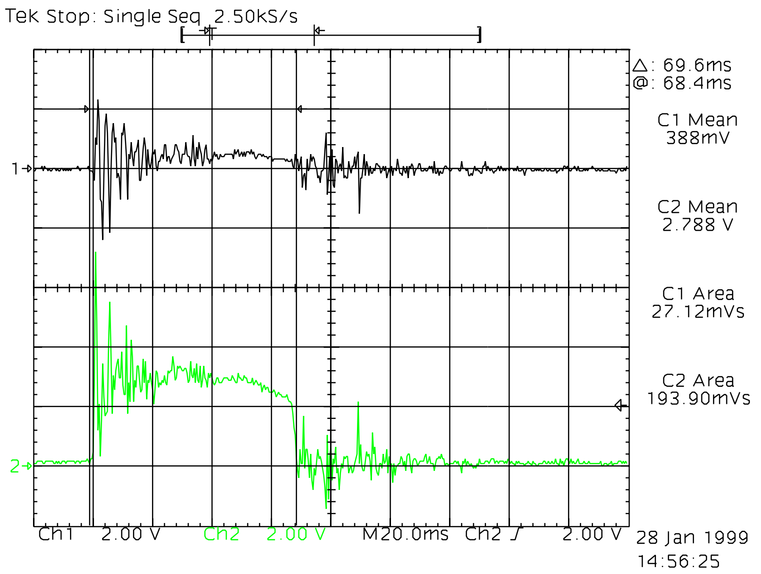


Figure 6.16: X and Y accelerometer outputs, falling from 40 cm in full gravity.

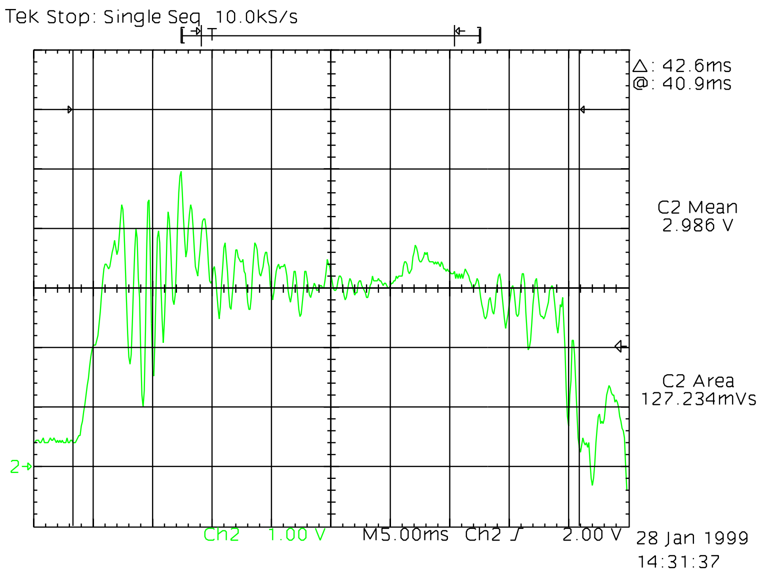


Figure 6.17: Y accelerometer output, falling from 50 cm in reduced gravity.

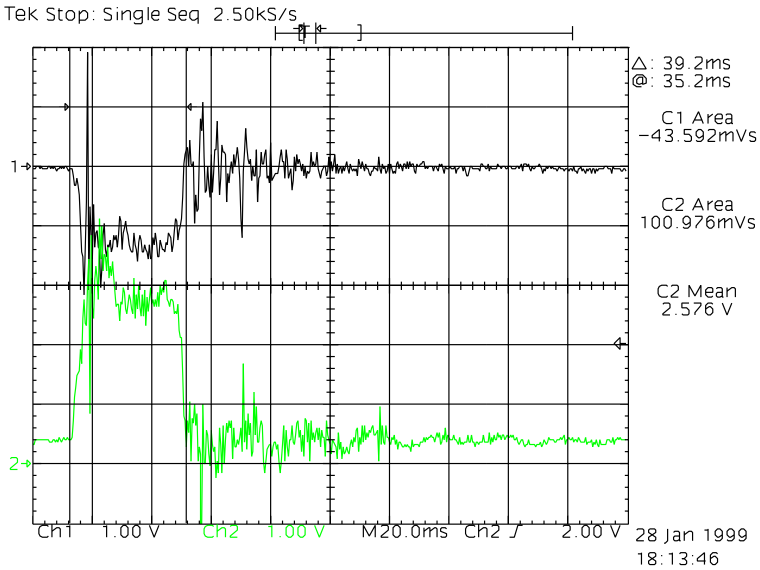


Figure 6.18: X and Y accelerometer output, falling from 50 cm, leg angle of 0.30.

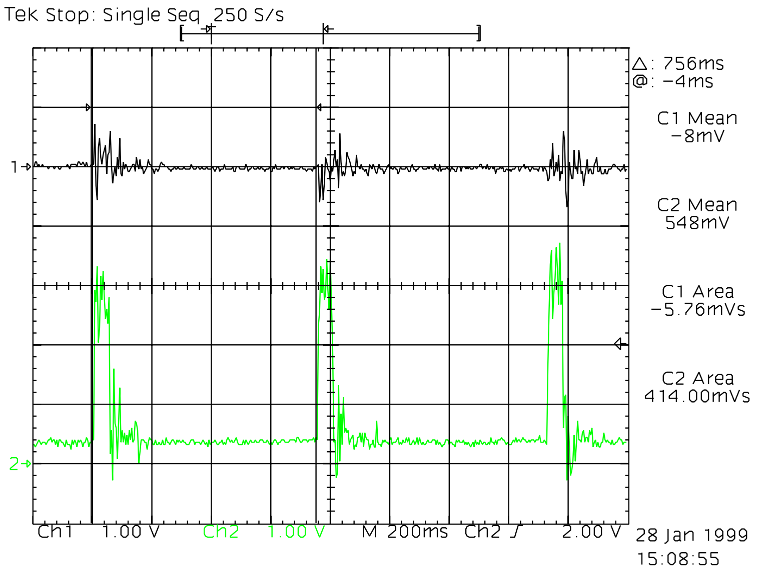


Figure 6.19: X and Y accelerometer output, hopping in place, 200 msec/div.

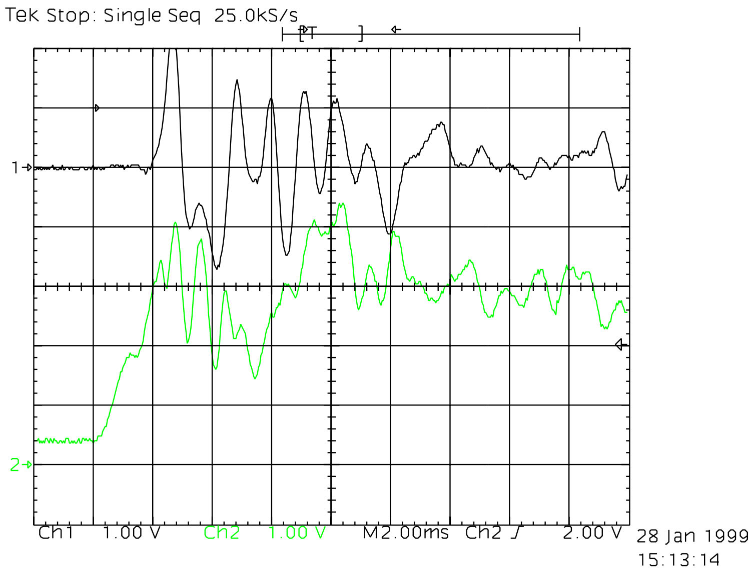


Figure 6.20: X and Y accelerometer output showing landing, hopping in place, 2 msec/div.

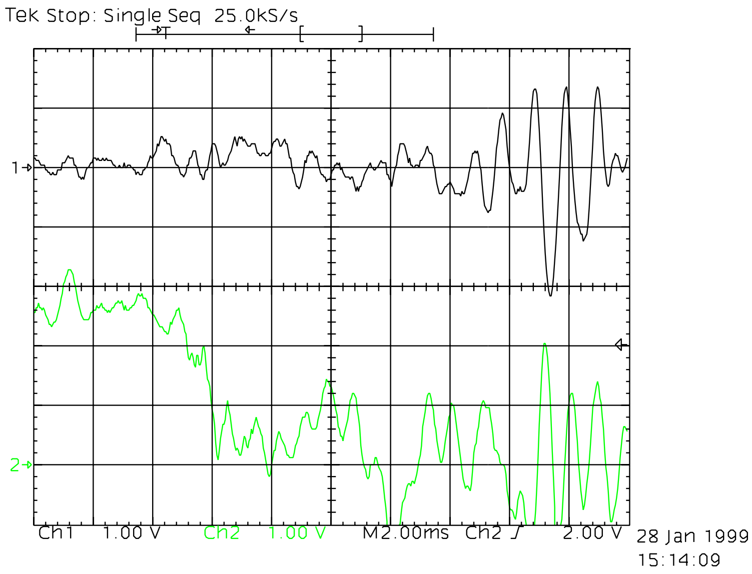


Figure 6.21: X and Y accelerometer output showing takeoff, hopping in place, 2 msec/div.

6.5 Specific Resistance Measurement

This section describes an experiment to measure the *specific resistance* of the Bow Leg Hopper. Specific resistance is a dimensionless metric used to characterize the energy consumption of a vehicle over level ground [GvK50] [Tuc75] [GAB97]. It is closely related to the "net cost of transport" [Ale77], and may be defined as follows:

$$\mathrm{S.R.} = \frac{\mathrm{Power}}{\mathrm{Weight} \cdot \mathrm{Velocity}} \tag{6.1}$$

In a ideal frictionless world this value could be zero. For most vehicles it is a function of velocity; i.e., as they go faster the power consumption isn't linear with velocity.

The experiment to measure a specific resistance function for the Bow Leg Hopper was performed as follows. A low-value resistor was calibrated and installed in series with the battery that supplies the power to the thrust and leg positioning motors. A digital oscilloscope was attached across the resistor in order to measure the voltage drop and infer the current consumption. The battery voltage was measured and a series of runs performed at different velocities. For each run the power consumption was measured using the oscilloscope to calculate the average current over several hops. The data was collected only after the hopper had bounced several times in order to reach a speed equilibrium. Since the battery voltage decreases during operation, the battery voltage was measured again after each set of runs and an average value used in the power calculation.

