Uniroo: A One Legged Dynamic Hopping Robot

by

Garth J. Zeglin

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Bachelor of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1991

Author ....
Department of Mechanical Engineering
May 10, 1991

Certified by ....
Marc H. Raibert
Professor
Thesis Supervisor

Accepted by ....
Prof. Peter Griffith
Chairman, Department Committee

Uniroo: A One Legged Dynamic Hopping Robot

by

Garth J. Zeglin

Submitted to the Department of Mechanical Engineering
on May 10, 1991, in partial fulfillment of the
requirements for the degree of
Bachelor of Science

Abstract

Legged animals exist in a variety of configurations, which produce a variety of modes of locomotion. We have created a machine that mimics kangaroo locomotion in order to gain insight into the nature of hopping.

The robot is kinematically similar to a kangaroo, but has one leg and is constrained to two dimensions. The machine has performed numerous runs of approximately ten hops, and two runs of approximately forty hops. The experimental results indicate that a virtual leg concept can be applied to control a three-link leg with revolute joints.

This paper includes discussion of the experimental results, an introduction to virtual leg ideas, and descriptions of the mechanical, electronic, and software details of the robot.

Thesis Supervisor: Marc H. Raibert
Title: Professor

Acknowledgments

I would like to thank Dave Barrett for his invaluable assistance at every stage of the project. I would like to thank Marc Raibert for his vision and guidance that made this project possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human beings and animals both have remarkable abilities to walk and run over a wide variety of terrain. It is desirable to understand these abilities from several perspectives. Knowing how humans and animals work is an interesting subject in its own right, and will advance our knowledge of anatomy and physiology by making the mechanisms behind locomotion more clear. This understanding could lead to practical benefits in medicine such as advances in prosthetics technology. Understanding the nature of legged locomotion could also lead to the development of machines capable of traversing a much wider variety of terrain than is possible with wheeled and tracked vehicles.

In the Leg Lab at MIT, we are concerned with examining the mechanisms behind the processes of legged locomotion. Our principle of investigation is a synthetic approach: by trying to create machines that have these behaviors, we will discover whether we have taken into account all of the factors that are fundamental to legged locomotion. In practice, the machines that we build are much simpler than real animals, since we are trying to address the fundamentals of locomotion without being distracted by details specific to animals.

To date, we have constructed a variety of robots that can hop and run. We have been concerned with dynamic legged locomotion, e.g. hopping and running, rather than statically stable locomotion. Dynamic locomotion means that the machine is in a constantly changing dynamic balance, as opposed to statically stable machines, where at every point in time, the machine is supported on its legs and could freeze its motion and remain standing.

As part of the ongoing research effort into legged locomotion, we have decided to move closer to the problems of animal locomotion by designing a machine kinematically similar to an kangaroo. The result is the Uniroo, which is the subject of this paper.

In the text of this paper, I will write from the perspective of this ongoing research effort. For a statement of my personal involvement with the project, see Appendix A.

1.1 The Uniroo

The Uniroo is an attempt to more closely simulate the physical structure of an animal. The leg is kinematically similar to a real kangaroo, and is proportional in size to a kangaroo of similar mass. Due to the limitations of human-built machines, which can not sport the sort of low-mass, high strength, high-degree of freedom actuation that animals have, the machine is not completely dynamically similar, and the structure is simplified. We decided to constrain the machine to two dimensions and only have one leg. Previous machines have demonstrated that the side-to-side motion is relatively decoupled from the forward motion, so we believe this is a reasonable simplification. Since a kangaroo uses its two legs in tandem when hopping, we decided to use a single leg and avoid problems of synchrony. At the time of this writing, the robot has had several successful trials of over 40 hops, and we believe that the performance of machine can still be improved.

1.2 Research Philosophy

When a human learns to walk, there is a long process of trial and error. The baby tries a behavior, and the results modify further attempts. In developing our robots, a similar procedure takes place. A behavior is planned, attempts are made, and we use our observations to modify the software to achieve the behavior we desire. The human experimenters are the learning loop for the robot. To enable our careful observation, we emphasize recording detailed data.

Since we are interested in studying the dynamics of locomotion, we have tried to strip away extraneous details. Our machines are not energetically or computationally autonomous, and have an umbilical to supply power and communications. We do not incorporate external sensing (except a foot contact switch); that information is provided by an operator, who can remotely steer the machines. For several of the machines, we have chosen to simplify the problem to two dimensions. This constraint captures many features of the problem while allowing easier development.

What we do is build machines that can dynamically locomote, automatically maintaining balance. On top of this primary behavior, we add control of forward speed, hopping height, foot placement, and transition between various gaits. Often, we test machine concepts by developing a simulation incorporating a dynamic model and control software. If the idea works in simulation, we design and construct a physical machine. Then begins a process of testing and refining the control algorithms until the machine can perform basic locomotion. After that experiments such as foot placement control, hopping height control, or gymnastics can be performed.

Chapter 2

Overview

In this section I will provide an overview of the Uniroo behavior, the virtual leg idea, and the physical implementation of the robot. For further details, please refer to chapters 3 through 5. An illustration of the machine is provided in Figure 2-1, and the various parts are schematically labelled in Figure 2-2. Figure 2-3 provides a diagram of the planarizing mechanism that constrains the machine to two dimensions. For further context, see the photographs in Appendix B.

2.1 Behavior

In this section, I will attempt to provide a description of what the machine looks like as it is operating, and a discussion of what hopping and virtual legs.

2.1.1 The Uniroo

Figure 2-4 shows a series of cartoons representing the Uniroo as it goes through one hop. In operation, the robot is suspended about 10 cm above the ground, and is dropped onto its toe. It slows and then springs up again into the air, then falls again, each hop taking about 0.7 seconds. During each hop, the robot moves forward, lands, sweeps its leg underneath, takes off again, and moves its leg forward during its flight. Each hop is a little different, the body pitching forward more or less, the robot moving a little faster or slower.

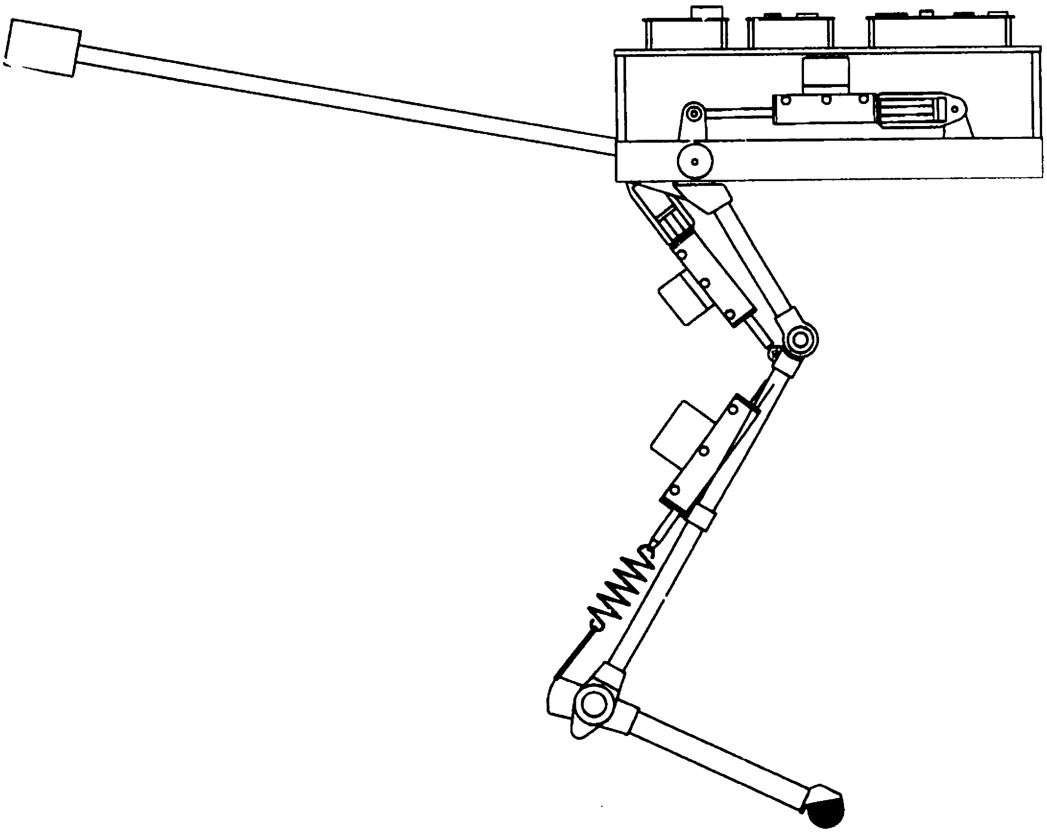


Figure 2-1: Illustration of Uniroo

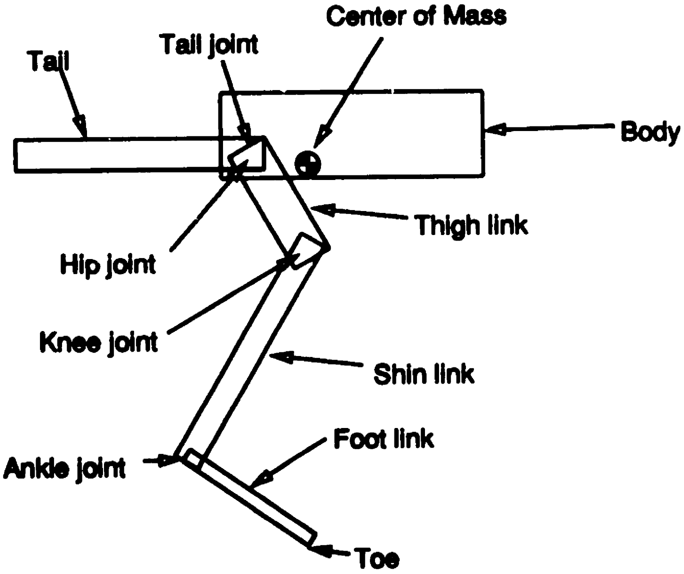
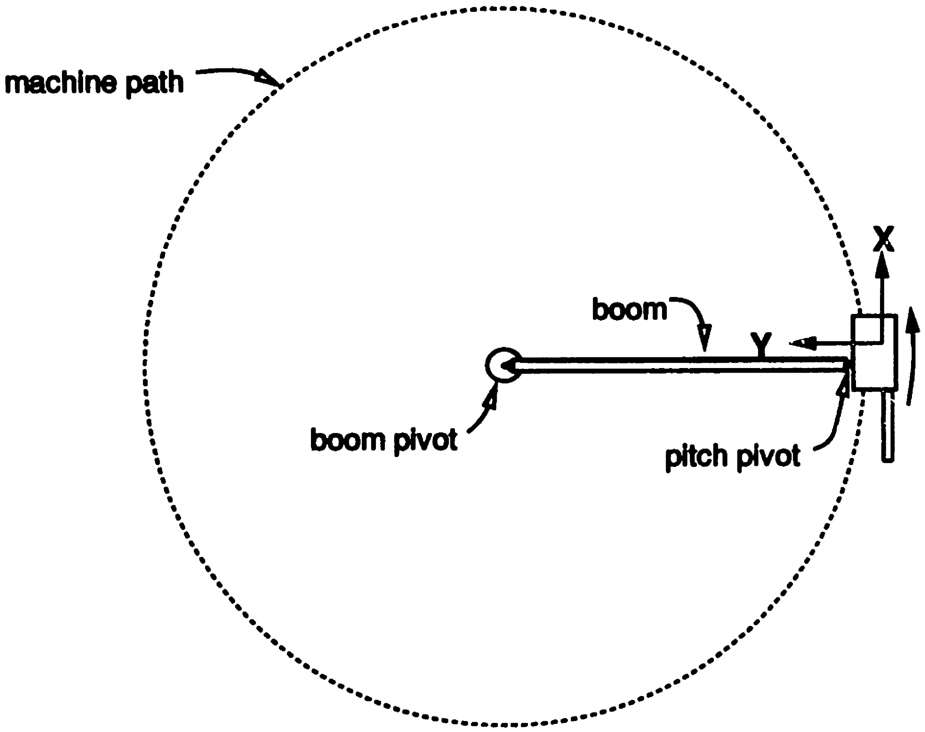
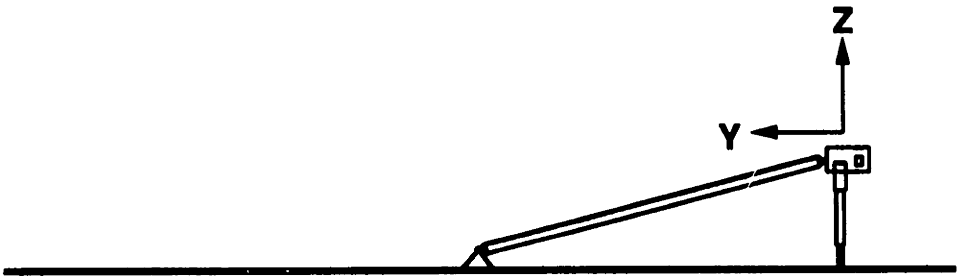


Figure 2-2: Schematic Diagram of Uniroo

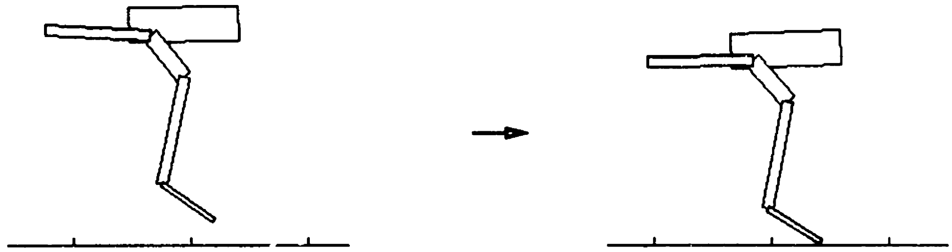


Top View



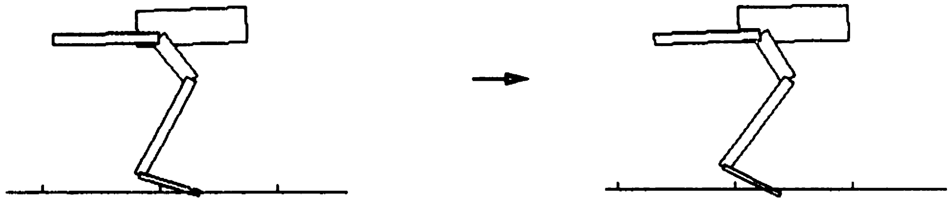
Side View

Figure 2-3: Schematic Diagram of Planarizer



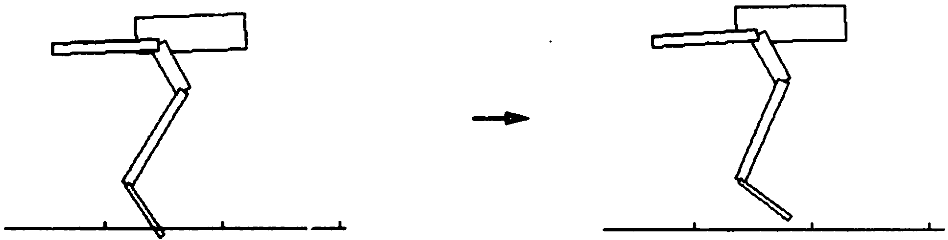
1. Flight: Descent

2. Landing



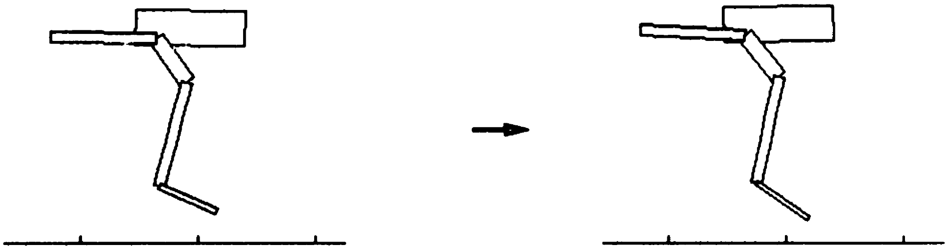
3. Compression

4. Thrust



5. Liftoff

6. Escape



7. Flight: Ascent

8. Flight: Descent

Figure 2-4: Cartoons of Machine States

Variable	Definition
q_tailly	tail joint angle
q_hipy	hip joint angle
q_kneey	knee joint angle
q_ankley	ankle joint angle
q_z	vertical distance from ground to center of mass
q_x	horizontal position
q_pitch	rotation of body around Y axis
inx	algorithm state machine index
q_fs	footswitch (1 if touching ground, else 0)
h_ankley	ankle actuator piston position
h_d_ankley	desired value for h_ankley
l3	distance ankle actuator piston is retracted for thrust

Table 2.1: Definitions of Variable Names (partial list)

The robot is mounted on the end of a boom (the "planarizer") which constrains the motion to a plane so the robot can only move forward and backward, up and down, and pitch forward and back. These three spatial degrees of freedom and the four actuated degrees of freedom (tail, hip, knee, and ankle joints) give the robot seven degrees of freedom. Forward motion takes the robot around a circle about 5 meters in diameter. The machine and boom together mass about 25 kg.

