40

Let there be Luxo: A Jumping Lamp Sheds Light on Heavy Legged Locomotion

by

Daniel J. Paluska

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

Bachelor of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1997
[June 1997]

Signature Redacted

Author ....
Department of Mechanical Engineering
May 9, 1997

Signature Redacted

Certified by ....
Gill A. Pratt
Assistant Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Signature Redacted

Accepted by ....
Peter Griffith
Chairman, Department Committee on Undergraduate Theses

Let there be Luxo: A Jumping Lamp Sheds Light on Heavy Legged Locomotion

by

Daniel J. Paluska

Submitted to the Department of Mechanical Engineering
on May 9, 1997, in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Mechanical Engineering

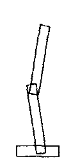
Abstract

After an animation company produced a short film involving a jumping Luxo lamp, two further papers were published using the jumping lamp as an example for physically realistic computer animation methods. In this thesis, the lamp is examined as a simple case of legged locomotion with heavy legs. The physics behind a successful Luxo jump is examined and summarized in the form of linear and angular momentum criterion for the initial launch. The lamp was simulated using a physically realistic software package. A simple control algorithm that incorporated virtual model control was used to achieve the launch criterion and successful jumps were completed. A physical model was built incorporating force controlled series elastic actuators and testing is planned.

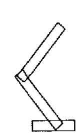
Thesis Supervisor: Gill A. Pratt
Title: Assistant Professor of Electrical Engineering and Computer Science

Acknowledgments

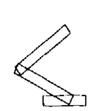
Thanks to all the staff and students of the Leg Lab for sharing their vast base of knowledge and experience with me. Special thanks to Jerry Pratt for having the foresight to make spares of everything and then donating all his spares to me so that he now has none. Special thanks to him as well for answering all my countless questions and calling my bluffs on everything from aerial dynamics to C code. Thanks to Pete and Mike for design advice and inspiration. Thanks to Hugh for introducing me to the "juice" and all its power. Very special thanks to Ben for making all my computer difficulties seem small in comparison. Dave deserves thanks and praise as fine purveyor of good cheer. Chew and John... you guys make me laugh. Andy, thanks for helping me with the quirks of the AI system. Rob Ringrose, thanks as well for computer help and an awesome software package. And thanks to Gill for letting me hang around an already well stocked lab. It'll be up and jumping by the end of May, I swear it!



Contents



List of Figures



Chapter 1

Introduction

This thesis documents the simulation, design and construction of a jumping Luxo lamp. The idea of a jumping lamp may seem a bit odd at first but it happens to be a good building block for more complex legged systems and is in many ways analogous to a single freestanding leg. It is hoped that this research will provide some insight into larger legged system problems.

The original idea of a jumping lamp comes from an animated video produced by the Pixar company entitled Luxo, Jr. [6] In this video an animator gave the lamp "life" and it performed a few simple jumps. It's actions were quite biological due to the skill of the animator. The concept was expanded upon by Fiume et al. [2] and Witkin and Kass [4] who both wished to develop a method in which the animator would not be responsible for making the animation look physical realistic but rather the software would have physics and control built in and the animator could simply supply instructions such as "jump high" or "jump far" and the path of motion would be solved by the computer. One of the main focuses of each paper was to ease the job of the animator, and allow commands to be reused. Several rather complicated motions were successfully completed, including a backflip.

However, neither of these works really looked into the specifics of the lamp construction and its possibilities as a simple tool for learning about legged locomotion, in particular, running or jumping with heavy legs. The presence of heavy legs changes the takeoff conditions necessary for a successful jump. This thesis lays out the initial conditions necessary, linear and rotational momentum, for a successful jump with heavy legs and proposes a way to achieve them. A successful jump from computer simulation is included. A physical model of the planar lamp was built but experiments have not yet been run on it.

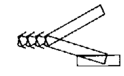
1.1 Summary of Thesis Contents

This thesis is organized as follows:

Chapter 2 talks about the simple physics behind a successful jump. It outlines some simple relations between the forces that need to be exerted and the lamp parameters. A set of takeoff conditions is given that will lead to a successful jump. It also talks about the difficulties of simplifying certain force applications, especially quantifying ankle motion to deliver a specific amount of angular momentum to the lamp.

Chapter 3 describes the simulation software used to analyze the system and talks about the development of a section of C code to control the motion of the robot. We discuss the successes and failures of the simulation. The difficulties in applying the physics and other limitations of the controller are addressed.