The sense resistor was determined to be 0.0454Ω by connecting it to a variable supply and taking the least-squares fit of the measured voltage drop at five different current levels. The hopper weight was measured at 6.99 N using a scale under the foot; this measured the weight of the combined hopper and boom in the reduced effective gravity, as supported by the foot.

The quiescent and hopping-in-place consumption were measured first. The quiescent power is the consumption of the motors when no control pulses are applied; it averaged 15 mW. The hopping-in-place power averaged 5.57 W.

The specific resistance as a function of velocity was measured in two sets. The results appear in Figure 6.22 with the sets indicated by different symbols. The power consumed is nearly independent of hopping speed, as shown in the top graph, which leads to a resistance that decreases at higher speeds.

It should be noted that this experiment measured the full power consumption of the motors, not just power delivered into kinetic energy. Most of the energy consumed was expended by the thrust motor holding a steady position against the load of the string, not doing work on the leg. If only kinetic energy is considered, the value may be estimated as follows: a hopper weighing 7 N hopping to 50 cm altitude

(25 cm fall) has 1.75 J kinetic energy at landing; making up for the 20% leg loss requires 0.35 J per cycle; assuming a flight time approximately 800 msec this requires a power of 0.44 W. The leg swing requires minimal power, as confirmed by both direct observation and the result that the power consumption is relatively independent of velocity. The specific resistance is then inversely proportional to velocity: at 1 m/s it is 0.06, at 0.5 m/s it is 0.12, and at 0.0 m/s it is ∞. This theoretical result suggests that mechanical redesign might substantially improve power efficiency.

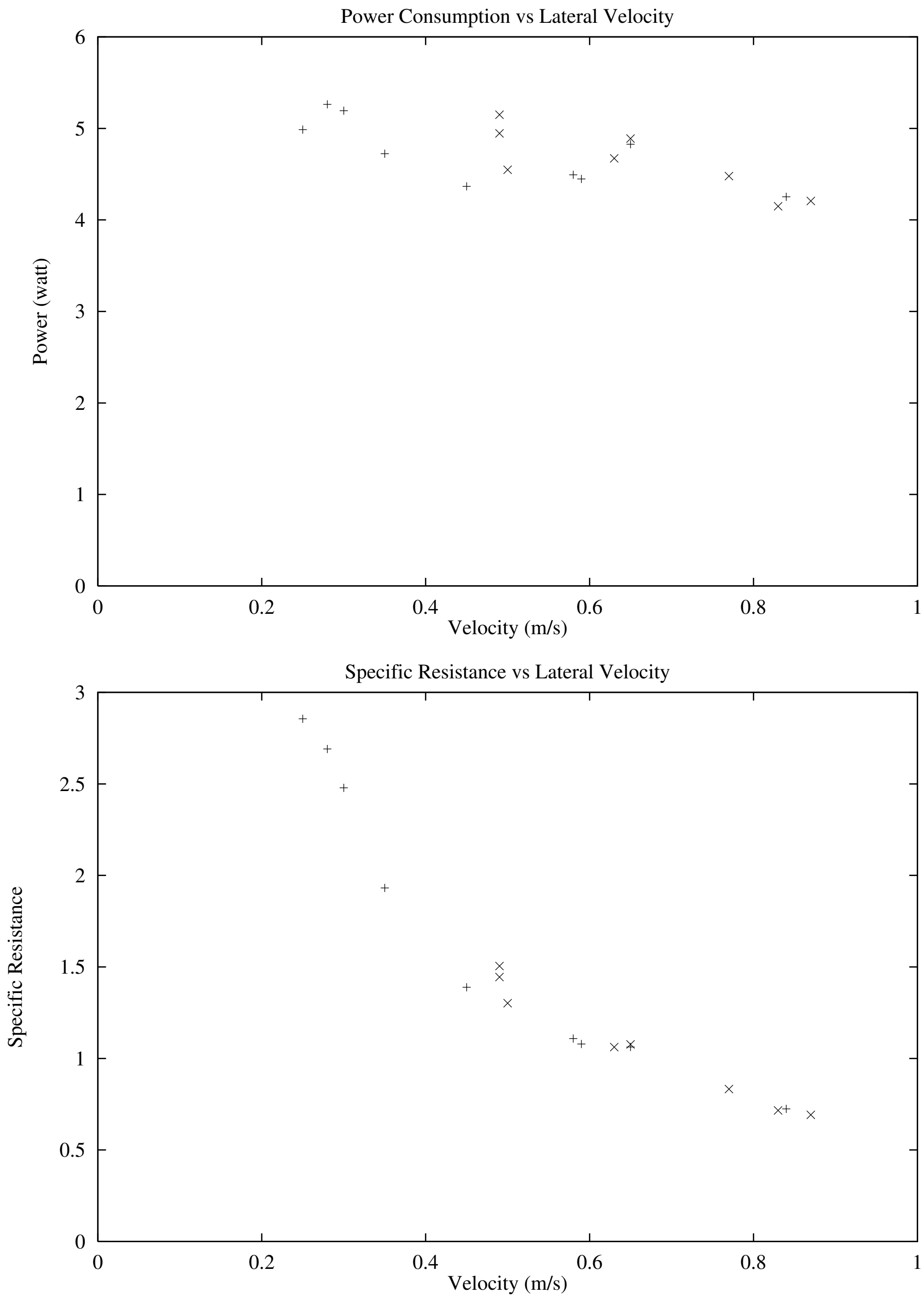


Figure 6.22: Results of the specific resistance experiment. The top plot shows the average motor power as a function of velocity. The bottom plot is the computed specific resistance, based on supply current measurement.

Chapter 7

Discussion

This thesis addresses a mélange of topics: path planning, discrete control, system modelling, programmable mechanism design, and terrain analysis. The unifying idea is "legged locomotion"—by focussing on the demonstration of locomotion tasks with a particular novel hardware design, all of these ideas are developed and the connections between them explored. To place this work in context, the discussion begins by considering the relation to previous work in locomotion.

7.1 Related Work

The field of dynamic legged locomotion is not terribly large. As a result, most of the related work consists of other efforts to build running machines. Also related are juggling efforts, which have similar dynamic properties, and some planning research.

7.1.1 Mechanisms That Run

There have been several efforts to build running robots. Matsuoka [Mat80] built a planar one-legged hopper that operated in low effective gravity on an inclined table. The machine had a short stance time, with thrust provided by a high-force electric solenoid. The following is quoted from [Rai86], as I was unable to find the original source material [Mat79]:

> Matsuoka was the first to build a machine that ran, where running is defined by periods of ballistic flight with all feet leaving the ground. His goal was to model repetitive hopping in the human. He formulated a model consisting of a body and one massless leg and he simplified the problem by assuming that the duration of the support phase was short compared with the ballistic flight phase. This extreme form of running,

> for which nearly the entire cycle was spent in flight, minimized the influence of tipping during support. This model permitted Matsuoka to derive a time-optimal state feedback controller that provided stability for hopping in place and for low speed translations (Matsuoka 1979).
>
> To test his method for control, Matsuoka built a planar one-legged hopping machine. The machine operated at low gravity by lying on a table inclined 10° from the horizontal rolling on ball bearings. An electric solenoid provided a rapid thrust at the foot, so the support period was short. The machine hopped in place at about 1 hop/s and traveled back and forth on the table. [Rai86]

Following Matsuoka were a series of running machines produced by Marc Raibert's Leg Lab [Rai86], first at Carnegie Mellon University and then at MIT. The first machine was a pneumatically actuated planar monopod. Following were a hydraulically actuated 3D monopod, planar biped, 3D quadruped, and 3D biped. Each used a telescoping leg with an internal air spring for compliance in series with a hydraulic thrust actuator. Experiments performed with these machines included low and high speed running, jumping over obstacles, climbing stairs, and a forward flip. Two machines with revolute joints were the planar monopod and the planar Uniroo. The planar monopod used revolute joints at the hip and ankle and a flexible fiberglass leaf spring for a foot. The foot provided the compliance for hopping, and was actuated hydraulically through a rigid tendon. The Uniroo [Zeg91] added revolute knee and tail joints, and used a rigid foot actuated hydraulically with a compliant tendon composed of steel springs. Gill Pratt now heads the MIT Leg Lab and is working on topics such as series-elastic actuation for locomotion [PW95] and virtual model control [PDP97].

All the Raibert machines were controlled by the same basic decomposition into three independent linear controllers: forward velocity controlled by foot placement, hopping height controlled by thrust, and pitch controlled by hip torque during stance. The Uniroo used its extra degree of knee freedom to minimize knee torque by keeping the knee along the line between toe and hip. In all Raibert machines, the control involved high force and power during stance.

Following Raibert are several examples of electrically actuated hoppers. Papantoniou constructed a one-leg electrically actuated planar hopper with a leg constructed from a four bar linkage with a tension spring [Pap91]. The motor for the vertical oscillation used a self locking lead screw. He considered the linkage design problem at length in order to limit the forces on the electric motors while maintaining reasonable kinematics.

Martin Buehler's group at McGill also moved to electric motors with a one-leg planar hopper similar to Raibert hoppers but with electric motors instead of hydraulics and a metal spring instead of an air cylinder [GAB93]. The machine used a Raibert style three-part control modified to distribute thrust over the hopping cycle to accommodate the lower power electric motors. They also added compliance in series with the hip actuator to create an oscillation to sweep the leg [AB97]. This made the leg sweep more efficient but required control to synchronize the hip oscillation and the vertical hopping oscillation.

Berkemeier and Desai at Boston University designed an electrically actuated leg with three revolute joints that uses an electric motor coupled with elastic tendons to drive the foot [BD96]. In this paper is discussed the design and open-loop one DOF testing.

Lebaudy, Prosser and Kam at Drexel designed an electrically actuated telescoping leg constrained to the vertical [LPK93]. It incorporated a DC motor driving a ball screw in series with a steel spring. The goal was to test different vertical height controllers. Their controller computed the inverse of an approximate model of the bounce then added integral error feedback to compensate for varying body mass.