Typically, the machine hops forward at a rate on the order of one meter/second. Due to the asymmetry of the leg, the Uniroo tends to hop forward, and has difficulty hopping backward. Hopping in place is marginal. Once started, the machine hops until it falls. A rope attached to the boom through an overhead pulley allows us to lift it up ("wrangle" it) when we want to stop it hopping.

To date, the robot has made many runs of approximately 10 hops, after which it falls. The Uniroo has recently succeeded in performing two runs of over 40 hops, carrying it about 30 meters, which we believe to be the current ability of the machine. We believe time constraints alone prevented us from performing more long runs.

The data from the second of these two runs in shown in Figures 2-5 and 2-6, which shows plots of the seven degrees of freedom. To interpret the data, see Table 2.1 for the definitions of the variable names and Figure 2-7 for a description of the coordinate system.

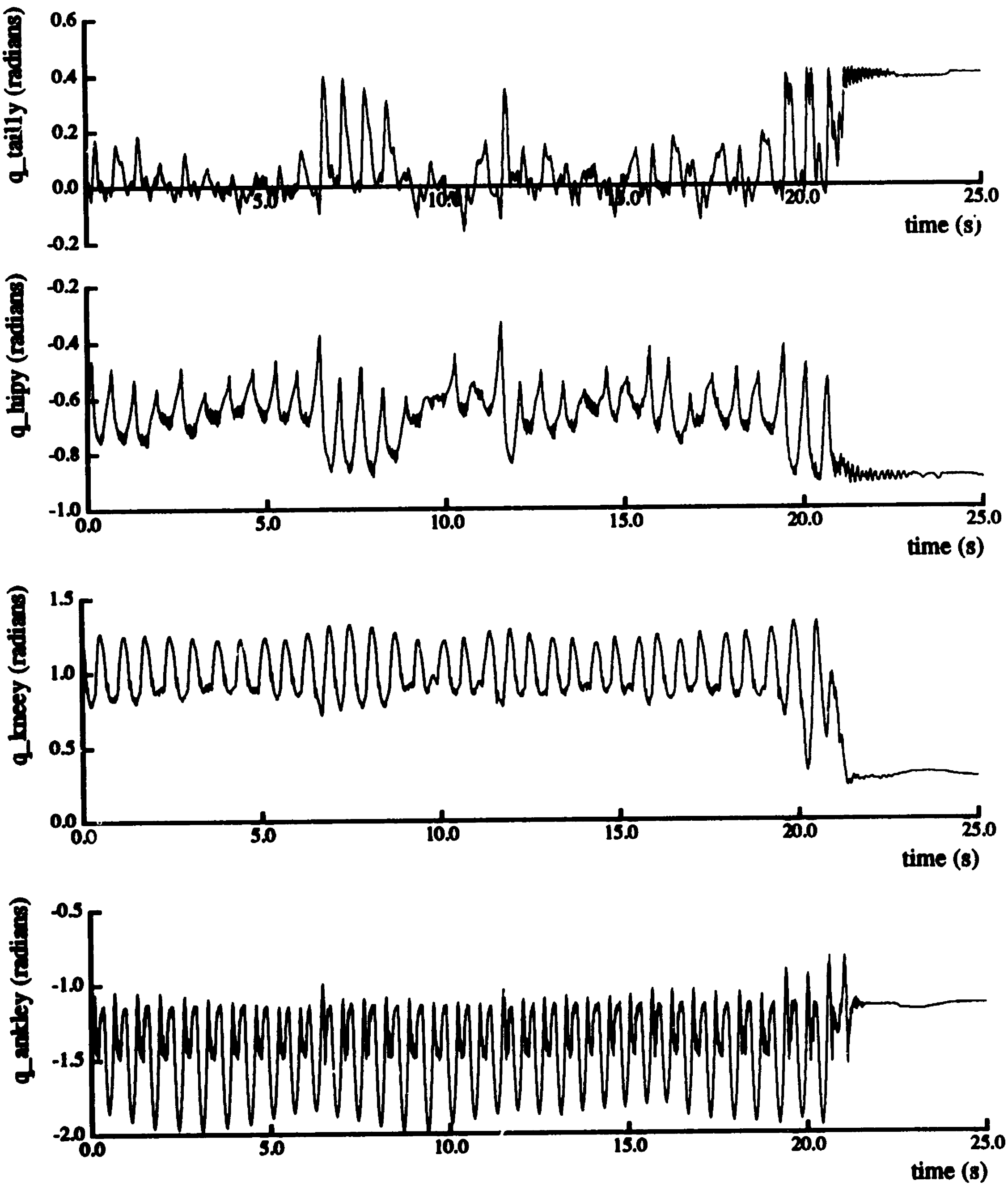


Figure 2-5: Machine Data: Run 119.43

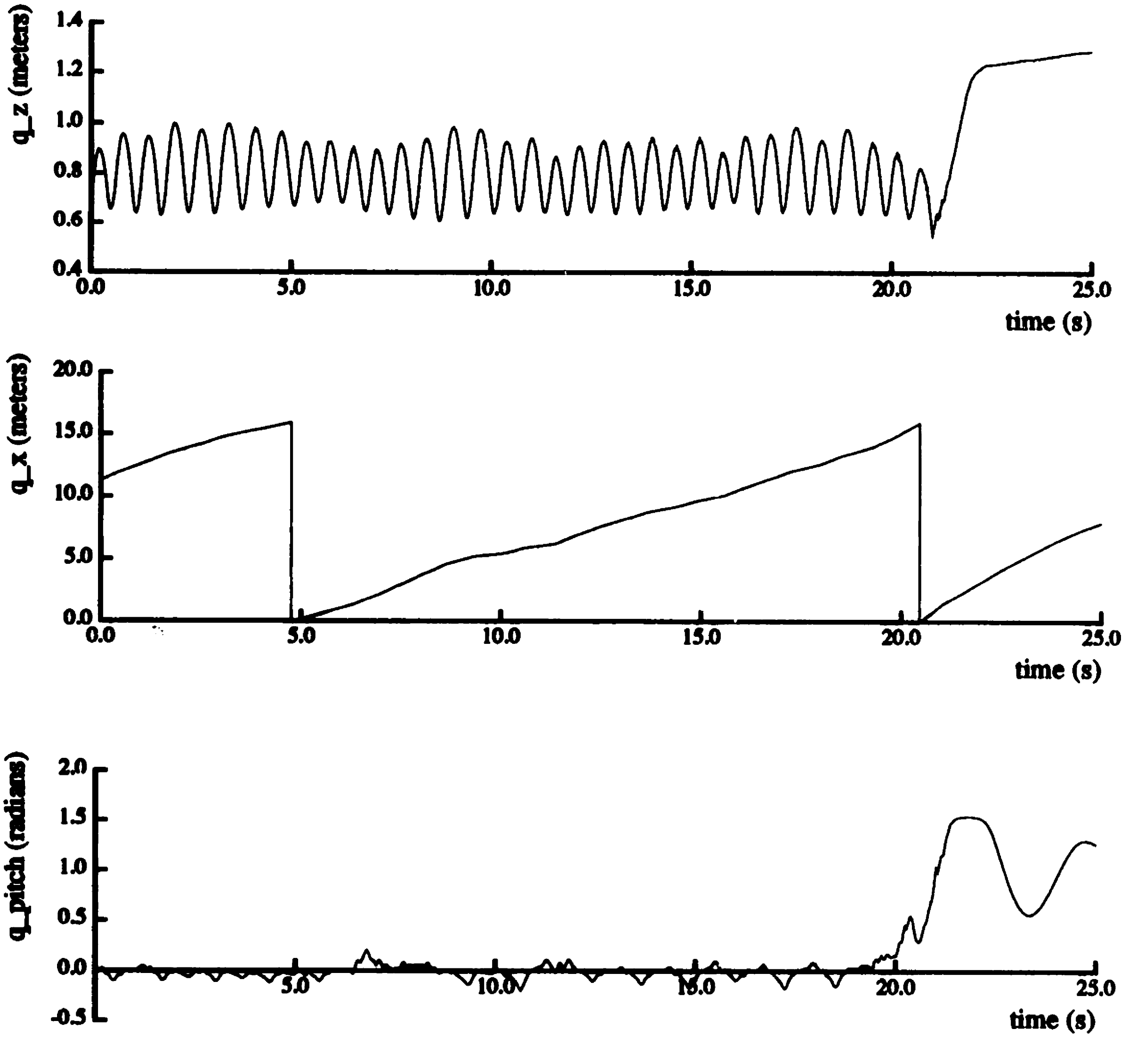
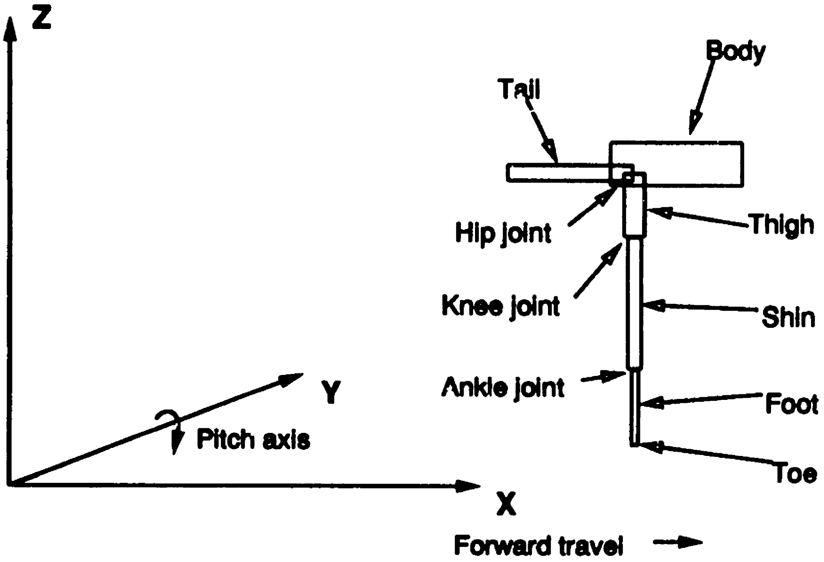


Figure 2-6: Machine Data: Run 119.43 (continued)



All rotations are measured with the right hand rule: the pitch arrow shown is a positive rotation around the Y axis. All angular measurements are made from the reference position shown on the right. The Z origin is the ground.

Figure 2-7: Coordinate System and Reference Position

2.1.2 What is hopping?

The Uniroo can hop. What does this mean? To answer this, let us consider what might define hopping.

Let us consider the simplest hopping machine of all: a bouncing ball [Mar76] (see Figure 2-8). If we drop the ball, it will fall to the ground, compress, and spring upward again. If the ball is moving forward, it will remain moving forward. This can be restated that the kinetic energy associated with its forward velocity is unchanged. If the ball is ideal, there will be no energy lost, and it will bounce to the same height from which it was dropped. The chief features of this hopping machine are that the vertical velocity reverses direction at impact, and that the energy associated with the vertical motion is transferred from the kinetic energy of falling into a potential energy of elastic deformation and back.

If we consider a slightly more complicated machine, we can get an idealized hopper as shown in Figure 2-9. It consists of a mass at one end of a telescoping leg, which is a two part leg with a sliding joint, which has a compressive spring inside and a

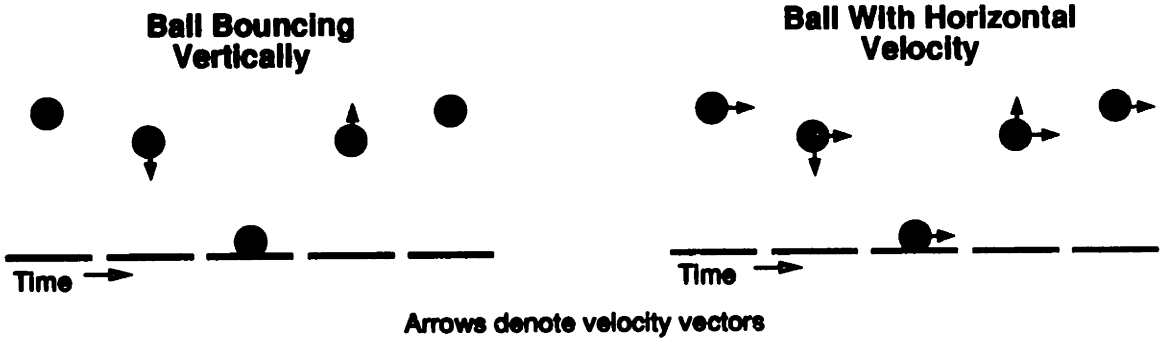


Figure 2-8: Bouncing Ball Hopper

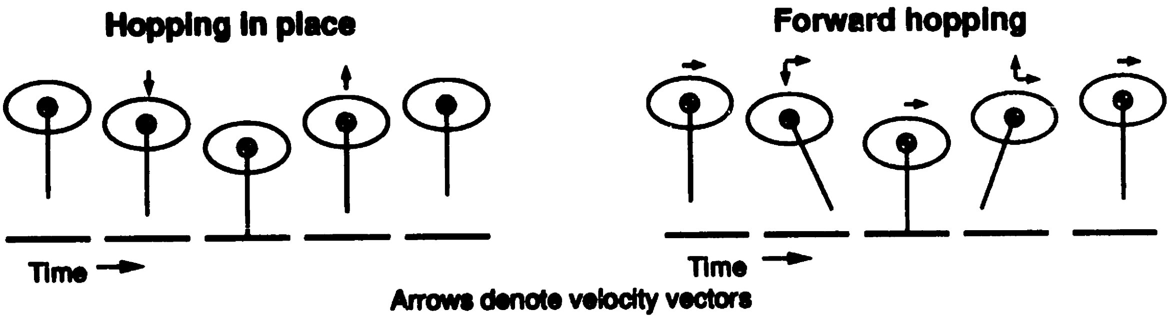


Figure 2-9: Idealized Hopper

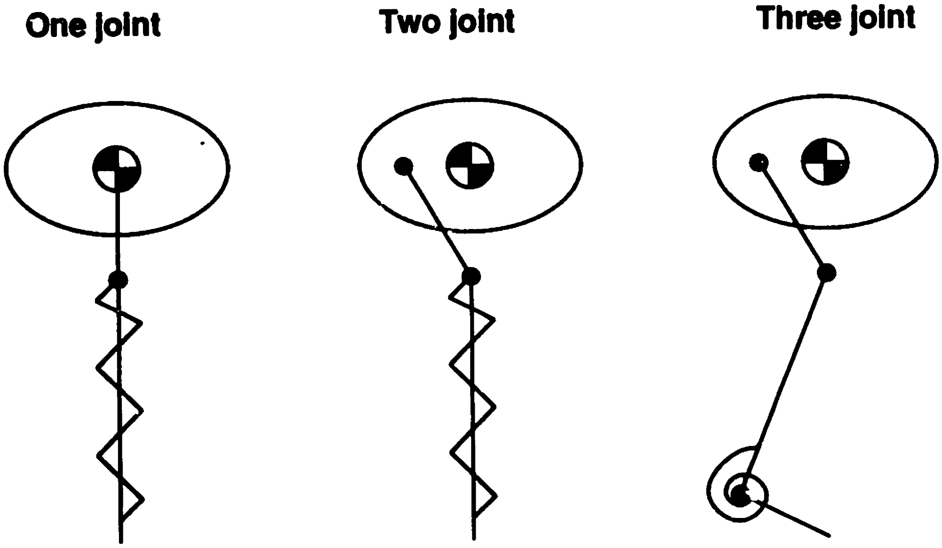


Figure 2-10: An Evolution of Legs

toe on the end. If this hopper is dropped straight down, and the toe is directly under the center of mass ("COM"), it will bounce very similarly to the ball, i.e., the kinetic energy of the body will be absorbed by the spring and released, the vertical velocity will have reversed direction, and the horizontal velocity will still be zero. This assumes a highly ideal operation, with the mass *exactly* over the toe, since the leg is an inherently unstable inverted pendulum. If the hopper is moving forward as it falls the horizontal kinetic energy can be absorbed and released similarly to the ball. For this to work, the leg must be positioned forward at landing at a position such that it will be at a symmetric position at liftoff, which will maintain the same horizontal speed. A problem that now arises now is that the leg must be moved from the liftoff position to the takeoff position, which requires applying torques that will rotate the body. Our solution to this problem is to include a tail to counterbalance the leg motion.

The hopping machines previously created at the Leg Lab are very similar to this ideal hopper. From this starting point, we can synthesize a design for a hopper with revolute joints, that is, joints that rotate instead of linearly sliding. If we first move the center of mass above the leg joint (the "knee"), we get the first machine ("One joint") shown in Figure 2-10. The telescoping portion of the leg is shown as a jagged line. This machine can hop, but if the knee is not on the line between the toe and the center of mass, the leg will buckle. If we keep the knee on this line (the toe-COM line) during forward hopping, the body will pitch forward as the knee moves along its trajectory. To solve this, we can add a joint at the body (the "hip") to give us another degree of freedom, which is the "Two joint" machine. To eliminate the linear spring, we can replace the linear spring with a torsional spring and another joint (the "ankle"), at which point we have created the Uniroo minus its tail (the "Three joint" machine).

It is possible to devise simple control laws to make the ideal hopper hop[Rai86]. One hypothesis of this experiment was that these control laws could be used for the Uniroo by considering the "virtual leg" which extends from the COM to the toe, which is implemented by the three leg links. This virtual leg is an idea that can let us program the three link leg to emulate the idealized hopper's telescoping leg. This makes all the machines in the machine evolution have conceptually similar virtual legs.

We would like to apply the idea of a virtual leg to the Uniroo leg since we already have a proven control mechanism to control the ideal hopper. The next question is, what exactly is meant by a virtual leg? It is the set of force conditions that define the ideal model, which are then imposed by the physical leg. The model is the ideal hopper, and it imposes three conditions. First, the force parallel to the leg F_p acts directly on the center of mass without imposing any torques on the body. Second, the spring force is in the F_p direction and is proportional to the leg length. Third, the force orthogonal to the leg F_o at the toe produces a torque at the hip.

Figure 2-11 shows a representation of where our Uniroo virtual leg exists. It is drawn to show that the leg represents a condition of forces acting between the toe and the center of mass. It is drawn as a spring to illustrate that it has compliance, and it is drawn as being directed through the knee joint to illustrate the virtual leg configuration we have chosen.

There are many possible virtual leg configurations for the Uniroo. A simple one has the single condition that the knee remain on a line between the toe and the center

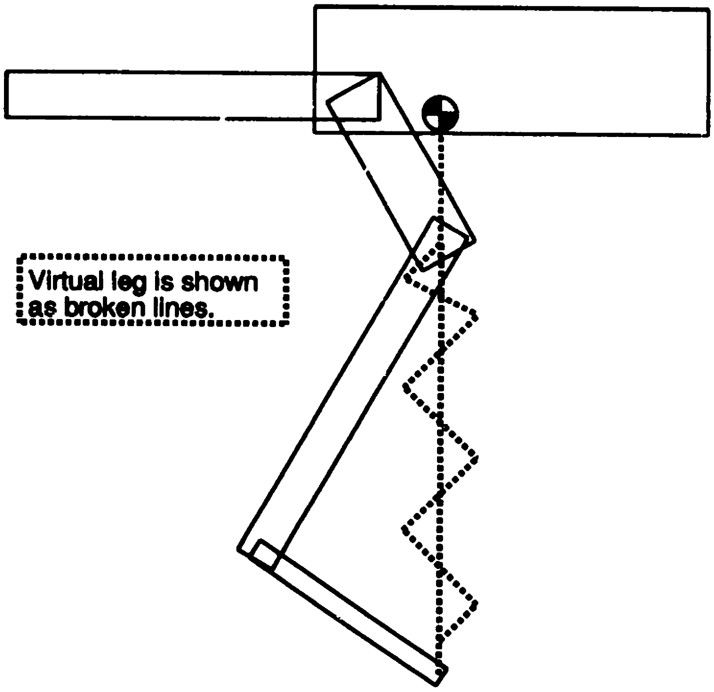


Figure 2-11: Schematic of a Virtual Leg

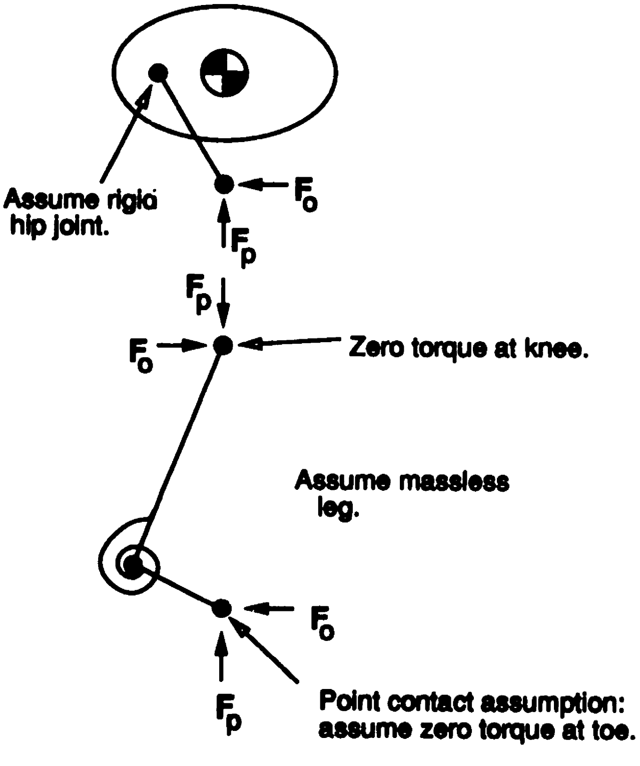


Figure 2-12: Free Body Diagram of Upper Body and Lower Leg

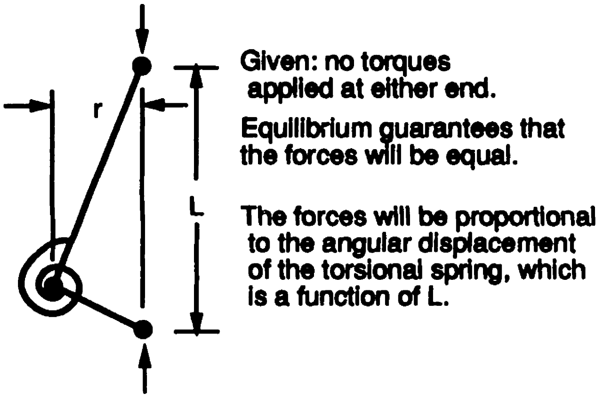


Figure 2-13: Free Body Diagram of Shin and Foot

of mass. This configuration produces zero torque at the knee. Figure 2-12 is a free body diagram to help illustrate this. Let us assume that the leg is massless, and that the point contact on the ground can produce no torque on the toe. Given that, the first virtual leg condition requires that F_p act along a line from the toe to the center of mass. That will be satisfied in any case since F_p by definition is directed along the virtual leg, but if the knee is along this line, F_p will be directed through the knee, and there will be no torque transmitted at the knee. Given that there are no knee torques, this is a simple virtual leg to maintain.

The second component of our virtual leg definition states that the spring force is in the F_p direction. Figure 2-13 illustrates that this is satisfied by this configuration. If there are no torques at the knee or toe, equilibrium considerations dictate that the forces at either end will be equal and directed from toe to knee. If there were a force in the orthogonal direction, torque equilibrium would not be satisfied. The torsional spring in the ankle produces a torque between the shin and the foot that is proportional to the angle between them, which is a function of L. F_p is equal to this torque divided by r, and hence is a function of L.

The third component of the virtual leg is not explicitly satisfied in the Uniroo implementation. That is, there is no specific mechanism to program the hip, knee, and ankle to exert the appropriate torques to couple a force F_o to the body. In fact, the torsional spring will deflect when F_o is applied, so there will be a kinematic difference between the ideal virtual leg and the Uniroo leg. In practice, the magnitude of F_o is much lower than F_p, so these effects can be treated as disturbances in the other servos.

As drawn, the idealized hopper has no energy input or control, and is a completely passive device. For this to work, the hopper would have to rotate completely around until the leg was in exactly the right position at landing, like a rimless wheel with only one spoke. Of course, this situation is a fiction. For this hopper to function, an energy source will have to be provided in the spring to restore both vertical and horizontal kinetic energy lost to friction, and a mechanism must be implemented to restore the leg from the liftoff position to the landing position during the flight phase. Furthermore, moving the leg forward during flight will apply a torque to the body that will cause the body to pitch forward, so a mechanism must be provided to adjust pitch. Finally, deviations in foot placement will cause the average horizontal velocity to vary from step to step, so a mechanism must be implemented to keep the machine from slowing to a halt or speeding up until it trips. These mechanism are all part of the control system, which will be discussed next.

2.2 Control

The cycle of hopping can be broken down into several phases, and these are reflected in the control algorithm. The cycle is divided into a flight phase, in which the robot is in the air; a landing phase, in the which the toe makes contact; a compression phase, during which the vertical energy is stored in the ankle spring; a thrust phase, in which the robot starts to move upwards again, the vertical energy is recovered from the spring, and additional energy is added; a takeoff phase, in which the toe leaves the ground; and the ascending and descending flight phases. If hopping with horizontal velocity, the horizontal energy also varies during the course of the cycle: energy is stored in the spring during compression and recovered during thrust, though the horizontal kinetic energy doesn't go to zero unless the machine is hopping in place.

One philosophy behind the control design was to try and decouple the control axes as much as possible. It seems likely that in a real animal, each controlled degree of freedom and each desired parameter is affected by many muscles. However, in order for us to be able to understand the behavior of the machine, we decided to control each desired effect (such as keeping a constant pitch) with as few degrees of freedom as possible. This architecture seems artificial considering the highly coupled systems of animals, but this separation is necessary since the humans are the learning elements in the loop, observing the behavior and correcting parameters. Without the decoupling, it would be much harder to determine the effects of parameters and see how to influence the behavior. Future work could involve a more complex control design in which a different set of coordinates is chosen to map onto the machine coordinates.

Similarly, the separation into discrete phases is a construct that makes development easier. Animals do not step between discrete states at trigger points, with the resulting step changes in desired values. Future work could also involve a more fluid description of the motion, possibly avoiding state machines or using one more based on continuous functions.

The different states and the definitions of the transitions between them are shown in Figure 2-14, and a description of the actions the machine takes in each state is in Table 2.2. The hopping cycle operates as follows: when the toe touches the ground, the machine transitions to the stance state, which is divided into a compression and a thrust phase. During both phases, the robot uses the knee to apply torques to the ground to level the pitch of the body, and the hip is used to keep the knee to a line between the toe and the center of mass. During the compression phase, the springs in the ankle are stretched, storing the energy of falling, and during the thrust phase, the ankle actuator pulls up on the springs, putting energy into the springs, which push the toe against the ground to launch the robot into flight. When the robot detects that ground contact has been lost, it enters the flight phase, in which it repositions the hip, knee, and toe for landing. Speed control is taken care of by the foot placement algorithm, which decides where to reposition the toe for the next landing. If the toe is placed farther out than nominal, more of the horizontal energy is stored (and gets transferred to vertical kinetic energy), decreasing forward velocity.