Robert Ringrose at the MIT Leg Lab has developed a self-stabilizing hopping monopod powered by a single constant speed motor that regulates altitude and body attitude using purely mechanical feedback [Rin96][Rin97]. He shows the conditions under which the altitude oscillation will phase-lock to the excitation. Simply put, when hopping too low, the foot lands early and this allows more work to be done by the motor, which increases the altitude and brings the vertical oscillation closer to the motor phase. A similar principles regulates the upper disturbances. The scheme is very similar to the juggler described in [SA93]. The attitude stability is provided by a large, curved foot that is shaped to cause a restoring torque of reasonable magnitude for each possible pitch error.

There is a substantial body of work analyzing and simulating legged robots. One related example is a study by Lapshin at the Russian Academy of Sciences that analyzed a hopper model with a telescoping leg attached to a body with an actuated weight [Lap92] intended to control pitch. He developed a formal analysis of hopping height and forward speed control that confirms Raibert's methods.

7.1.2 Bow Leg Mechanism

The Bow Leg Hopper has a morphology dictated by the process of running: it has a body, a leg, and a spring for energy storage. In many ways the design is a descendant of the work of Marc Raibert and the others described in the previous section. However, the freely pivoting hip and the passive body stability are a profound departure.

The free hip decouples the body from the leg disturbance torques and enables the passive stability created by placing the body mass below the hip. The free hip also allows control using low-torque kinematic reconfiguration during flight. These ideas extend beyond the Bow Leg and could be applied to other leg designs. However, the Bow Leg also has the efficiency and low mass that minimize the power and force requirements and enable the machine to operate using on-board batteries.

Another key difference is the use of the leg for energy storage. By moving the energy input to the flight interval the peak power is greatly reduced. This is an idea that could be widely applied to other mechanisms with intermittent contact.

Like much other work the prototypes were limited to a plane. This simplifies 3D engineering problems such as controlling yaw and roll or sensing pose from an autonomous vehicle. Related work [Rai86] has shown that 2D locomotion results transfer well to 3D.

The hopper can operate with the basic linear control outlined in Chapter 4 or the planning system in Chapter 5. The linear control is very similar to Raibert three-part control: the energy regulation is similar to the Raibert altitude control and the foot placement is the same as the Raibert running speed control. The third part, attitude control, is performed mechanically via the passive body stability.

The most obvious control differences result from the displacement of control to flight, which reduces the feedback rate to once per bounce. The physics is rephrased as a mapping from one trajectory to another, and the control as computation of a discrete parameter vector. A side benefit is that sensor bandwidth is much lower in the absence of high-speed feedback loops.

However, the emphasis of the control has shifted from controlling a steady state oscillation to performing intricate tasks by dextrous control of each step. The new approach is based on physical modelling and state space path planning. Some demonstrations performed by previous machines (running, stairs, etc.) were repeated but with a minimum of special programming; the human specifies the task as a few constraints and the machine automatically generates plans for a variety of performances. This is a step toward automatic solution of natural terrain, treated in the following section.

7.1.3 Rough Terrain Locomotion

There is a great deal of literature concerned with the navigation of rough terrain using statically stable walking machines. However, this body of work is largely concerned with topics not relevant to dynamic locomotion, such as selection of foot placements given a kinematic body path or leg force control for terrain adaptation (e.g. [Hir84]).

The most significant body of work related to dynamic terrain navigation was performed by Jessica Hodgins and Marc Raibert as documented in her Ph.D. thesis [Hod89] and several papers [Hod88][HR91]. The thesis names many of the general issues, then focuses on the problem of controlling step length as a necessary precursor to placing a foot at precise locations.

The work addresses three approaches to controlling step length: control of forward speed, running height (i.e., vertical speed), and stance duration (i.e., leg stiffness). Forward speed and running height are two parts of the Raibert three-part control, and the stance duration involves mechanical adjustment of the leg spring pneumatic pressure. The difficulty lies in adjusting any one of these parameters without disturbing the stability of the other independent control loops. The work includes analyses and experiments to measure the precision of the different methods and concludes that "the forward speed method produced the widest range of step lengths." [HR91]

The work also uses step length control to address a few laboratory terrain problems. The problem of stepping on a single point is solved by coming "in-phase" with the obstacle by adjusting the step lengths over the few flights prior to landing on the point, either incrementally or on one particular landing. Similarly addressed are leaping over a single obstacle and climbing and descending three steps.

For the Bow Leg Hopper, the step length problem is addressed using the empirical model discussed in Chapter 3 to compute the control parameters that will result in the desired flight parabola and thus control the step length. This is simpler with the Bow Leg mechanism than the Raibert machines since energy and directional control are physically decoupled. Essentially, the takeoff velocity vector can be controlled in the polar coordinates of magnitude (related to energy) and direction, whereas the Raibert formulation involved the Cartesian coordinates of horizontal and vertical velocity. The polar formulation reduces the coupling between the control axes as the step length is varied. The side effect is that the step length and hopping height are constrained by the total energy, which has a dynamic range limited by the dissipation and thrust limits.

The difference in the terrain experiments is one of generality. The Hodgins work examines specific patterns of step length adjustment as motivated by animal and human experiments. In contrast, the planning algorithm in this thesis can consider any path compatible with the dynamics and the terrain (subject to time and heuristic limitations). In principle this approach could produce any of the Hodgins solutions given the right cost functions. The planning solution also considers multiple constraints along the path and can operate on arbitrary terrain within some assumptions. Of course, accommodating all cases in a general approach is difficult. This thesis doesn't fully answer the problem of generality; some of the Hodgins experiments were loosely replicated with the planning control (stairs, single foothold, etc.), and the others involve only moderately more complex terrain (i.e. six or ten irregularly spaced level stepping stones).

Part of the benefit of the Bow Leg is the simplicity of the dynamics that makes the closed-form empirical models possible. However, this is not the first attempt to model a leg in closed form. Schwind and Koditschek [SK97] have an approximate closed-form analysis of a simplified hopper with a finite stance time for particular leg potential functions. However, the choice was made for this thesis to solve the problem pragmatically as outlined in Chapter 3.

7.1.4 Biological Studies

There have been a number of studies of the biomechanics of running behavior in both animals and humans. The topics have included kinematic measurement, energy consumption and efficiency, and inference of control strategies. Much of the difficulty is simply acquiring data about the internal state of the creature from external measurements such as images, force plate recordings, and oxygen consumption [AV75] [McM84]. In a few cases, direct internal measurements have been possible [RMWT97].

The Bow Leg Hopper is only loosely modeled on biological systems and so the fundamental issues are shared but many of the details of these studies are only suggestive. For example, Warren [WYL86] studied step length regulation in humans running on a treadmill, and determined they primarily controlled step length by varying vertical impulse. As described above, the Bow Leg Hopper controls the magnitude and direction of takeoff velocity so the problem is cast in different terms.

The important roles of elasticity and negative work in animal locomotion is well-recognized [CHT77] [McM85]. Right on the boundary between the robotics literature and the biological world is an paper by Alexander [Ale90] that discusses three fundamental uses of springs in legged locomotion: vertical rebound, leg sweep, and foot contact. The paper discusses the idea of negative work in leg actuators and proposes various solutions, emphasizing the use of a large leg spring. The Bow Leg is a response to this work by simplifies the entire leg down to a single large spring. The idea of a leg sweep spring that would conservatively accelerate and decelerate the leg sweep is not addressed in this thesis but has been analyzed by Ahmadi and Buehler [AB97]. The third spring application is the foot pad, which is answered in the Bow Leg design by having a low mass foot and a simple rubber covering.

7.1.5 Planning

Most of the path planning work discussed in Chapter 5 is concerned with choosing heuristics that narrow the search space sufficiently for the planner to operate in real time. The algorithm itself is standard, but the fact that the system is real time means it may be seen as an elementary case of an *anytime planner* as defined by Dean and Boddy [DB88] [BD89]. The algorithm is time-dependent since it has limited time available (i.e., the flight interval) and produces results that improve with increased computation time (i.e., as the search deepens). The cost function allows the planner to select successively better results and utilize the best result found so far at any point in the planning process.

However, the system doesn't use any form of metalevel control, e.g. performance profiling as a means of allocating computation time, or more elaborate methods [ZR93]. The reason is that the time available is determined solely by the physical state of the machine, not by the planning process. There is no choice to act early so there is no tradeoff between early termination of planning and increased plan quality. Rather, the cost of computation is opportunity cost: a fixed time that can be spent using whatever algorithm will produce the best result in that interval. Given that only one planning algorithm was implemented, this choice is trivial. Still, the anytime formulation is convenient since no explicit attention must be paid to estimating planning time; the control output is available whenever the hopper needs it. A good general reference on these issues was written by Zilberstein [Zil96].

Likewise, the planning system exhibits an interleaving of planning and execution [Nou97], again without metalevel control. Interleaving has several purposes. The general use is to simplify planning problems by computing a partial plan up to a subgoal then actually executing it before continuing planning. This can enable additional sensing to further refine the planning problem or accommodate uncertainty in execution. It can also enable a time-dependent tradeoff on plan quality, or allow "pipelining" if computation can continue during execution.

The hopper planning system is primarily governed by the flight time limit but in the course of operation makes use of these properties. The subgoals are not explicitly chosen but are simply the best solutions found after each planning episode. Since the hopper must execute the plan immediately, new sense data is obtained and the next planning iteration may begin from a new state, presumably closer to the goal. This process could be improved, however, by more explicit attention to time-dependent planning. The graphical methods posed in Chapter 4 could be used to generate intermediate goals or to suggest skeleton plans. It would be worthwhile allocating time between a long-range coarse planning process that finds globally feasible paths and the short-term foot placement planning. Each of these processes would need to be structured as interleaved planning to accommodate the influx of terrain sense data.

There has also been some work in the graphics literature planning simulated dynamic motions across terrain [HvdP96]. The design constraints are different since planners for graphics do not need to operate in real time and the simulated sensors are usually precise. The Huang and Panne paper cited presents search algorithms that plan dynamic motions for a Luxo lamp and an Acrobot. A low-level control search slices time into uniform intervals, and a high-level terrain planner chooses an ordered sequence of actions from a set of five types of jumping motion. The resulting search allows the simulation to anticipate terrain features, but runs much slower than real time. In contrast, the Bow Leg planner takes advantage of the periodic nature of hopping to produce a significantly shallower search tree and works in real time as the hopper moves.