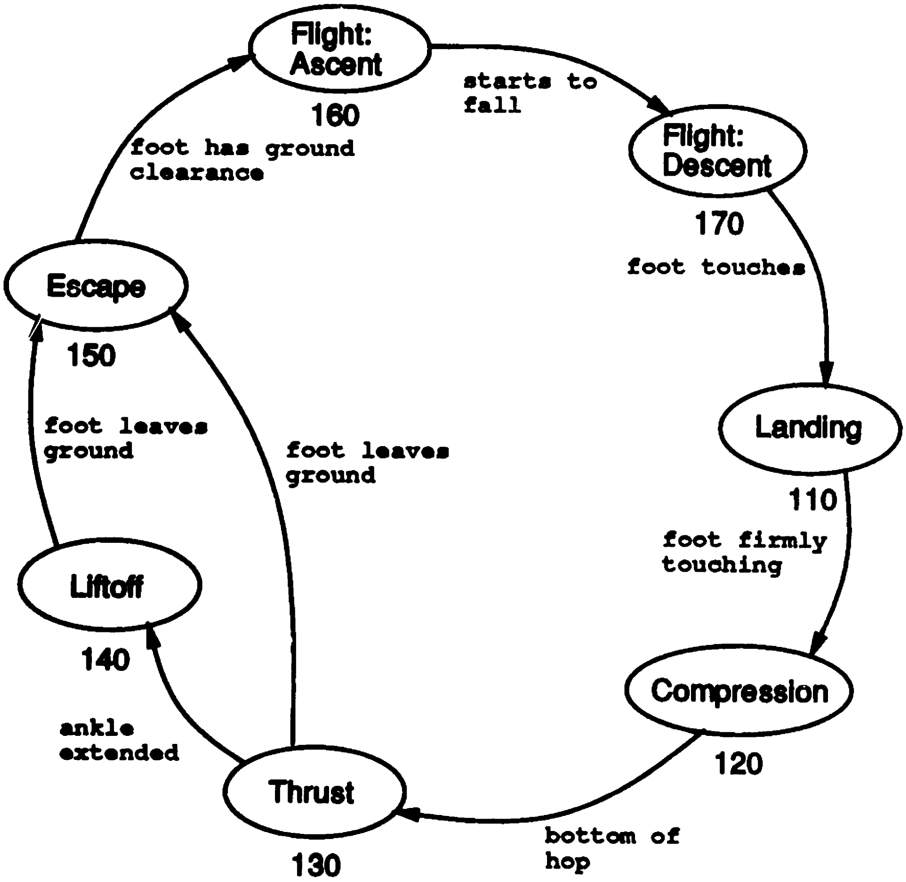


Figure 2-14: Hopping Algorithm States

State	Action
Flight: Descent	tail mirrors hip position and zeros pitch positions hip, knee, and ankle for landing
Landing	tail mirrors hip position hip maintains landing position knee keeps toe at landing position ankle maintains landing position
Compression	tail mirrors hip position hip keeps knee on toe-COM line knee zeros pitch and pitch velocity ankle moves to dissipate energy
Thrust	tail mirrors hip position hip keeps knee on toe-COM line knee zeros pitch and pitch velocity ankle moves to add energy
Liftoff	tail mirrors hip position hip maintains position knee maintains position ankle maintains position
Escape	tail mirrors hip position hip maintains position knee starts to move to landing position ankle moves to flight position
Flight: Ascent	tail mirrors hip position and zeros pitch hip starts to move to landing position knee moves toward landing position ankle moves to flight position

Table 2.2: State Machine Actions

Similarly, by placing the foot a little closer, it can speed up.

The robot thus has three levels of control. The lowest level controls the servos at each instant for positional control of each actuator. The second level operates during stance to choose the setpoints for the lowest level, and is responsible for controlling the hip position, zeroing pitch, etc. The third level operates between steps to control forward speed and hopping height.

The virtual leg idea is implemented in the control of the hip. The hip is generally responsible for keeping the knee on the line between the toe and the center of mass. This configuration makes the leg behave as a single telescoping leg between the COM and the toe.

There are several sorts of stability that are implemented. First, there is a continuous control during flight and stance to stabilize pitch. Secondly, there is a cycle-to-cycle stability that uses discrete control to stabilize forward speed. By controlling foot placement, a discrete operation, the machine should converge on a given forward speed.

One underlying assumption behind our control strategies is that the various degrees of freedom can be decoupled. An example is that in the case of a telescoping leg hopper during the stance phase the hip is applying torques to zero the body pitch while the leg is applying thrust forces to launch the body from the ground. Here, control of the vertical oscillation and body pitch are assumed to be independently controllable. Another form of decoupling is between forward velocity and hopping height. Forward velocity is controlled by varying foot placement, and hopping height is controlled by varying the amount of thrust. These two control mechanisms do not act simultaneously, and are assumed to be independent.

Over the regions in which we are interested, these assumptions generally appear to work. Coupling can be observed between some of these axes, but each independent control system treats the effects of the other control axes as disturbances, and the robot settles into an equilibrium.

Before the physical machine was finished, a simulation incorporating the estimated machine physical parameters was constructed to test the control strategy. Due to the similar basic algorithm, much of the software developed to control the simulation was borrowed from previous machines. The code to compute the physical model of the machine was generated by a program from Symbolic Dynamics entitled SD/FAST.

2.3 Mechanical Design

An illustration of the machine appears in Figure 2-1. The machine has been constrained to operate in two dimensions by attaching it to a planarizing boom (see Figure 2-3) approximately 2.5 meters long. The robot is mounted on a pivot on the end of the boom to allow it to pitch freely, so the robot has three spatial degrees of freedom: X translation, Z translation, and the pitch rotation. Yaw, roll, and Y translation are prohibited by the boom. The boom also allows the X translation axis to be mapped onto a circle so that there is an essentially unlimited travel. The boom also permits easy "wrangling" of the machine during operation by pulling on a rope attached through a pulley at the ceiling.

The robot consists of a body, leg, and tail. The body is a bolted framework of aluminum struts, and the leg is a framework of welded aluminum tubes. There are four joints: the tail and hip have collinear pivot axes at the hip, and there is a knee and an ankle joint. There are four actuators, one for each joint: the hip and tail actuators are mounted side by side on the frame, the knee actuator is mounted on the back of thigh, and the ankle actuator, mounted on the back of shin, is attached to the foot through a system of springs and a tendon cable. At the end of the foot is a hemispherical rubber toe with a switch to detect ground pressure. The actuators are low-friction double-ended hydraulic actuators controlled by high-speed hydrodynamic servovalves, and were made in-house.

The torsional spring discussed in the previous section is implemented as a set of steel coil springs mounted between the ankle actuator and a constant radius drum at the back of the ankle (see Figure 2-1.

The robot has only one leg. Since a kangaroo uses its pair of legs in tandem when hopping, we decided to simplify the machine by only having one leg, which eliminates problems of synchronizing ground contact. The leg was originally designed to be used on either a two dimensional or three dimensional body, so it is capable of supporting high side loads.

The tail consists of a lightweight aluminum tube with several movable masses that can be adjusted to tune the moment of inertia. Weights were also placed on an additional tube clamped to the side of the robot to tune the position of the center of mass.

The machine is powered hydraulically from an external pump. A tether carrying hydraulic supply and return, electrical power, and a host interface for the onboard computer runs out along the boom.

2.4 Electronics

Computing is handled on two levels (see Figure 2-15). A microcontroller based on a TI 320C30 digital signal processor ("DSP board") mounted on the top of the robot handles all the real-time control. A VAX 11/785 interfaces to the DSP board and handles data collection, operator control inputs, and user interface. The DSP board is fast enough to servo the machine at 500Hz, although the VAX can only sample the data at 100Hz.

The DSP board was designed in-house. It has 16 analog inputs and four analog outputs. There are sixteen sensors on the machine: seven position sensors, one for each of the degrees of freedom, two length sensors to measure the extension of the hip and ankle actuators, and six pressure transducers to measure the hydraulic pressures in the hip, tail, and knee actuators. The X position measurement uses a quadrature pot, so it requires two input channels. In addition, there is a digital input from a switch in the foot to determine ground contact. The four analog outputs control the four actuator servovalves.

2.5 Future Work

There are several issues that future work can address. The disturbance rejection of the machine can be improved to enable more consistent performance. Experiments could be performed in foot placement or gymnastics. Since folding legs can achieve more ground clearance than telescoping legs, hurdling might be an interesting problem. Also, it would be interesting to investigate high speed hopping.

Future work might also include a more detailed investigation of the virtual leg hypothesis and possible reformulations of the control algorithm, possibly avoiding state machines or using one more based on continuous functions.

Mechanically, it would be interesting to try adding compliance at other joints than the ankle, especially the knee. Perhaps the second inverted pendulum formed by the body, thigh, and tail could be partially stabilized by springs at the knee. Also,

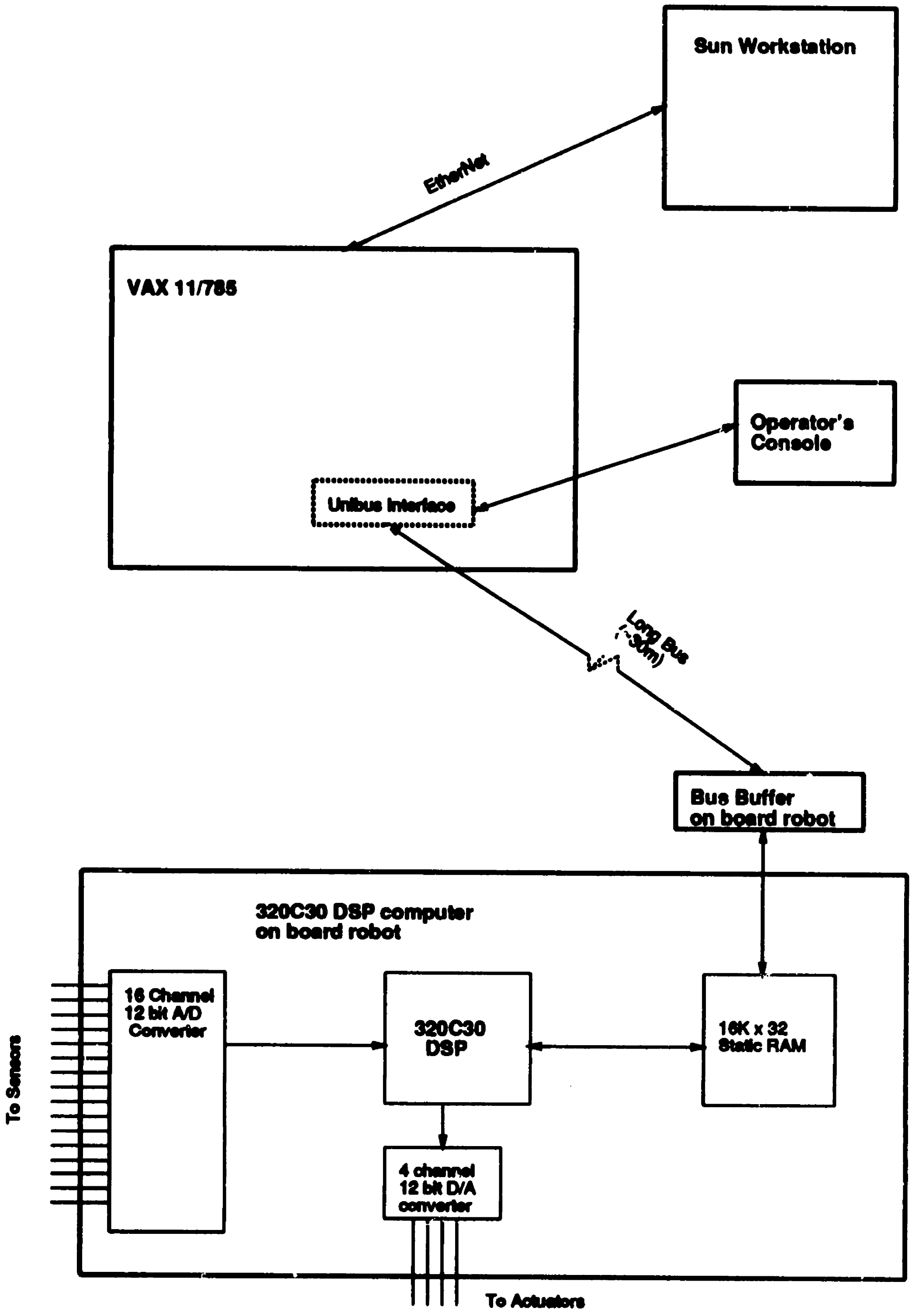


Figure 2-15: Computer Data Flow

perhaps a variable compliance spring could be devised for the ankle to try and modify the vertical resonant frequency of the machine. Finally, the leg could be moved onto a three-dimensional machine to learn how the folding leg approach works for lateral stability.

Chapter 3

Mechanical Design

This chapter documents the work performed during the summer of 1990 to design the leg.

3.1 Abstract

The statics and kinematics of a three-link leg with revolute joints for a hopping robot have been analyzed to establish design parameters. Assumptions have been made based on telescoping leg hopping robots. The result is a leg design modeled after a kangaroo leg, with elastic energy storage and high ground clearance. Included is analysis of the three-link leg forces and discussion of the heuristic and analytic design decisions.

3.2 Introduction

Much work has already been done on legged robots which balance and hop[Rai86]. Raibert's robots have demonstrated hopping and running behaviors with machines from one to four legs. One common feature of the robots constructed so far is a two link pneumatic leg that changes length by telescoping. A linear hydraulic actuator and an airspring in series inside the leg provide energy input and storage. This design works, but suffers from low ground clearance and mechanical difficulties associated with the long linear bearing and the number of custom built parts.

Given these problems, a natural development is to redesign the leg with revolute joints. Having hinge joints instead of sliding joints will alleviate some of the mechanical issues by using simpler and standardized parts, and the similarity to animal structure will hopefully provide insight into problems of animal hopping. This paper presents assumptions, analysis, and decisions that went into the design of the new leg, and a description of the final design.

3.3 Design Requirements

The primary design criteria was to model an animal leg, which includes using revolute joints, elastic energy storage, and high ground clearance. We chose to model a kangaroo because its hopping behavior can be realized with only one leg. We decided to select proportions of the leg members (the thigh, shin, and foot) to be the same as a kangaroo of the same mass as the robot, and to elastically store the kinetic energy stored and recovered during hopping in the tendons. To simplify the design, we chose to only store energy in the ankle tendon, and not have any elastic elements at the hip or knee. We decided that the similarity to the kangaroo should be limited to these two factors, and that it was not relevant to design a similar bone structure or use similar materials, actuation, etc.

This decision to model a kangaroo fixed the general shape of the leg, and we envisioned a shape similar to that shown in Figure 3-1. The figure is schematic, and the actuators are shown for reference, although there are several options for their placement. The particular body shape is not relevant—an outline is shown for visual reference, and is not meant to imply a specific shape. The leg is shown in the nominal uncompressed position. This position is defined by an assumption that the knee will be controlled to remain near the line between the toe and the center of mass, which is a configuration that minimizes knee torques due to ground forces. Evidence suggests that this is a natural configuration[AV75], and we are investigating this assumption further.

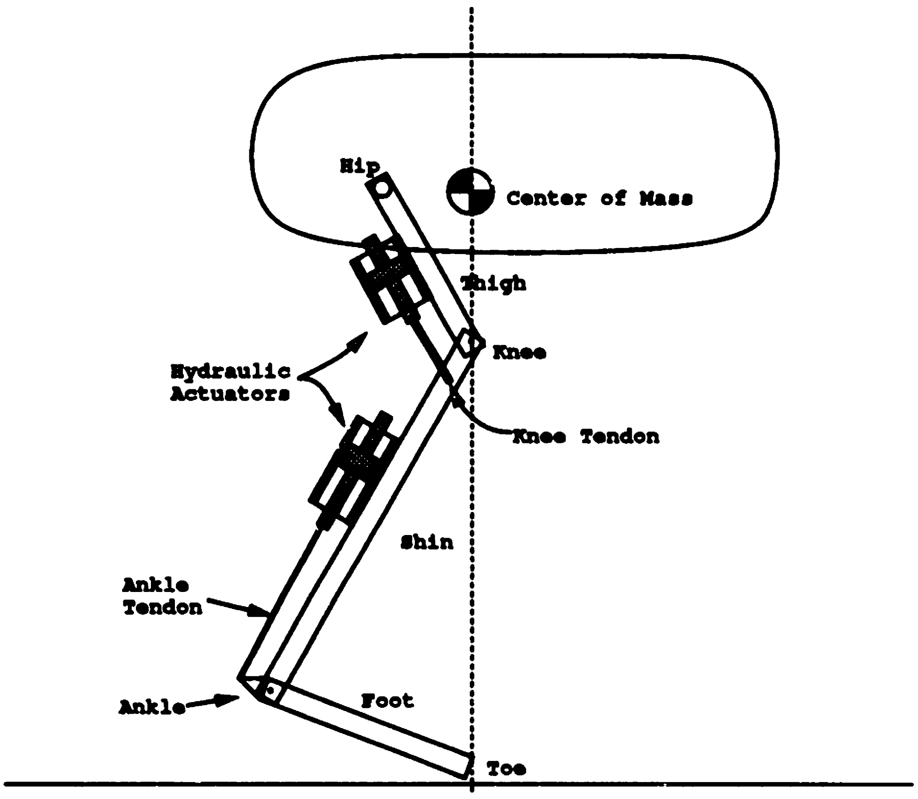


Figure 3-1: Schematic of Proposed Leg in Nominal Configuration

A practical assumption was that we would use actuators we already possessed. These are linear hydraulic actuators with a peak force of approximately 3000 Newtons, either pushing or pulling.

Based on the experience in the lab with building robots, we envisioned a lightweight welded tubular aluminum frame. We also wanted to be able to attach the new leg to different bodies, so we wanted to design the leg to be compatible with several robot bodies.

We also decided that the knee tendon would be a rigid connection to support knee extension and retraction with one actuator. We also assumed that the ankle tendon would be an arrangement of cables and springs, only capable of supporting extension, and would be attached at the ankle at a constant radius by means of a drum.

This initial design process heuristically determined many features. The next step was to estimate the loadings on the various members to help decide how to structure the rigid members and joints. From the results of previous robots, we had some idea of the kinematics of a hopping machine, and we used these results in estimating the forces in the new leg.

3.4 Physical Analysis

We needed to determine the peak loadings in the rigid members and tendons, and calculate appropriate values for the spring constant of the ankle tendon and the attachment radii of the ankle and knee tendons. We shall see that the two calculations are connected.

We can assume what while the foot is in contact with the ground, the leg member velocities are low enough for us to ignore dynamic effects. This is reasonable for a rough engineering analysis to estimate forces. The moments and forces at each of the joints are then dependent only on the ground forces. We will assume that the foot has point contact without slipping on the ground, so that foot-ground interaction can be modeled as a pin joint.

By static equilibrium of forces, we know that the forces and torques on each member must be balanced (see Figure 3-2). We can also see that the forces on each member will have a ground force component which will be equal for all members, and a component due to the forces exerted on each member by the actuators. At the ankle joint, the relative magnitude of the force due to the actuator is much higher than that due to the ground force. This should be clear if we see that the lever arm of the ground force around the ankle joint is five to ten times longer than the lever arm of the ankle tendon, and realize that for a torque equilibrium to exist, the tendon forces and hence actuator forces must be five to ten times larger than the ground forces. At the knee, the actuator forces will generally be much lower than at the ankle if we can keep the knee close to the line between the center-of-mass and the toe.

To calculate the forces, we have used an energy approach. Let us assume that on a typical hop the robot falls from an altitude of 10 cm, and the springs compress over a center-of-mass travel of 20 cm. Furthermore, let us estimate the robot weight to be 20 kg, so if it falls 30 cm, 60 J of energy must be stored. During compression,

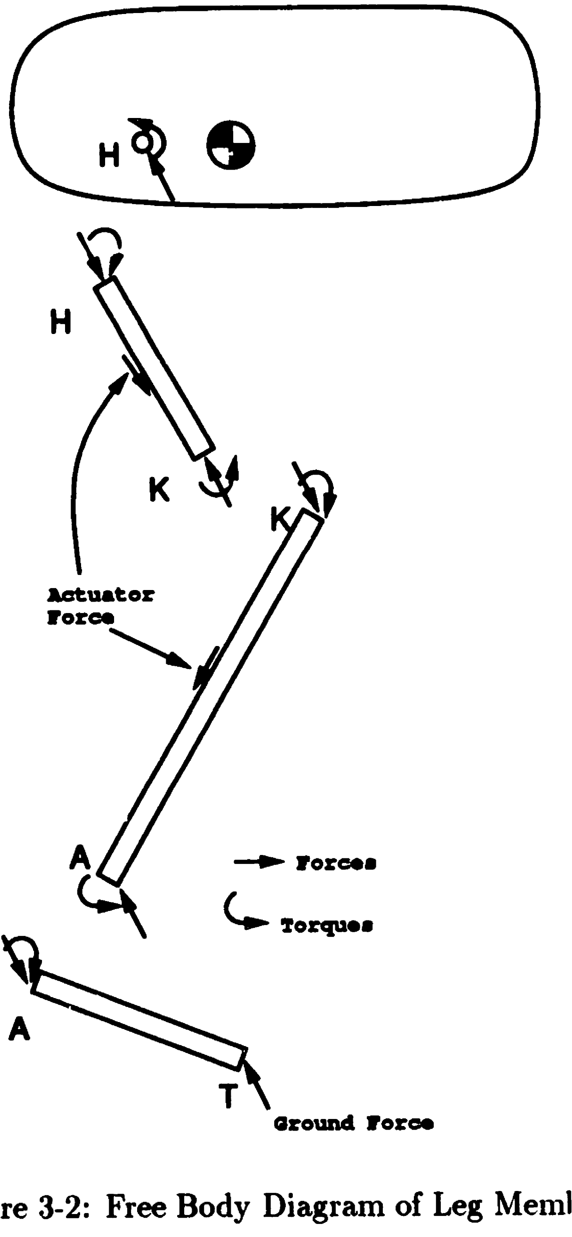


Figure 3-2: Free Body Diagram of Leg Members

the angle of the ankle joint changes 50°, so the extension of the ankle tendon X_t is equal to $(50\pi/180)R_a$, where R_a is the radius of the tendon connection at the ankle. To be conservative, let us assume that all of the energy is stored in the ankle tendon, and none is dissipated in the knee actuator. If we use a linear spring for the ankle tendon, then the stored energy E_t is described by $E_t = (K_t/2)X_t^2$. The force in the ankle tendon F_t at this extension is K_tX_t.

Because the lever arms at the ankle create high actuator forces, we are motivated to pick a large value for the ankle tendon connection radius. We can combine the equations above to find that $R_a = (180 \cdot 2/50\pi)E_t/F_t$ which demonstrates that the ankle radius in inversely proportional to the maximum tendon force. We picked a maximum tendon force of 2700 N, which is 300 N lower than the maximum force the actuator can support. On this basis we chose a value of 5.08 cm (2 inches) for R_a, which makes the ratio between the ankle-toe distance and R_a approximately 5. In an actual kangaroo, this ratio is approximately 4.

Given this selection, we can compute that the tendon travel X_t is 4.4 cm. The energy to be stored E_t is 60 J, so the spring constant K_t is 60 kN/m. The maximum tendon force F_t is then 2600 N. We can see that even for a modest hop, the forces in the tendon and ankle joint can achieve some remarkably high numbers.

Since the energy stored in a linear spring depends only upon its length, this analysis is reasonable. Of course, as the foot and the ground force will rarely be perpendicular, the spring function of the leg as a whole will not be linear.

We can also demonstrate for a given ankle radius, the peak ankle tendon force and the maximum energy that can be stored are proportional to each other. In a system with the spring in series with the actuator, the peak tendon force is limited by the peak force the actuator can withstand, so the actuator limits the energy that can be stored, which in turn limits the possible hopping height. This is a limitation of a series spring arrangement.

3.5 Leg Design

Based on this analysis, we understood several points for the design. First, the ankle joint and tendon have to withstand forces on the order of 5000 N. Second, the shin and thigh members have to withstand compressive loadings from the actuator of the same order. Third, the nominal configuration defined positions for the tendon attachments and joint limits. Also, we decided to go ahead with a ankle tendon spring in series with the actuator for the first generation design.

A sketch of the leg design is shown in Figure 3-3. The standing height is about 70 cm (28 inches) hip to toe. The shin and thigh members are composed of 1.91 cm diameter aluminum tubes and the foot is a 2.86 cm diameter aluminum tube. We used double tubes for the thigh and shin members to provide rigidity. The tubes are welded into aluminum blocks at the joints that house the pivot bearings. An ankle tendon of aircraft cable in series with a coil spring forms the ankle tendon, and is attached around a circular drum section at the back of the ankle. The knee tendon is the steel actuator pushrod. The toe is a rubber hemisphere mounted on an aluminum block with a switch inside to detect ground contact. Also shown are the actuator bodies. The square protrusions on the back of the actuators are hydraulic servovalves.

3.6 Summary

We have designed a new robot leg based on a kangaroo model. The leg is now being constructed, and future work will involve mechanical design revisions of the leg as well as the design of the control system that will animate it.

One specific area of design revision will be to consider alternative arrangements for the ankle tendon springs to allow them to store more energy. One possibility is to redesign the tendon so that it attaches to both a spring mounted in series with the ankle actuator and to a spring fixed to the shin. Other areas of revision will involve decreasing the mass and moments of inertia and redesigning parts that fail.

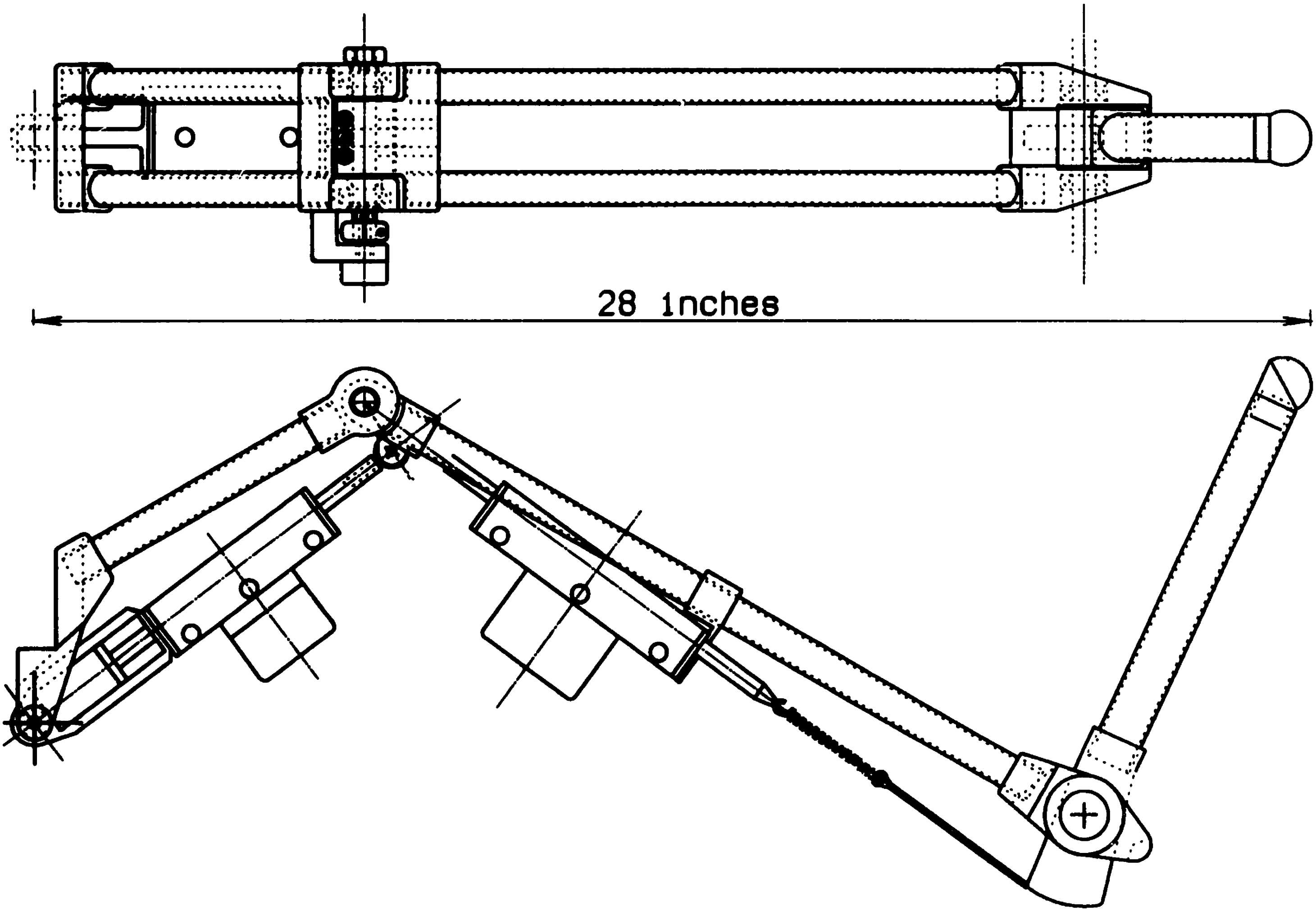


Figure 3-3: Uniroo Robot Leg

Chapter 4

Electronics

The control system of the robot is divided into several major components. Refer to Figure 2-15 for an overview.

4.1 DSP controller

The DSP controller is a relatively simple computer. The CPU is a Texas Instruments 320C30, and the board has 16K 32 bit words of memory, 16 analog input channels, 4 analog outputs, 8 bits of digital input, and 8 bits of digital output. In addition, there is a host interface to allow a VAX host computer to read and write to the memory.

4.1.1 Digital Subsystem

The 320C30 is a digital signal processing microprocessor with an on-chip floating point processor, instruction cache, memory, timer/counters, and DMA controller. The chip has dual 32 bit busses with a easy interface that allows simple external logic. In our implementation, the auxiliary bus is used to communicate to the I/O devices and the main bus is connected to the external 16K word SRAM. The 320C30 is clocked at 20 MHz, which allows a peak instruction rate of 10 MIPS, though in practice, this is only realized when the four stage pipeline is full, and the CPU is reading from the on chip cache and memory. A floating point multiply can be executed in a single clock cycle, allowing a peak rate of 10 MFlops under similar conditions. We did not accurately measure the performance of the processor, but found that it was adequate to run an entire cycle of the control code in just over a millisecond. In practice, a 500Hz servo cycle time was used.