7.1.6 Other Mechanisms

There are several other lines of work that deserve mention for inspiring some of the principles embodied in the Bow Leg Hopper.

Juggling is hopping turned upside down. In the juggling domain, Koditschek and his students at Yale and University of Michigan have developed surprisingly simple "mirror law" control strategies for bounce juggling. In [BRK99] is discussed an idea for backchaining locally stable control laws from a goal to guarantee convergence over a large domain. The hopping analog would be to construct a set of functions for known terrain that map every hopper trajectory to a subset that moves toward the goal. A partial solution that addresses foot placement is discussed in [SK95].

Another manipulation example is Wes Huang's thesis on impulsive manipulation [Hua97], concerning a sliding object manipulated by an actuator that delivers precisely calibrated taps. The actuator was controlled by storing energy in a spring, positioning it near the object, and releasing a small hammer. This actuation uses a similar principle as the Bow Leg since it is programmed at low power to control a high-energy event. The modelling difficulty is backwards, though, since the "flight phase" sliding across the table is difficult to predict and the "stance phase" during the actuator collision is relatively well understood.

Tad McGeer at Simon Fraser University has developed a series of passive bipedal walkers that walk down fixed angle slopes using purely mechanical feedback [McG89] [McG90]. Although this dissertation is concerned with running, not walking, this work deserves particular mention for its appealing simple design.

Finally, an amusing extreme of the hopping machine literature is the hopping tank described in the patent [Wal]. The illustrations describe a cylindrical armored body large enough to carry six gunners and a pilot. A single telescoping leg may extend from the bottom of the body to lift it off the ground and hop across rough terrain. The leg is powered by internal fuel combustion using a two cycle Diesel principle: the ground impact compresses the fuel-air mixture, which combusts and expands to perform work during liftoff. The body has large gyroscopes for attitude stability. Despite the fanciful concept and illustrations, the machine does incorporate a self-timing principle similar to the Bow Leg; instead of the Bow String slipping off a pulley to release leg energy, the fuel mixture is compressed to the combustion conditions.

7.2 Future Work

The most exciting future work would be development of an untethered 3D Bow Leg Hopper. However, some other interesting experiments for the planar version would be as follows:

- Demonstrate richer terrain crossing tasks with steep stairs, high walls, and narrow footholds.
- Define gaits as constraints on the planner, e.g. "alternate short and long steps."
- Simulate the use of terrain sensing by modifying the terrain model during execution. The planner operates on-line so this would be a modest extension.
- Try storing substantial energy in horizontal motion. This energy could be employed for hill climbing or long jumping, or converted to vertical motion in a "pole vaulting" mode.
- Overhead obstacles. This constrains the hopping height and might force trading off vertical velocity for horizontal velocity.
- Tilted stepping stones. These are stones which are straight but inclined at an angle.
- Gymnastics. Perform a forward flip. Kick bounce off a wall.

The planner could further develop along several avenues. The planning process could include uncertainty estimates to indicate potentially risky plans. The planner could consider energy constraints so plans would anticipate maneuvers that impose bounds on total energy. A maneuver such as crossing a high wall might require several bounces to build up high energy. Similarly, running under an overhang might require several dissipation steps to reduce the total energy. The terrain model will be refined to include obstacles such as walls or overhangs.

Some other interesting questions:

1. What would be the advantage of adding a tail?
2. Could the leg have a spring loaded sweep mechanism for high speed travel?
3. Could non-contact pitch control mechanisms, e.g., reaction wheels, be used for stability with extremely long flight times?
4. Could the body be free to rotate more than 2π? Could significant energy be stored in body rotation?

5. Could pitch be controlled by a spring mechanism that exerts torques directly on the leg? Could the Bow Leg spring itself be used for this using tangential loading?

7.3 Comments

An underlying message of this dissertation is to pay careful attention to mechanical design. The Bow Leg design elegantly fits the problem—it is efficient, stable, and energetic. It is a programmable mechanism that extends the control path from computation to electrical actuation to mechanical actuation that applies spring and bearing forces. By considering the entire pathway the design takes advantage of the dynamics to handle high mechanical power with very low control power.

This programmable mechanism idea is not unfamiliar; designers of low-cost devices like answering machines frequently use complex cams and small actuators to control mechanisms powered by a single motor. But applying the same notion to dynamic systems, like fast-moving running robots or manipulators, can shift dynamic control into the mechanical realm and achieve a balance between actuators and the natural dynamic processes.

On a broader note, the Bow Leg mimics an essentially biological activity without mimicking biological form. It also performs a robotic function—computer controlled activity—while confirming that a robot is more that just a computer interface to the world. The lesson is that *mechanism can think*. We can use our insight to choose the mechanism that interacts with the world with the behavior we need; a mechanical oscillator with the right feedback forces is a computer that directly connects action and computation.

Ultimately, this dissertation is about a future of practical legged robots that extend our ability to send machines to any corner of this world or another. The Bow Leg is an elegantly simple idea that can make this future happen.

Appendix A

Symbols

Physical Constants and Machine Parameters	
M	body mass
m	same as M
g	gravitational acceleration (effective)
ϵ	leg spring restitution (v_{nr}/v_{ni})
k	stiffness of a linear spring leg
F_0	force of a constant force leg spring
ρ	radius of gyration of body
μ	coefficient of Coulomb friction at foot
Kinematic and State Variables	
(x, y)	horizontal and vertical center of mass position
θ	body pitch angle with respect to (wrt) world coordinates
ϕ	leg angle wrt world frame
l	leg length
$\dot{x}$	horizontal body velocity
$\dot{y}$	vertical body velocity
$\mathbf{v}$	body velocity vector
$\dot{\theta}$	body angular velocity wrt world coordinates
ω	same as $\dot{\theta}$
$\ddot{x}$	horizontal body acceleration
$\ddot{y}$	vertical body acceleration
(x_n, y_n)	position of body at apex of flight n
ϕ_n	leg position wrt world at impact n
Δx	lateral distance the hopper falls from apex to impact
Δy	vertical distance the hopper falls from apex to impact
Δl	maximum leg compression during stance
r	the moment arm of ground force on the COM

continued on next page

$\mathbf{r}$	the vector from the COM to the toe
h	the translation of the hip wrt COM
$h_{\max}, h_{\min}$	range of possible hip translations
y_{gnd}	altitude of ground under foot
$\mathbf{v_i}$	body velocity vector at impact
$\mathbf{v_r}$	body velocity vector at 'rise' (i.e., takeoff, liftoff)
ω_i	angular velocity at impact
ω_r	angular velocity at takeoff
Energy	
ΔE	general change in energy, lumping gain and loss
ξ	energy stored in thrust mechanism
ξ_n	energy stored in thrust mechanism prior to impact n
E_{diss}	energy dissipated by restitution loss
E_i	kinetic energy at impact
E_r	kinetic energy at rise
$E_{s\max}$	maximum energy stored by leg during stance
y_m	maximum altitude available given energy constraint
$\dot{x}_m$	maximum lateral velocity available given energy constraint
Force, Torque, Time	
τ	the torque on the center of mass
$\mathbf{F}$	ground force vector
F_n	normal component of ground force
t_s	stance duration
t_f	flight duration
t_{fall}	time from apex to ground contact
t_r	time from liftoff to apex
Restitution and Impulse	
$\hat{\mathbf{T}}$	unit tangent vector perpendicular to the leg
$\hat{\mathbf{N}}$	unit normal vector along leg axis
(v_t, v_n)	body velocity in leg frame (i.e, tangent, normal)
(v_{ti}, v_{ni})	impact velocity in leg frame
(v_{tr}, v_{nr})	takeoff velocity in leg frame
v_{nc}	normal velocity of body at maximum compression
I_c	compression impulse
I_n	normal impulse: total restitutive impulse change
I_r	restitution impulse
ϵ_t	fictional thrust restitution
$(\Delta v_t, \Delta v_n)$	change in velocity in leg frame
$\Delta\dot{\theta}$	change in angular velocity

continued on next page

Graphical Methods	
ν_i	angle of body velocity at impact
ν_r	angle of body velocity at takeoff
ϕ_μ	angle that is half the width of the friction cone
Empirical Modelling	
ϕ_{sweep}	leg sweep angle during stance
$\phi_{\text{effective}}$	'virtual' leg angle at midpoint of sweep
k_{sweep}	tangent velocity coefficient for leg sweep
ϕ_i	leg position at impact
ϕ_r	leg position at takeoff (after sweeping during stance)
u_ϕ	the leg mechanism angle (wrt body)
$\ddot{x}_{\text{lat}}$	parameter for lateral acceleration during flight
u_e	the thrust mechanism control value (proportional to angle)
p_1	linear term for thrust calibration
p_2	quadratic term for thrust calibration
$(\sigma_x, \sigma_y, \sigma_{\dot{x}})$	standard deviations of error in apex prediction
Sensor Filtering	
x_{raw}	raw lateral position sensor reading
y_{raw}	raw vertical position sensor reading
$\tilde{x}$	difference between estimated and measured x position
$\tilde{y}$	difference between estimated and measured y position
Δt	sensor sampling interval
$\ddot{y}_{\text{est}}$	estimate of y acceleration for feedforward
k_1	feedback gain for position estimator
k_2	feedback gain for velocity estimator
$\vec{x}_{\text{raw}}$	set of raw sensor readings for parabolic fit
$\mathbf{A}$	fitting matrix for parabolic fit
$\mathbf{B}$	basis function matrix for parabolic fit
t_{peak}	time of the apex wrt clock
t_p	time of the apex wrt center of data set
Control and Planning	
$f()$	abstract plan mapping apex state to control values
$P()$	abstract plan mapping one apex state to the next
q_{te}	total energy in m^2/s^2 units (i.e., normalized for M)
$\bar{\dot{x}}$	desired x velocity
$\bar{x}$	desired x position
$\bar{q}_{\text{te}}$	desired total energy
k_{xdx}	distance to velocity gain for linear controller
k_{xdphi}	velocity to leg angle gain for linear controller
k_{thrust}	gain for linear thrust regulator

continued on next page

$\dot{x}_m$	maximum lateral velocity possible given energy
$\dot{x}_{\text{err}}$	error in lateral velocity
score	heuristic evaluation value for a search node
cost	total heuristic cost of a bounce
cost_f	cost component representing friction limits
cost_v	cost component constraining takeoff velocity
cost_r	cost component representing foothold risk
cost to go	estimated cost to goal
ϕ_{norm}	leg angle normalized to friction cone
$\text{footx}_{\text{norm}}$	foot placement normalized across foothold boundary
k_{xcost}	heuristic scaling factor for cost-to-go
x_d	position goal in old planning heuristic
$\tilde{x}$	position error in old planning heuristic
k_v	velocity gain in old planning heuristic
k_l	path length penalty in old planning heuristic
p	number of bounces in path for old planning heuristic
Experimental Data	
`q.x`	body lateral position
`q.y`	body vertical position
`q.theta`	body angular position
`inx`	state machine index
`u.phi`	commanded leg angle (wrt world)
`q.phi`	measured leg angle (wrt body)
`qd.x`	body lateral velocity
`qd.y`	body vertical velocity
`qraw.y`	body vertical position sensor reading
`qraw.y`	body angular position sensor reading
S.R.	specific resistance
Design Analysis	
KE	kinetic energy
P_{motor}	average motor power
y_{min}	minimum stable hopping altitude
ψ	thrust mechanism drive angle
λ	length scaling factor