The board was constructed by manually wire-wrapping the circuit on a custom designed board which has power planes, pads, and drilled holes. This construction methodology was adequate for 20 MHz operation, although there were mysterious problems with the host interface at faster rates. There were occasional mechanical problems from machine vibration and hard falls, but reseating chips was sufficient to fix these.

4.1.2 Analog Subsystem

The analog portion of the circuit consists of a 16 channel A/D converter and a four channel D/A converter. The A/D (a DATEL HDAS-16MC) is a 12-bit converter with a conversion time of 20μsec in a hybrid module with a 16 channel multiplexer and sample and hold amplifier. TL084 op amps are used to buffer the input signals. The sensor signals are all high-level (+/- 15V) so there were no noise problems.

The D/A converter is a Analog Devices AD390KD 12 bit converter with LH0041CJ power op amps wired as current amplifiers to drive the hydraulic servovalves. In addition there are 8 bits each of digital input and output, which were used to drive a diagnostic LED and read the footswitch input.

4.1.3 Host Interface

Most of the complexity of the digital portion of the controller is due to the host interface. In order to boot the DSP board, record data, and provide control information, the VAX can read and write to the DSP memory and control its reset input. The 320C30 has a mechanism whereby an external device can request the external bus, which is used to implement this shared memory scheme.

The only consistent failures of the board were centered around this interface. At high microprocessor clock rates, reads and writes to certain addresses would intermittently fail. In the interest of time, we chose to run at a reduced clock rate rather than find the exact cause of the problem.

External to the DSP board is a bus buffer board that has line drivers to receive and send signals down the long line to the VAX.

4.2 Sensors

There are only three sensor types on the robot: pressure, linear position, and rotary position. There are six pressure sensors with integral amplifiers to sense oil pressure on each side of the hip, tail, and knee actuators. There are two linear potentiometers to measure the actuator length of the hip and ankle actuators. There are seven rotary potentiometers, one for each of the degrees of freedom: tail, hip, knee, and ankle positions, body pitch, X and Z displacement. The X measurement uses a quadrature potentiometer that requires two input channels. In Figure 2-6, the X position variable **q_x** shows a step singularity as the robot comes past the zero point on its circular track.

The offsets and gains of the amplifiers are not tightly controlled. Instead, precision is obtained by a calibration procedure in which the variable being sensed (a position or pressure) is manually set to several desired values, and the binary A/D conversion values are used to compute scaling factors. These factors are used to transform the binary input into physical coordinates when the sensors are read, so all sensor inputs are immediately converted into floating point values in real world units.

4.3 VAX host

The VAX 11/785 host has a custom built interface to drive the long lines from the bus buffer. This machine is used only to read and write data from the DSP, and is part of the system for historical reasons. Future machines should use a more modern workstation for this function. The machine also has an interface to an operator's console that has a number of switches, sliders, LEDs, and a joystick.

4.4 Workstations

Data viewing and analysis take place on Sun workstations running X windows. The lab has developed software to display machine and simulation data graphically, both as plots and as animated cartoons. The cross-compiler for the 320C30 also runs on the Suns.

Chapter 5

Control Software

This chapter is intended to provide some detail on the structure of the control software and on the specifics of the hopping algorithm.

5.1 Software Architecture

The software architecture reflects the divisions of the control strategy. There is program code to read the sensors, make state transitions, decide state actions, compute actuator commands based on the servocontrol modes, and write to the actuators. There is also code to synchronize the DSP operation with the VAX host.

The interface to the software on the VAX host is through a set of shared memory buffers, which are used for values that are read from the DSP such as sensor readings and program variables. There are also a set of single buffered areas which are used for values written to the DSP such as gains, operator inputs, and initial values. During normal operation, every 10 milliseconds the VAX gives the DSP a start command and reads data from the inactive double buffer. The DSP uses its internal timer to perform 5 servo cycles, writing to the active double buffer, at which point the DSP waits for the VAX command to start again. In this way, synchrony is maintained between the DSP and the VAX.

The sensors are read at the beginning of each DSP cycle. Analog to digital conversions are performed, the binary values are read and are immediately scaled to a

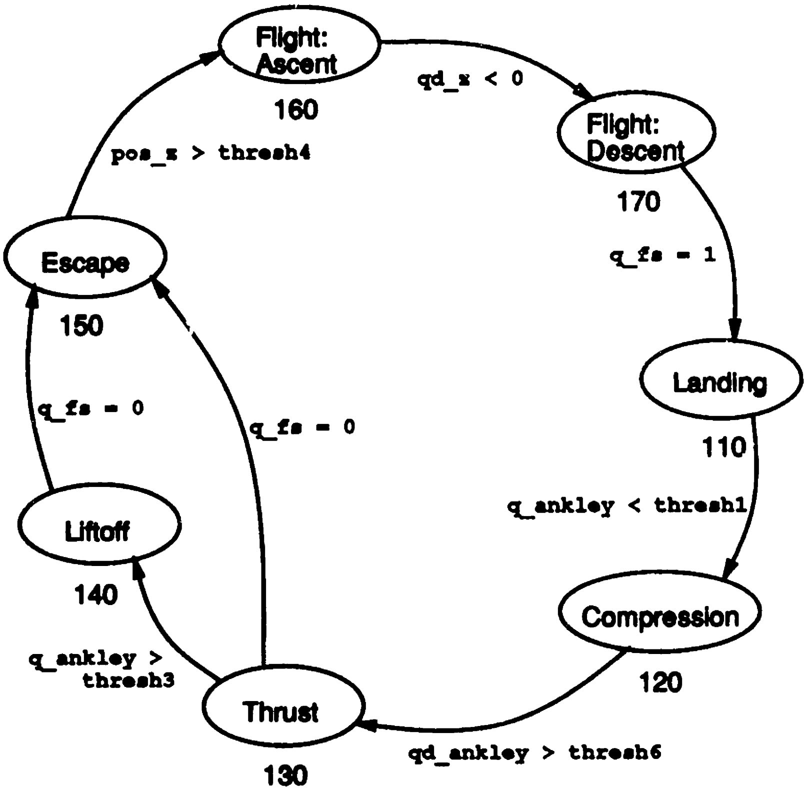


Figure 5-1: Hopping State Machine

floating point value in real units. Derivatives are computed with a set of ring buffers which keep typically 10 previous values for that sensor channel. The oldest and newest values are differenced each time step and scaled by the time duration to be in real world velocity units.

The hopping code is a implemented as a state machine. A diagram of the states and state transitions appears in Figure 5-1. The code to determine the machine action for each state follows the state transition code, and Table 2.2 has a list of the actions.

To control each actuator, simple PD controllers are implemented, which have the following form:

$$\begin{aligned}\text{actuator command} &= k\,(\text{desired position} - \text{measured position}) - \\ &\quad k_d\,(\text{velocity})\end{aligned}$$

This example is a position control servo; k and k_d are the gains. For each actuator there are several servo modes; some are simple position control, while others couple different coordinate axes, such as a knee mode in which the knee is controlled by the pitch and pitch rate. In a few cases, multiple axes are coupled, as a tail mode in which it is controlled by both the hip position and the pitch angle.

5.2 The Algorithm

The basic algorithm is very similar to the three part control discussed in [Rai86]. It should once again be emphasized that the choice of algorithm is only one among many possible algorithms which might work. In our synthetic approach to hopping, we do not assume that we have found a control strategy necessarily similar to the actual animal, or that we have found an optimal control strategy.

5.2.1 Pitch Control

Pitch is primarily controlled by the knee during stance. A torque is exerted at the knee between the thigh and the shin. Since the hip actuator uses a flow control servovalve, it is relatively stiff and transmits the torque to the body without deflection. The torque to the shin is coupled through the ankle to the ground. The knee torques depends on both pitch and pitch velocity, so the servo tends to return the pitch to zero but is highly damped. A chief concern is to leave the ground with zero pitch velocity, since it will integrate during flight and yield a large pitch error on the next landing. Many failures of the machine occur when the body is pitched far forward, so this servo is key for both intrastep and interstep stability.

One side effect of the knee servo is that the torque to zero the pitch couples into a ground force in the X direction. If the machine is pitched forward, the knee exerts a torque in the negative direction on the body, which corresponds to retracting the knee. This retraction causes a ground force that accelerates the body forward.

5.2.2 Forward Speed Control

Forward speed control is a interstep control. By this, I mean that it is a discrete control strategy that operates between each stance phase by selecting a foot placement that will speed up or slow down the machine. The calculation to compute foot placement is as follows:

$$\begin{aligned}\text{toe position} &= (\text{velocity})(\text{stance time})/2 + \\ &\quad k(\text{velocity} - \text{desired velocity}) + \\ &\quad \text{bias}\end{aligned}$$

This toe position is measured from a point directly beneath the center of mass, and is used to calculate hip, knee, and ankle positions during flight. In Figure 5-2 appears a graph of the X velocity and a graph of the foot placement decisions. Notice that the X velocity has several negative spikes, which were caused by sensor noise. Notice also that the foot placement calculations are done at several discrete points during the flight phase.

The origin of this control law is discussed in Reference [Rai86]. It is an approximation of the foot placement for a symmetric hop. What is meant by symmetric? For the bouncing ball, if a uniform ball is bouncing on a smooth surface, the angle of recession equals the angle of incidence, i.e., the motion is naturally symmetric. This leads to a functional definition of the symmetric hop as one in which the relative magnitude of horizontal and vertical kinetic energies was the same after impact as before. In practice, we have not calculated explicitly the exact point of symmetry, but have used an approximation good enough that the machine tends to converge on a symmetric solution.

In order to make the Uniroo hop, a bias of 2 centimeters was introduced into the foot placement, which makes the toe go a little further ahead than nominal on each step. This mechanism needs further refinement, as the same bias prevents the

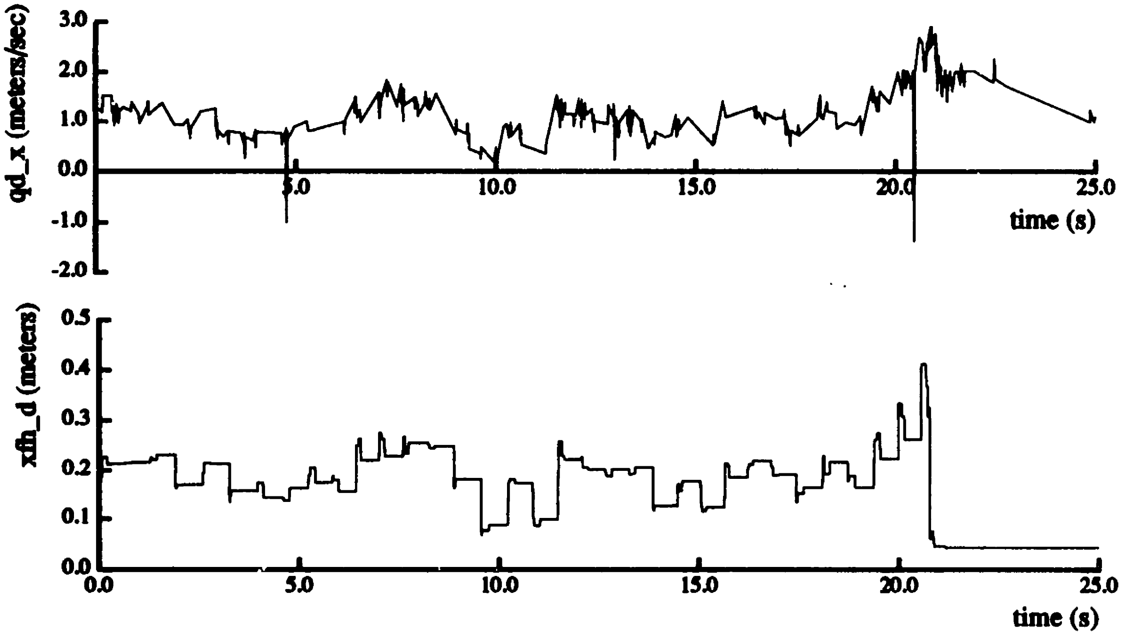


Figure 5-2: Foot Placement

machine from hopping in place.

5.2.3 Thrust Control

Thrust control is also a interstep control. At the apogee of each flight phase, the vertical position is compared to a desired vertical position and the result is integrated into a control variable. The thrust control variable determines how much the ankle actuator retracts during the thrust phase when it is adding energy into the ankle spring.

Figure 5-3 shows the vertical machine position and the thrust parameter. There appears to be an oscillation in hopping height. This seems to be coupled to an oscillation in horizontal velocity. It seems that as the speed control mechanism is slowing the machine down, it is adding energy to the vertical motion. The thrust servo is responding by decreasing thrust, which also has some effect on the hopping height. The two mechanisms seem to form a coupled oscillator which is not well damped. Further refinements of the algorithm and choice of gains might seek to damp these oscillations and achieve a constant horizontal velocity and hopping height.