Appendix B

Design Analysis

In the following section I work out the basic relations that express the forces and energy changes that occur during hopping. The example considered is a Bow Leg Hopper bouncing in place in equilibrium. The leg is idealized as a massless constant force spring with non-ideal restitution. This leg does dissipate energy but the controller is assumed to exactly compensate using thrust.

Following the analysis is a discussion of scaling issues and some worked examples for different parameter values.

B.1 Body Dynamics

The hopper falls from altitude y and first touches the ground when the center of mass is at the height of the leg length l; the total flight time is t_f:

$$\Delta y = l - y = 1/2\ g \left(\frac{t_f}{2}\right)^2 \tag{B.1}$$

Gravitational acceleration g is negative, as is fall distance Δy and the change in leg length Δl. The vertical velocity at impact:

$$\dot{y}_i = -\sqrt{2g\Delta y} \tag{B.2}$$

Solving for the full flight time (up and down):

$$t_f = \sqrt{\frac{8\Delta y}{g}} \tag{B.3}$$

The momentum at impact is exactly reversed by the constant force leg spring applied over the stance interval:

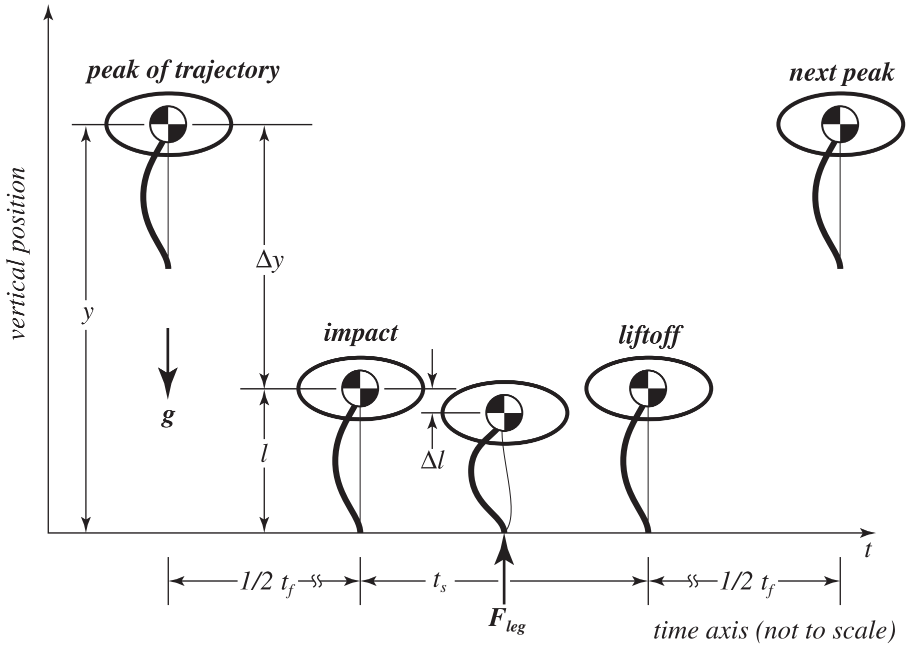


Figure B.1: Variables used in the design discussion.

$$M\Delta\dot{y} = F_0 t_s = -2M\dot{y}_i \tag{B.4}$$

Solving for the stance time:

$$t_s = \frac{-2M\dot{y}_i}{F_0} = \frac{M}{F_0}\sqrt{8g\Delta y} \tag{B.5}$$

Stance time is a function of altitude, but the ratio of stance time to flight time depends only on the constant forces:

$$\frac{t_s}{t_f} = \frac{M}{F_0}\sqrt{8g\Delta y} \cdot \sqrt{\frac{g}{8\Delta y}} = \frac{Mg}{F_0} \tag{B.6}$$

The kinetic energy at impact:

$$\mathrm{KE}_i = \frac{M\dot{y}_i^2}{2} = Mg\Delta y \tag{B.7}$$

In this notation, the total energy change is the sum of restitutive losses and thrust ξ, which is regulated to zero by the controller:

$$\Delta E = E_i - E_{\text{diss}} + \xi = 0 \tag{B.8}$$

The following considers the case with no thrust to develop the dissipation expressions. With an instantaneous impact and no thrust, the leg spring is assumed to exhibit an impulse restitution ϵ. This implies energy loss $(1 - \epsilon^2)$:

$$M\dot{y}_r = -\epsilon M\dot{y}_i \tag{B.9}$$

$$\mathrm{KE}_r = \frac{1}{2}M\dot{y}_r^2 = \epsilon^2\frac{1}{2}M\dot{y}_i^2 = \epsilon^2\mathrm{KE}_i \tag{B.10}$$

$$E_{\mathrm{diss}} = \mathrm{KE}_i - \mathrm{KE}_r = (1 - \epsilon^2)\mathrm{KE}_i \tag{B.11}$$

(The subscript r indicates "rise" since "l" for liftoff would be similar to i for "impact.")

The following assumes a finite stance duration to develop the leg compression expression. The leg compresses to store kinetic energy in leg potential energy; the energy stored was gained from the potential change from the highest trajectory point to the point of maximum compression:

$$Mg\Delta y + Mg\Delta l = -\Delta l F_0 \tag{B.12}$$

Solving for the change in leg length as a function of equilibrium hopping altitude:

$$\Delta l = \frac{-Mg\Delta y}{F_0 + Mg} \tag{B.13}$$

Actually, since the leg is dissipating some of that energy as it is stored, this underestimates the leg compression, but assuming that the dissipation is low the effect of dissipation on the estimate of leg compression may be neglected.

B.2 Thrust Mechanism

The thrust mechanism may use the entire flight time to store energy in the leg. The thrust energy provided to the leg at equilibrium is exactly equal to the dissipation. The average net motor power (after frictional losses) can be estimated using the flight duration:

$$P_{\mathrm{motor}} = \frac{\xi}{t_f} = \frac{(1 - \epsilon^2)Mg\Delta y}{\sqrt{\frac{8\Delta y}{g}}} = (1 - \epsilon^2)M\sqrt{\frac{g^3\Delta y}{8}} \tag{B.14}$$

The required thrust energy increases monotonically with altitude, as does the average motor power.

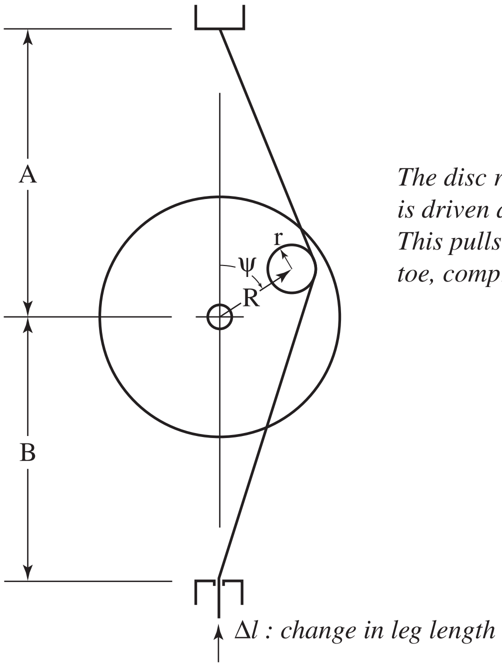


Figure B.2: Kinematics of the thrust pulley. As ψ increases, the pulley is moved into the string, which pulls the string in from the leg.