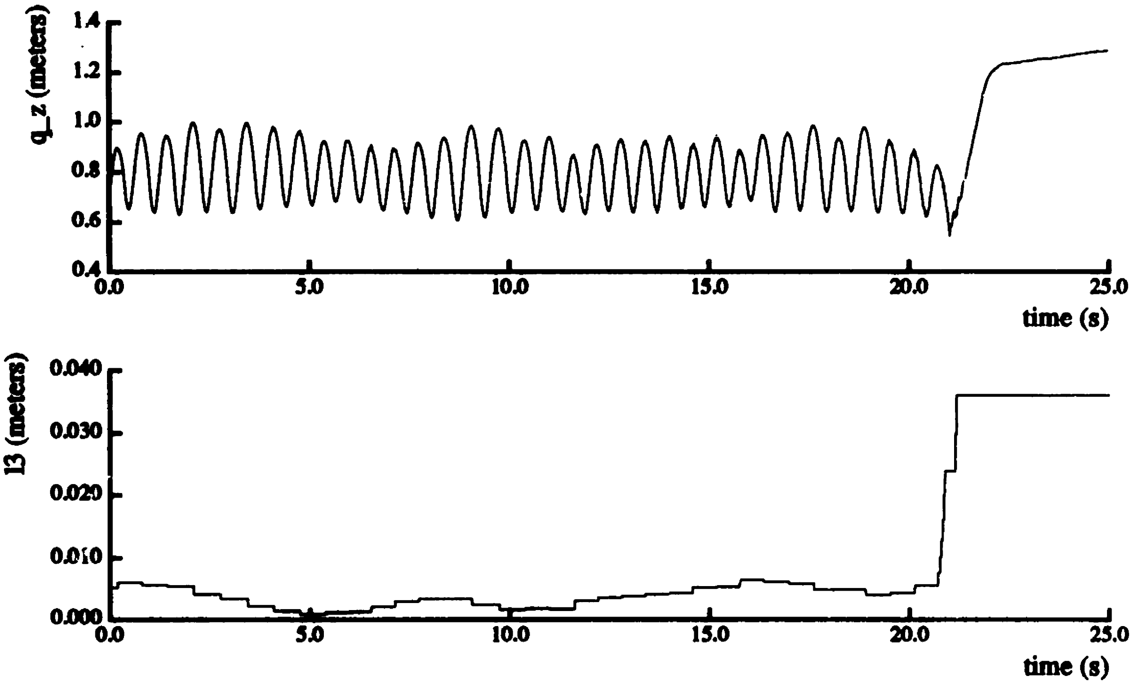


Figure 5-3: Thrust Servo

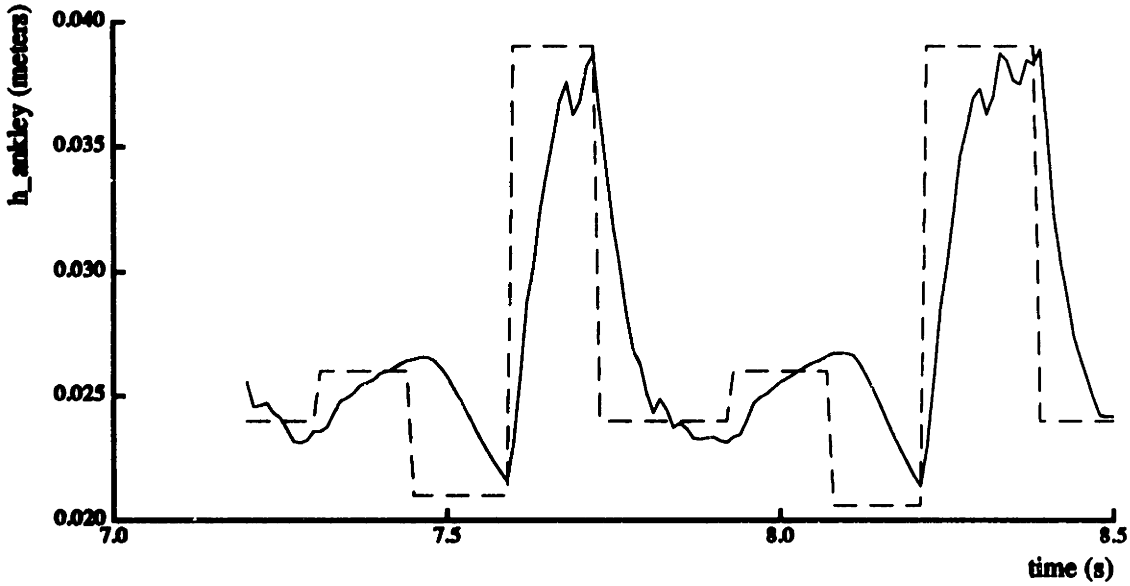


Figure 5-4: Ankle Hydraulic Servo Operation

Figure 5-4 shows the piston position of the hydraulic actuator as a solid line and the desired position as a dashed line. Higher values correspond to the actuator increasing in length, which extend the ankle tendon, bringing the toe upwards. There are four definite phases to the desired position. The peak is a flight position that brings the toe up for ground clearance. The low following it is the landing position. The small step upwards following it occurs during the compression phase, and is intended to dissipate some energy in the spring. This mechanism was an attempt to add damping to the vertical oscillation of the machine. In practice, it had a fairly minor effect. The following step down is the thrust step in which the actuator pulled upwards on the tendon, storing energy in the ankle tendon spring assembly.

5.3 Data

Figure 5-5 is intended to illustrate the state transitions over the course of two steps. The variable `inx` at the top is the state machine index. Below are the footswitch input, the ankle position, and the vertical machine position. The machine is initially in the flight descent phase 170. When the foot switches closes on ground contact, the index changes from 170 to 110. In the first hop shown, state 110 is not visible. Since the VAX can only sample the state of the DSP at 100 Hz, only every fifth cycle is recorded, and state 110 generally lasts a short time. When the ankle joint angle `q_ankley` falls below a threshold, the index changes to 120. When the ankle joint velocity reaches zero, the index changes to 130. In the hops shown in the figure, the index changes to 150 when the footswitch opens again. After a certain ground clearance is reached, the state changes to 160, and when the vertical velocity reaches zero at the apogee, the state changes to 170.

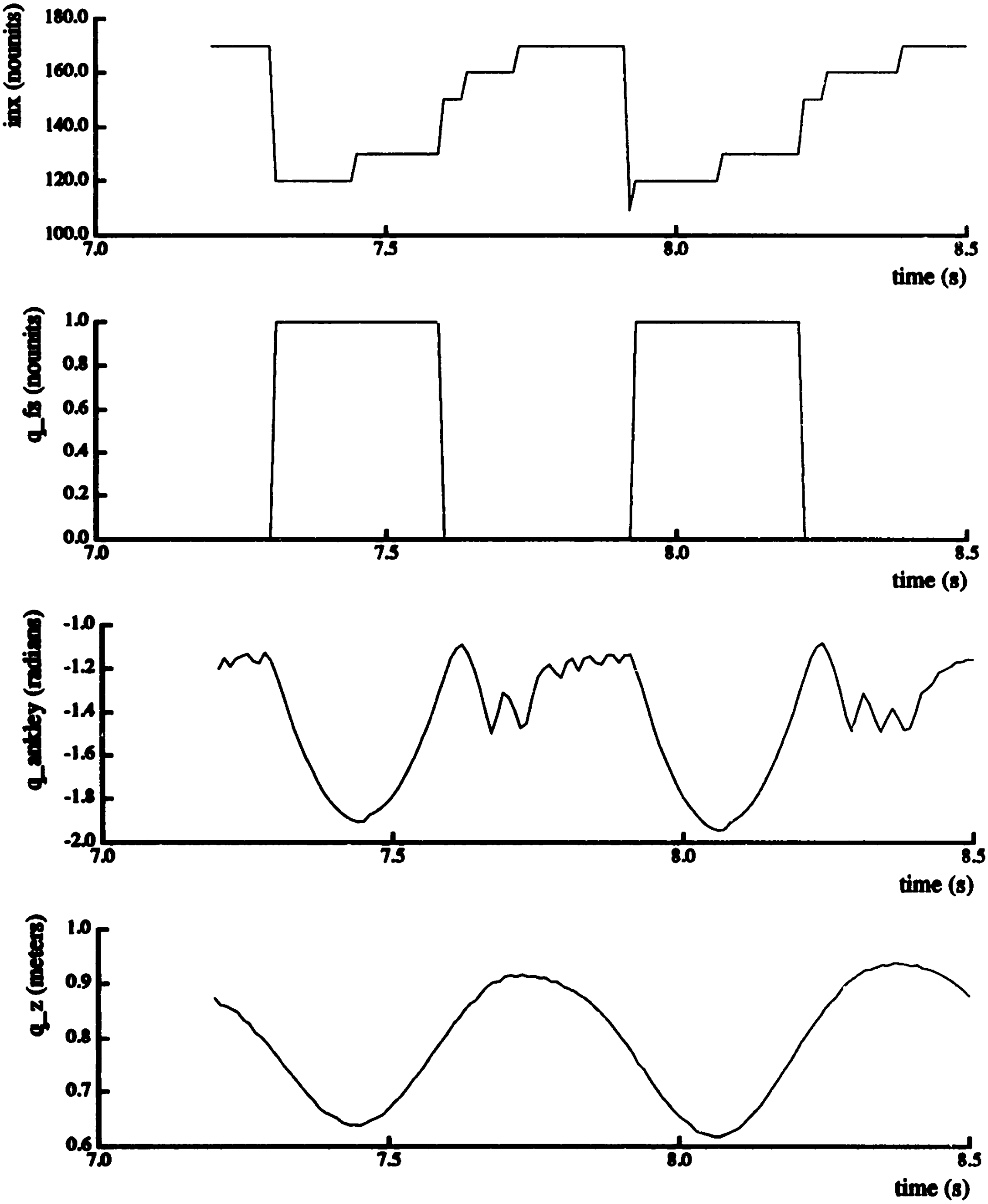


Figure 5-5: Illustration of State Transitions

Appendix A

Statement of Contribution

As part of the Leg Lab, my work on the Uniroo was part of a continuing research process. As such, the work is the product of the research group both past and present. For the purposes of writing this paper, I have written from the standpoint of the laboratory, and describe a project that is part of an ongoing effort without specific reference to my personal contribution.

For the purpose of this undergraduate thesis, I will now state the extent of my personal involvement. During the summer of 1990, I participated in the parametric design of the leg, i.e., performed calculations and estimates which shaped the detailed design of the leg. The actual detailed design was done for the most part by David Barrett, to whom I owe my utmost thanks, and I was only responsible for a few small drawings. The same summer, I took an existing prototype of the 320C30 DSP computer, debugged the design, rewrote software libraries from an earlier generation of DSP controller, and manufactured two working boards. In the fall of 1990, I took the code that controlled the 3D Biped and created a version to control the Uniroo. That semester, David Barrett designed the body for the Uniroo, and he and I collaborated on determining the necessary features. I took the finished body and was responsible for all of the wiring and electrical testing. The same semester, I took an existing simulation of a Uniroo, and adapted it to simulate the physical machine. In the spring semester of 1991, the last few parts of the machine were manufactured and a long process of debugging began. During this process I was responsible for debugging all software, writing new algorithms, fixing electrical problems, and the long process of actually getting the machine to hop. This last process involved understanding the physics of hopping and slowly testing different algorithms and adjusting the myriad parameters that define the machine's behavior. This culminated in the hopping behavior as described in the text.

Throughout the course of the project, I was responsible for coordinating the development of the various aspects of the robot; the mechanical components, electronics, control software, and control algorithms and parameters.

Appendix B

Photographs

Notes on Photographs

Photo 1

Photo 1 is portrait of the robot. All photographs were taken with the machine unpowered, hanging on its wrangling rope. The front of the Uniroo is on the left side of the picture. The leg of the machine is partially obscured behind the boom.

Photo 2

Photo 2 shows the Uniroo as viewed from the inside of its track. Notice the boom pivot in the lower right hand corner, with the umbilical snaking around it and up the boom.The tail is the slender tube on the right side of the body, angled down at about 30 degrees. The tube that extends beyond the front and rear of the machine is used to adjust the balance of the machine, and a mass can be seen on the front end of it.

Photo 3

Photo 3 shows a closeup of the leg from a slightly forward perspective on the left side of the machine. Notice the rounded toe on the end of the foot, the rubber return springs from the top of the foot to the front of the shin, and the hydraulic hoses that drape around the leg.

Photo 4

Photo 4 is a profile of the shin and ankle from the left side of the robot. Notice the ankle tendon that attaches to the back of the ankle to an assembly of four springs. The ankle actuator is attached near the top of the shin on the back, and the piston rod extends down to the top of the ankle assembly.

Photo 5

Photo 5 is a closeup of the knee joint, taken from the rear of the machine on the right side. The double tube thigh member extends from the top of the picture to the knee joint. On the back of the thigh is the knee actuator, whose piston rod extends down to a clamp block which is pivoted at a clevis on the back of the upper shin.

Photo 6

Photo 6 shows the machine from the outside of its track, on the right side of the machine.