But the mechanical advantage of the thrust mechanism varies and so the motor load varies during the leg compression. The full form of the relation between motor angle and leg compression is as follows:

$$
\begin{aligned}
\Delta l \;=\; & A + B && -\sqrt{A^2 - 2\,AR\cos(\psi) + R^2 - r^2} \\
& && +\sqrt{B^2 + 2\,BR\cos(\psi) + R^2 - r^2} \\
& + 2\pi r && -r\arcsin\left(\frac{\sqrt{A^2 - 2\,AR\cos(\psi) + R^2 - r^2}}{\sqrt{A^2 - 2\,AR\cos(\psi) + R^2}}\right) \\
& && -r\arcsin\left(\frac{\sqrt{B^2 + 2\,BR\cos(\psi) + R^2 - r^2}}{\sqrt{B^2 + 2\,BR\cos(\psi) + R^2}}\right) \\
& && -r\arccos\left(\frac{-AB - AR\cos(\psi) + R^2 + BR\cos(\psi)}{\sqrt{A^2 - 2\,AR\cos(\psi) + R^2}\sqrt{B^2 + 2\,BR\cos(\psi) + R^2}}\right)
\end{aligned}
\tag{B.15}
$$

This function (generated using Maple) describes the mechanism illustrated in Figure B.2. A useful approximation is to set the pulley radius r to zero:

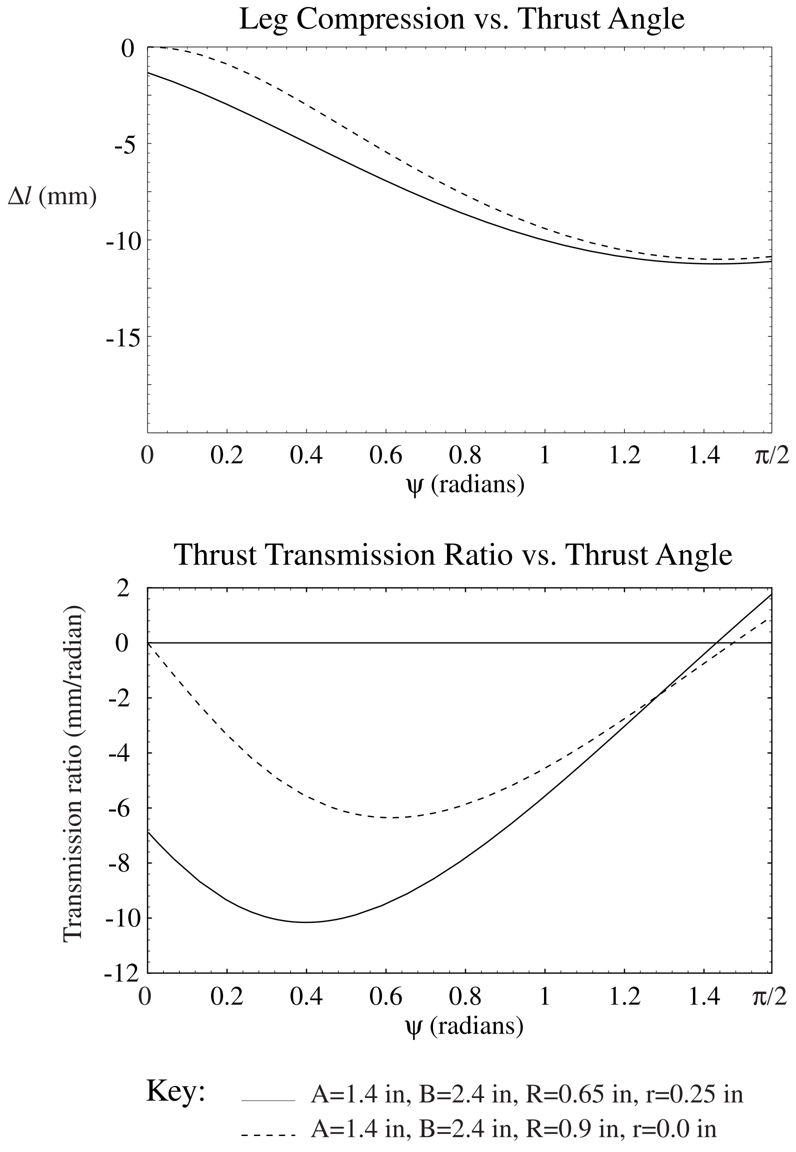


Figure B.3: Kinematics of the thrust pulley. The top figure plots the length of string retracted for each thrust disc angle; the bottom figure plots its first derivative. Two cases are shown with different linetypes illustrating the effect of the drive pulley radius. Note how the compression curve flattens at the right; this is a stable region in which the leg can be held in compression with minimal torque.

$$\Delta l = A + B - \sqrt{A^2 - 2AR\cos(\psi) + R^2} + \sqrt{B^2 + 2BR\cos(\psi) + R^2} \quad \text{(B.16)}$$

The transmission ratio is the derivative of the leg displacement with respect to thrust motor angle. For simplicity, it is only shown for the case with pulley radius r equal to zero:

$$\frac{\Delta l}{\Delta \psi} = \frac{BR\sin\psi}{\sqrt{B^2 + R^2 + 2BR\cos\psi}} - \frac{AR\sin\psi}{\sqrt{A^2 + R^2 - 2AR\cos\psi}} \quad \text{(B.17)}$$

Examples of the displacement curve and transmission ratio curve are illustrated in Figure B.3.

The peak of the transmission ratio curve determines the maximum torque supported by the motor. The flight time determines the time available for rotating 90 degrees to recontact the string (with no load) and then another 90 degrees under varying load to store energy. A conservative solution is to guarantee that the motor could perform this cycle under a constant maximum load for the duration of a minimum altitude hop:

$$\frac{\frac{1}{2}\pi}{\omega(\tau = 0)} + \frac{\frac{1}{2}\pi}{\omega(\tau_{\max})} < t_f \quad \left(t_f = \sqrt{\frac{8(y_{\min} - l)}{g}}\right) \quad \text{(B.18)}$$

A more optimized solution could integrate the modelled motor velocity under the varying load and find a somewhat smaller motor that could still operate within the flight time.

B.3 Quick Analyses

The following analysis illustrates the design parameters for the boom-mounted Bow Leg hopper prototype hopping in 35% effective gravity.

Quantity	Variable	Value
mass	M	2.5 kg
leg length	l	0.25 m
gravity	g	3.4 m/sec^2
leg force	F_0	150 N
hopping height	y_m	0.5 m
energy restitution	ϵ^2	0.8

Using the formulas in the previous section, the following values may be obtained for hopping at the height y_m:

Quantity	Variable	Estimated Value
stance time	t_s	0.04 sec
flight time	t_f	0.77 sec
impact velocity	$\dot{y}_i$	-1.3 m/sec
energy stored	E	2.3 J
leg compression	Δl	0.015 m
energy dissipated	E_d	0.43 J
average motor power	P_{motor}	0.55 W

In practice the measured peak motor power is about 5 Watt, an order of magnitude higher than the P_{motor} estimate. However, the motor is a hobby servo that expends about that much just holding a position under the load of the full leg force at the point of minimal mechanical advantage.

The following example estimates parameters for a hypothetical hopper operating in full gravity.

Quantity	Variable	Value
mass	M	10 kg
leg length	l	1 m
gravity	g	9.81 m/sec^2
leg force	F_0	500 N
hopping height	y_m	3 m
energy restitution	ϵ^2	0.8

Plugging in the values:

Quantity	Variable	Estimated Value
stance time	t_s	0.25 sec
flight time	t_f	1.28 sec
impact velocity	$\dot{y}_i$	-6.3 m/sec
energy stored	E	244 J
leg compression	Δl	0.49 m
energy dissipated	E_d	39 J
average motor power	P_{motor}	31 W

B.4 Scaling

The generalization of the design analysis is to examine how the parameters vary as a function of scale by introduce a length scaling factor λ. If we assume the mechanical dimensions are uniformly scaled, we get the following table:

Quantity	Variable	Scaling
hopper mass	M	λ^3
gravity	g	1
leg length	l	λ
leg compression	Δl	λ
maximum energy storage	E	λ^3

The maximum energy storage is assumed to be a function of the mass of the leg. Given that the force of a constant force leg spring $\mathbf{F}_0$ is related to the maximum energy storage and compression:

$$F_0 = \frac{E}{\Delta l} \sim \frac{\lambda^3}{\lambda} \tag{B.19}$$

then the leg force varies as λ^2. This is compatible with an assumption that the ground can withstand constant pressure; as the foot area increases as λ^2, the maximum ground force will also increase as λ^2.

The hopping height, however, remains constant. The energy that must be stored is as follows:

$$E \simeq Mg\Delta y \sim \lambda^3 \cdot 1 \cdot \Delta y \tag{B.20}$$

Since E and M already vary as λ^3, then Δy must be constant. So if the hopping height is to increase in scale with the machine, the energy storage and hence the leg mass must increase as λ^4. The ground force must then increase as λ^3, which will eventually exceed the ground strength.

Another way to see this is that each unit mass of leg spring can store enough energy to launch itself some distance vertically. The distance the entire machine can be launched depends on the ratio of leg mass to total machine mass. With a uniform scaling, this ratio is fixed and so the launch distance remains independent of scale.

Appendix C

3D Hopper Design

This appendix presents some preliminary design notes for a "3D" unconstrained hopper. Although this is not within the thesis scope, this may be helpful for future work.

C.1 Design Issues

The 3D Bow Leg Hopper operates on the same design principles as the 2D Hopper: it uses an efficient spring for a leg that is freely pivoted at the hip and positioned during flight by control strings, and has the center of mass balanced below the hip. The chief difference is that the leg must freely pivot in two DOF, which substantially complicates the mechanical design.

The big questions seem to be guaranteeing sufficient leg angular freedom, placing the control string drive with similar freedom, routing the Bow String through the hip without interference, and keeping the mass distribution low enough for stability. I anticipate that the existing thrust mechanisms should work fine since they in no way depend on the freedom of the leg. The chief difficulty is that the mass of the thrust drive is well above the hip and will need to be balanced by weight elsewhere.

Another issue is general robustness. The 3D hopper is much more likely to hit the ground during falls and the leg to hit the joint limits. If the machine comes to rest on the foot and an opposite body point the leg will undergo significant lateral force at the impingement point.

The limited experience with the air-table planar prototypes suggests that the attitude stability may be marginal without the damping of the boom pitch bearing. This is expected to make passive attitude stability more difficult and active hip positioning may be required. Alternatively, another stabilizing mechanism such as a gyroscope or aerodynamic control surface may be needed. However, the free hip of the Bow Leg should still keep the disturbances low so control torques are likely to be small.