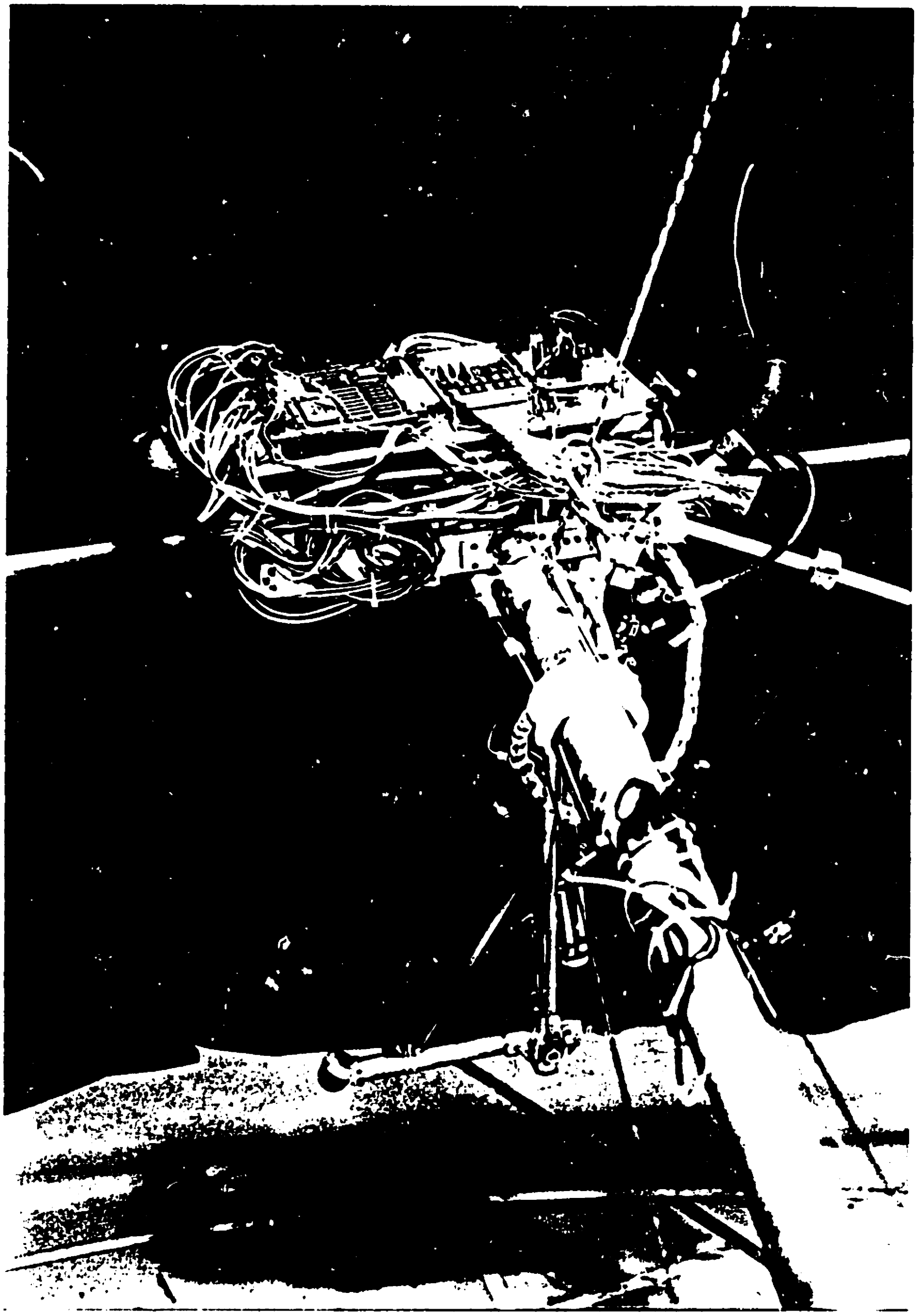


Photo 1

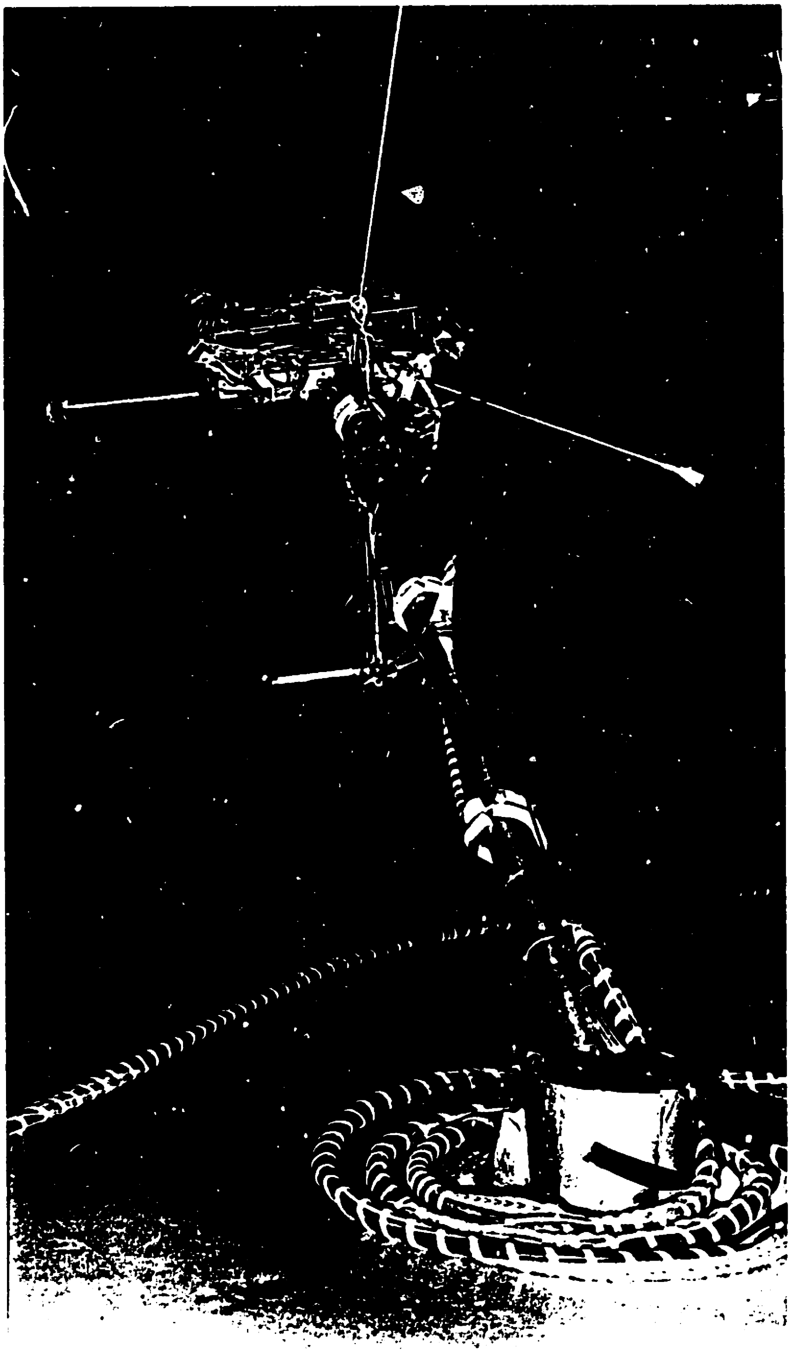


Photo 2

Photo 3

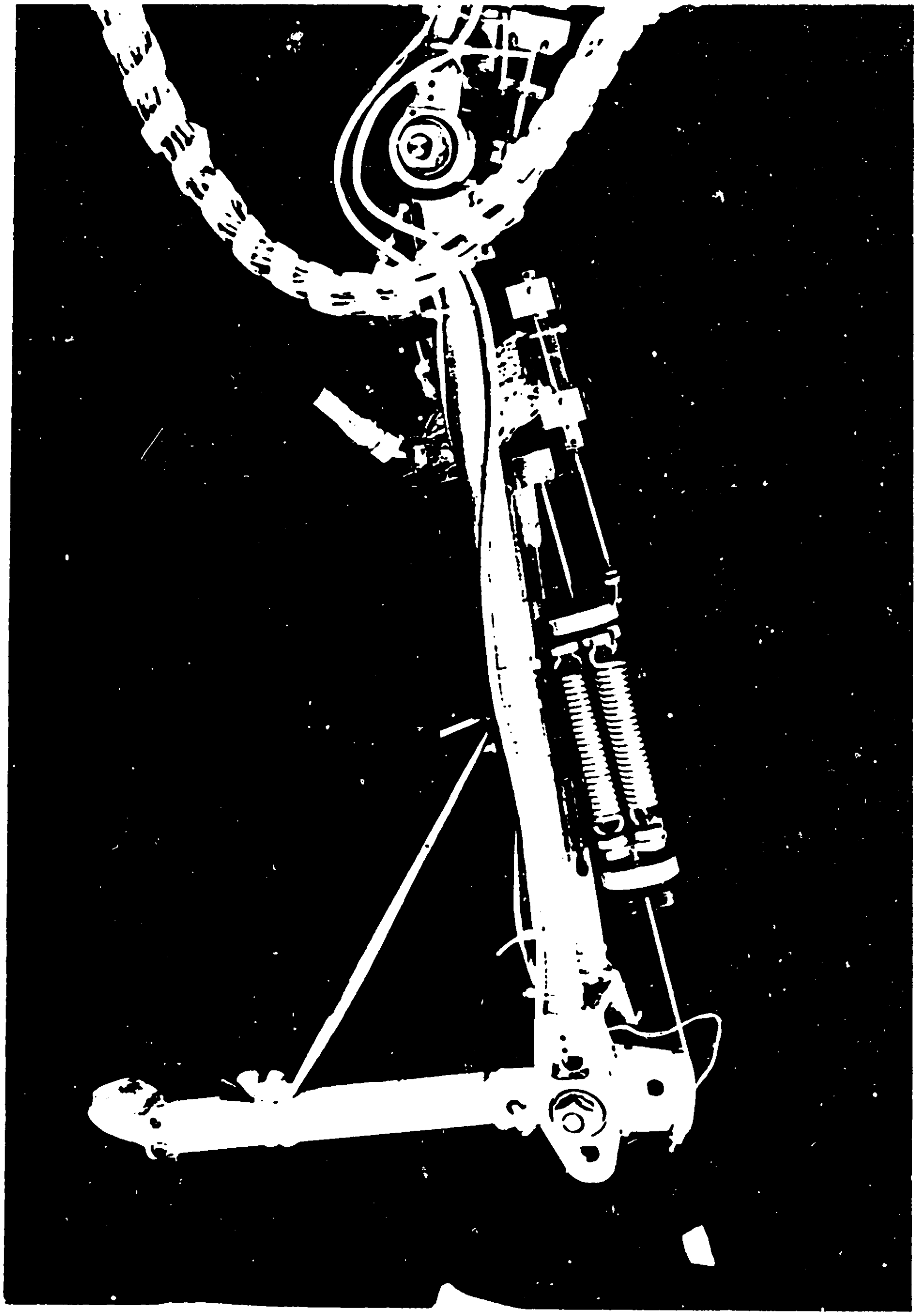


Photo 4

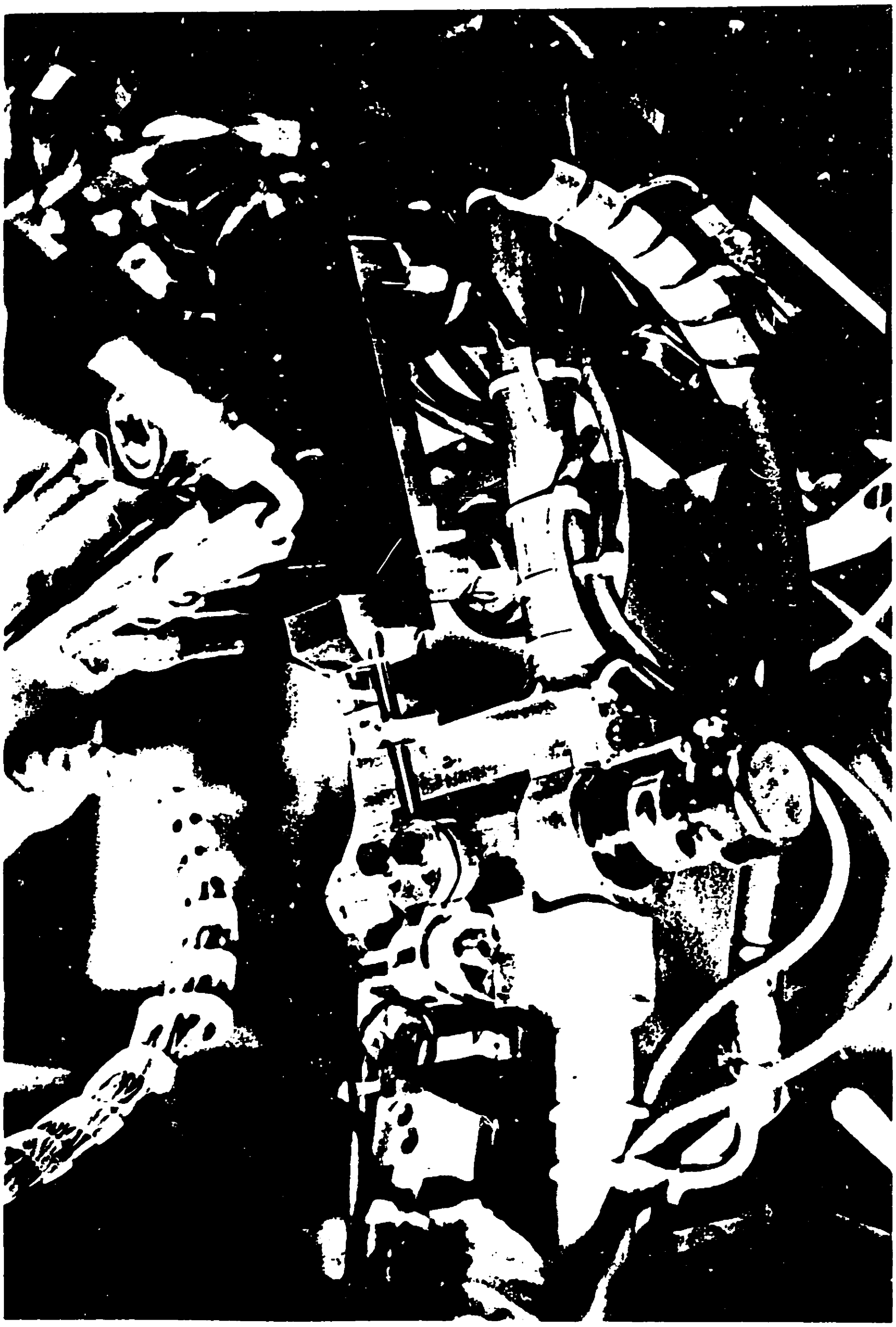


Photo 5

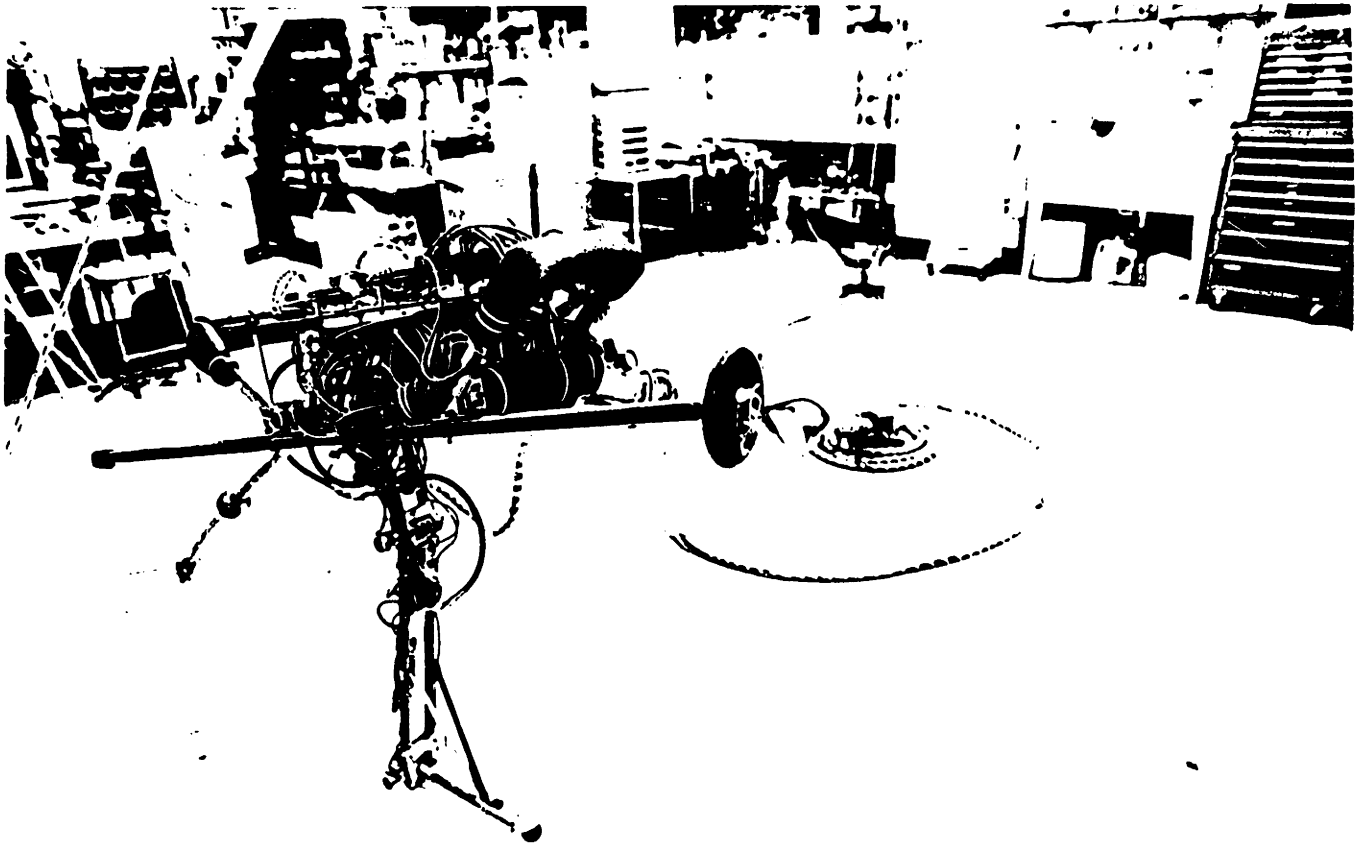


Photo 6

Bibliography

[AV75] R. McN. Alexander and A. Vernon. The mechanics of hopping by kangaroos (Macropodidas). *J. Zoology (London)*, pages 177:265–303, 1975.

[Mar76] Rodolfo Margaria. *Biomechanics and Energetics of Muscular Exercise.* Clarendon Press, Oxford, England, 1976.

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