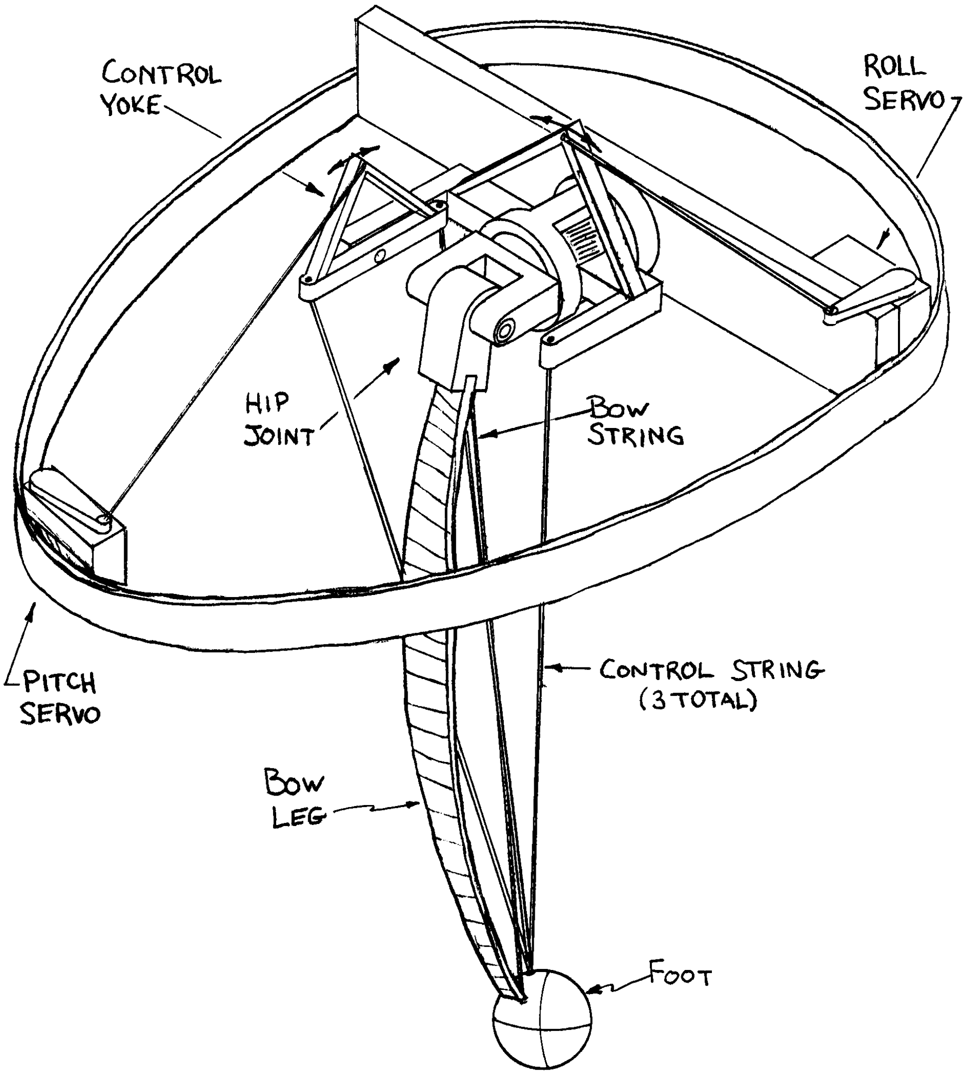


Figure C.1: Preliminary sketch of a gravity-powered 3D Bow Leg Hopper. Not shown are batteries and the R/C radio receiver. The hip assembly combines two universal joints: the interior one is a freely pivoting hip that allows the leg to move freely during stance, and the exterior one is a control yoke gimbal driven by two hobby servos, which is used to position the leg during flight. The Bow String holds the Bow Leg in compression.

A good starting point will be to build a remote controlled prototype without thrust. This tests the fundamental leg and hip design but keeps the mass low and avoids the difficult sensing and communication problems. A preliminary sketch of such a machine is shown in Figure C.1.

C.2 Leg Freedom

For control purposes the leg only needs to be placed within the friction cone around the terrain surface normal. However, the oscillation of the passive attitude stability may mean that the body is not level at impact, and so a much wider cone of motion is desirable. Although the control treats the foot motion as a point on a sphere centered on the hip (i.e., 2 DOF), the leg is asymmetric due to bending during compression. The ideal hip joint would be a frictionless ball joint with no freedom about the leg axis, which would avoid singularities but keep the leg oriented properly for bending clearance.

A reasonable compromise is to use a universal joint (gimbal). It will have low friction and may be constructed of standard roller bearings. The chief disadvantage are the singularities which prevent free body pivoting if the attitude moves far from the level position. A secondary disadvantage is the preferred direction: the first rotation axis will have somewhat lower rotational inertia and is better for the primary direction travel. Sideways hopping will involve accelerating more bearing inertia and will cause greater attitude disturbances.

During stance, the leg flex causes significant hip rotation around the axis perpendicular to the plane of the leg. This significantly increases the "free space" requirement; for a design goal of ± 60 degrees leg travel, a full compression adds about 80 more degrees of travel: the clearance interval is then $[-60, 140]$ degrees at the hip (200 degrees total clearance). For this reason the preliminary design uses a cantilever for the second gimbal axis in order to leave the volume behind the leg unconstrained.

The leg itself can probably be substantially similar to the 2 DOF leg. The forces are still axial, although the foot will need to be able to handle contact at roll and pitch angles. The gimbal design illustrated moves the hip bearing off the leg to a pair around the shaft; this allows string clearance in the center and handles higher side loads, but does require a redesign of the leg hip shaft.

The bearings selected are low profile roller bearings. There are two motivations: ensuring string clearance and keeping the mass and radius of the parts low. As described below, the string needs a wide clearance cone, so a low profile hip cage maximizes freedom.

The body frame needs attention paid to the impingement at different leg orientations: the hip gimbal frame limits the leg forward pitch, and the body side frame limits leg roll and backward pitch. The contact points should be reasonably far from the hip center since they may act as the fulcrum during a fall if the leg is pinned between the ground and body—a short lever arm could cause very high impingement forces at the contact point. Unfortunately, nothing prevents the leg from striking the yoke in certain roll configurations, since the extremal limits of the travel freedoms overlap (e.g., a full left leg roll will strike the string yoke gimbal positioned for a full right roll).

C.3 Yoke Freedom

The leg positioning yoke operates in similar fashion to the planar hopper, moving the leg control strings during flight and becoming decoupled during stance. Like the leg, it also needs 2 DOF, which implies the use of three or more control strings. Three strings avoids redundancy, i.e., the possibility of multiple position equilibria.

It is probably best if the rotational center of the yoke is coincident with that of the hip so the control string lengths do not change as the yoke rotates. This leads to the illustrated design with a gimbal concentric with the leg gimbal, both centered on the hip. Ideally the angular freedom of the yoke would be as large as the leg positioning cone. There are no direct problems with the additional angles of leg flex so the freedom is a cone (unlike the hip clearance for the leg), but the yoke does need to stay clear of the volume behind the leg.

The trickiest issue with the yoke is placing the two drive motors. The simplest design would have a direct drive motor mounted on the cage, but this greatly increases the cage inertia and swept volume. Another solution is to place both motors on the body and use a linkage, but the parallel drive means that the pitch freedom is coupled to the roll freedom. It is possible to use a differential linkage to decouple the axes, but it adds a lot of mechanical complexity and backlash. A better possibility is to mount the pitch servo along the roll axis near the rear of the body; the linkage to the pitch yoke may need to be long, but the axes will be decoupled.

Another issue is that the yoke has a larger radius than the leg gimbal and will sweep a larger volume that also must be unoccupied by the body. A solution to this is to use as small a yoke radius as possible and increase the control string tension to compensate.

C.4 String Placement

The Bow String needs to go through the hip so that the string tension does not exert torques on the leg. In practice, it is impossible to make the center of force remain exactly on the hip axis as the leg moves about, so some parasitic torques are inevitable. Since the string moves with the leg, the bottom of the hip shaft needs a fan-shaped opening to accommodate the range of angles associated with the leg compression. (It is not clear exactly what this range is, since the string is only under tension during a part of this compression.) The string leaves the hip on top to connect to the thrust mechanism fixed on the body; since the string orientation doesn't change with respect to the body, ensuring body clearance is trivial, but the string clearance on the top of the hip needs to accommodate the full motion of the leg. This means a wide cone of clearance is needed in the top of the hip shaft and the cage of the hip gimbal. This cone also needs to accommodate the leg flex, so the hip shaft clearance is the convolution of the leg sweep cone volume with a leg flex fan volume (of uncertain extent).

In this design, the control strings again form a separate control yoke that meets near the foot, passes through a pulley in the foot, and is tensioned with an elastic element. Three strings avoids redundancy, but every arrangement with a finite number of strings has anisotropic lateral stiffness at foot. In the design illustrated the strings are arranged to avoid interference with the leg flex at the back and to keep the joints oriented along orthogonal axes parallel with the machine axes for independent control of roll and pitch leg orientation. The stiffness variation may be ameliorated by choosing equal lever arms for the left pair and the right string so the side to side stiffness is equal. If pressed forward, however, the foot will tend to also curve to the right, since the left rear string will go slack and the foot will pivot perpendicular to the line between the other two string connections. In addition, the tension in the right string will be twice that of each of the left pair.

The chief disadvantage of this design that the strings become tight before full leg extension, so there is always the possibility of a torque impulse at takeoff. It is also awkward to have the pulley and elastic element near the foot. There are other possibilities. One is to return to a positioner design that pushes laterally on the Bow String, with the attendant positioning error. Another is to use three winches fixed to the body and use active control to manage the redundancy. This eliminates the gimbal but requires fast precise control of three motors. However, this could be programmed to pay out slack during stance to eliminate takeoff torques from the control strings. With especially fast and powerful motors the three strings could also be used to tension the leg, but all the advantages of mechanical thrust triggering would be lost.

Appendix D

Programmable Mechanisms

The Bow Leg is a novel design for a leg that is lightweight and efficient with high energy storage. This thesis explores its use in a one-legged hopping robot, but it could could also be used for multilegged locomotion or as an actuator for manipulation. It may also be seen as part of a larger class of “programmable mechanisms” that use small configuration actuators to guide the behavior of a mechanism that passively carries the task loads. This appendix is intended simply to suggest some of the ideas that might be generalized to other mechanisms. Some of the principles are as follows:

1. Use natural oscillators to support load forces conservatively.

2. Use mechanical energy storage as a buffer between low power energy sources (electric motors) and high power loads.

3. Keep actuator forces orthogonal to dynamic forces, either in space or in time. Dynamic forces may be supported using bearings instead of actuators.

4. Eliminate disturbance forces by design.

5. Use mechanical feedback wherever possible. This might be a discrete event such as triggering a spring release, or a continuous process like a flyball governor. In the hopper, mass placement is sufficient to create mechanical feedback.

6. Incorporate a cycle into the design that reduces the control and sensing bandwidth.

The Bow Leg Hopper displaces actuation in time (to flight) by using the leg spring to buffer thrust energy and the initial leg position to passively generate “control” forces. The low actuator force is a double win: not only is the actuator power low, but the forces are low. This reduces the size and weight of the actuators, body structure, and the power supply. High stresses is confined to a limited piece of the hopper structure.

There are certainly other solutions that offer similar properties. The pitch could be controlled by a spring mechanism that exerts torques directly on the leg, or by a non-contact torque source such as a momentum wheel or aerodynamic vanes. The Bow Leg energy mechanism is completely passive during stance, but low power electrical actuators could be used to trigger the mechanical release.

The Bow Leg mechanism itself is more general that a one or multilegged hopper. As a mechanically configured impulsive actuator using a small power source, it could be used for low duty cycle impulsive manipulation operations like batting an object off a conveyor belt. Here are several ideas for applications that could use similar ideas:

- Satellite rotation. The report [Pit90] describes a system for rapidly reorienting a satellite by accelerating a flywheel using energy stored in a spring, then recovering the energy by clutching to another spring.

- Active automotive suspension. A conservative mechanism could conceivably use the energy of one bump to drive the suspension down for the next pothole (thanks to Professor Bill Messner for this suggestion).

- Conveyor belt manipulator. A Bow Leg could be used as a finger to knock an object off a conveyor belt. It would be tensioned and positioned as the object approached, and then at contact would mechanically trigger and deliver a controlled impulse. Self-timing could dramatically reduce the sensing required.

- Walking robots. The paper [Jam85] and patent [Jam] describe a gyroscope-stabilized two-legged walking device that uses no actuators. The gyro axis is nominally parallel to the ground, oriented across the body. In this configuration the torque exerted by the body weight acting around the stance foot causes a precession that rotates the body, both slowing the body fall and causing a forward stride. The gyroscope is also used as a flywheel to power the alternating leg lift using a linkage.

Bibliography

[AB97] M. Ahmadi and M. Buehler. Stable control of a simulated one-legged running robot with hip and leg compliance. *IEEE Transactions on Robotics and Automation*, 13(1), 1997.

[Ale77] R. McN. Alexander. Mechanics and scaling of terrestrial locomotion. In T.J. Pedley, editor, *Scale Effects in Animal Locomotion*, chapter 6, pages 93–111. Academic Press, London, 1977.

[Ale90] R. McN. Alexander. Three uses for springs in legged locomotion. *International Journal of Robotics Research*, 9(2), 1990.

[AV75] R. McN. Alexander and Alexandra Vernon. The mechanics of hopping by kangaroos (macropodidae). *Journal of Zoology*, 177:265–303, 1975.

[BD89] M. Boddy and T. Dean. Solving time-dependent planning problems. In *IJCAI Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, volume 2, pages 979–84, Detroit, MI, 1989.

[BD96] M.D. Berkemeier and K.V. Desai. Design of a robot leg with elastic energy storage, comparison to biology, and preliminary experimental results. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 213–8, 1996.

[BRK99] R.R. Burridge, A.A. Rizzi, and D.E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research*, 18(6):534–55, June 1999.

[BZ98] H.B. Brown and G.J. Zeglin. The Bow Leg Hopping Robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1998.

[CHT77] Giovanni A. Cavagna, Norman C. Heglund, and C. Richard Taylor. Mechanical work in terrestrial locomotion: two basic mechanisms for minimizing energy expenditure. *Am. J. Physiol.*, 233(5):R243–R261, 1977.

[DB88] T. Dean and M. Boddy. An analysis of time-dependent planning. In *AAAI Seventh National Conference on Artificial Intelligence*, volume 1, pages 49–54, Saint Paul, MN, 1988.

[GAB93] P. Gregorio, M. Ahmadi, and M. Buehler. Experiments with an electrically actuated planar hopping robot. In *International Symposium on Experimental Robotics III*, Kyoto, Japan, 1993.

[GAB97] P. Gregorio, M. Ahmadi, and M. Buehler. Design, control, and energetics of an electrically actuated legged robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 27B(4):626–634, 1997.

[GvK50] G. Gabrielli and T.H. von Karman. What price speed? *Mechanical Engineering*, 72(10):775–781, 1950.

[Hir84] Shigeo Hirose. A study of design and control of a quadruped walking vehicle. *International Journal of Robotics Research*, 3(2):113–133, 1984.

[Hod88] J. Hodgins. Legged robots on rough terrain: experiments in adjusting step length. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 824–6, Philadelphia, PA, 1988.

[Hod89] Jessica Kate Hodgins. *Legged robots on rough terrain: experiments in adjusting step length.* PhD thesis, Carnegie Mellon University, November 1989. CMU-CS-89-151.

[HR91] J.K. Hodgins and M.N. Raibert. Adjusting step length for rough terrain locomotion. *IEEE Transactions on Robotics and Automation*, 7(3):289–98, June 1991.

[HT85] M. Hubbard and J.C. Trinkle. Clearing maximum height with constrained kinetic energy. *Journal of Applied Mechanics*, 52(1):702–4, March 1985.

[Hua97] Wesley H. Huang. *Impulsive Manipulation*. PhD thesis, Carnegie Mellon University, August 1997. CMU-RI-TR-97-29.

[HvdP96] Pedro S. Huang and Michiel van de Panne. A planning algorithm for dynamic motions. In *Computer Animation and Simulation '96 – Proceedings of the 7th Eurographics Workshop on Simulation and Animation*, pages 169–182, Poitiers, France, 1996. Springer Verlag.

[Jam] John W. Jameson. Gyroscopic walking toy. U.S. Patent 4,365,437. Filed Apr. 15, 1981, Granted Dec. 28, 1982. 7 claims.

[Jam85] John W. Jameson. The walking gyro. *Robotics Age*, pages 7–10, January 1985.

[Lap92] V.V. Lapshin. Vertical and horizontal motion control of a one-legged hopping machine. *International Journal of Robotics Research*, 11(5):491–8, 1992.

[LPK93] A. Lebaudy, J. Prosser, and M. Kam. Control algorithms for a vertically-constrained one-legged hopping machine. In *Proceedings of 32nd IEEE Conference on Decision and Control*, volume 3, pages 2688–93, 1993.

[Mat79] K. Matsuoka. A model of repetitive hopping movements in man. In *Proceedings of Fifth World Congress on Theory of Machines and Mechanisms*. International Federation for Information Processing, 1979.

[Mat80] K. Matsuoka. A mechanical model of repetitive hopping movements. *Biomechanisms*, 5:251–8, 1980.

[McG89] Tad McGeer. Powered flight, child's play, silly wheels and walking machines. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1592–7, 1989.

[McG90] T. McGeer. Passive dynamic walking (two-legged machines). *International Journal of Robotics Research*, 9(2):62–82, 1990.

[McM84] T.A. McMahon. Mechanics of locomotion. *International Journal of Robotics Research*, 3(2):4–28, Summer 1984.

[McM85] T.A. McMahon. The role of compliance in mammalian running gaits. *J. Exp. Biol*, 115:263–282, 1985.

[Nou97] Illah Reza Nourbakhsh. *Interleaving Planning and Execution for Autonomous Robots*. Kluwer Academic Publishers, Boston, 1997.

[Pap91] 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 1553–60, 1991.

[PDP97] J. Pratt, P. Dilworth, and G. Pratt. Virtual model control of a bipedal walking robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, Albuquerque, NM, 1997.

[Pit90] Lee C. Pittenger. The use of mechanical storage and recovery of energy to effect repeated rapid rotations of spacecraft. Technical Report UCRL-JC-105874, Lawrence Livermore National Laboratory, 1990.

[PW95] G. Pratt and M. Williamson. Series elastic actuators. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Pittsburgh, PA, 1995.

[Rai86] M.H. Raibert. *Legged Robots That Balance*. MIT Press, Cambridge, MA, 1986.

[RBC+85] M.H. Raibert, H.B. Brown, M. Chepponis, J. Hodgins, J. Koechling, J. Miller, K.N. Murphy, S.S. Murthy, and A. Stentz. Dynamically stable legged locomotion—fourth annual report. Technical Report CMU-LL-4-1985, Carnegie-Mellon University, 1985.

[Rin96] Robert Ringrose. *Self-stabilizing running*. PhD thesis, Massachusetts Institute of Technology, October 1996.

[Rin97] Robert Ringrose. Self-stabilizing running. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 487–93, 1997.

[RMWT97] T.J. Roberts, R.L. Marsh, P.G. Weyand, and C.R. Taylor. Muscular force in running turkeys: The economy of minimizing work. *Science*, 275, Feb. 21 1997.

[Rui91] Andrew Ruina. The Human Power, Biomechanics, and Robotics Laboratory at Cornell University. MIT Leg Lab talk, 1991.

[SA93] S. Schaal and C.G. Atkeson. Open loop stable control strategies for robot juggling. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 913–18, Atlanta, Georgia, 1993.

[SK95] W.J. Schwind and D.E. Koditschek. Control of forward velocity for a simplified planar hopping robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 691–6, 1995.

[SK97] W.J. Schwind and D.E. Koditschek. Characterization of monoped equilibrium gaits. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1997.

[Tuc75] A. Tucker. The energetic cost of moving about. *American Scientist*, 63:413–19, July-August 1975.

[Wal] Henry W. Wallace. Vehicle. ("..a tank which is propelled by means of an operating leg."). US Patent 2,371,368. Filed October 16, 1942, Granted Mar. 13, 1945. 11 claims.

[WYL86] William H. Warren, Jr., David S. Young, and David N. Lee. Visual control of step length during running over irregular terrain. *Journal of Experimental Psychology: Human Perception and Performance*, 12(3):259–266, 1986.

[ZB98] G.J. Zeglin and H.B. Brown. Control of a Bow Leg Hopping Robot. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1998.

[Zeg91] G.J. Zeglin. Uniroo: A one legged dynamic hopping robot. B.S. thesis, MIT Dept. of Mechanical Engineering, 1991.

[Zil96] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73–83, 1996.

[ZR93] S. Zilberstein and S. J. Russell. Anytime sensing, planning and action: A practical model for robot control. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1402–07, Chambery, France, 1993.