Chapter 4 describes the design and creation of a prototype Luxo lamp. The prototype lamp incorporates two series elastic actuators and will be controlled using the code developed through



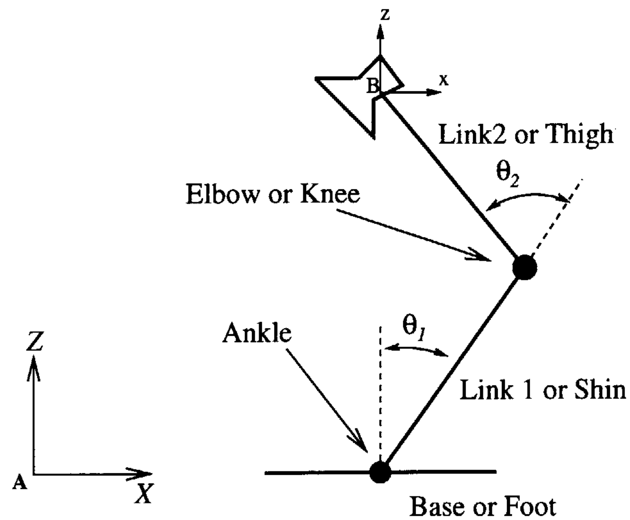


Figure 1-1: A simplified representation of the lamp and it's elements. The parts will be referred to later in the text as they are name above.

the computer simulation. Possible problem areas where the actual robot differs from the simulation are noted and discussed.

Chapter 5 talks about things learned and the path of future research.

Chapter 2

Physics of a Jump

In this chapter we describe some of the physics behind a successful jump. We present a simplified model which provides a simple guide for determining the forces and actions necessary to compel the lamp to takeoff and land as desired.

Raibert found a simple solution for controlled hopping of a 2-D monopod and expanded his work incrementally up to a 3-D biped [9]. These robots had relatively lightweight legs and point feet. The solution involved position control of the legs while the robot was airborne. In order to speed up or slow down in a given direction, foot placement was altered. Since the mass of the legs was small in comparison with that of the body, no significant pitching occured during flight. Therefore, no significant, only minor, torques had to be applied when the robot's foot was in contact with the ground to maintain balance. When in contact with the ground, an inverted pendulum based control algorithm was implemented to guide the robots through the correct trajectory to ready themselves for the next jump. With this simple ballistic foot placement technique and pendulum balancing algorithm the robots ran successfully and rejected small disturbances. However, this control scheme is not sufficient to control the Luxo lamp.

The Luxo lamp has a leg that constitutes a greater fraction of the entire robot mass(the robot is a leg!) and also a foot with an actuated ankle. These differences change the dynamics of a jump and dictate that different control be implemented. Heavy legs are not able to be swung into position for without significantly changing the attitude of the body while ballistic. This chapter presents the initial conditions necessary at takeoff to result in a successful jump. These conditions are an angular momentum H, a linear velocity of the center of mass v_o, and a takeoff angle ϕ as they relate to the moment of inertia, $\overline{I_{yy}}$, of the ballistic lamp. Given the latter three quantities, the H for a successful jump is defined by the following equation:

$$H = \frac{2\phi g \overline{I_{yy}}}{v_o \cos\phi} \tag{2.1}$$

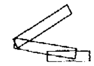
This equation is a combination of Equations 2.3, 2.6, and 2.5.

2.1 Projectile Motion

The ballistic motion of the robot hopping can largely be described by projectile motion physics. From Newton's second law, the maximum height, z_{max}, the maximum distance, x_{max}, and the air-time, t_a, can all be found easily as functions of the takeoff parameters.

$$z_{max} = \frac{v_o^2 \cos\phi^2}{2g} \tag{2.2}$$

$$t_a = \frac{v_o \cos\phi}{g} \tag{2.3}$$



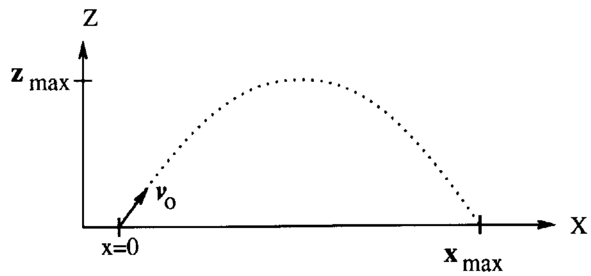


Figure 2-1: The trajectory of a projectile in a plane. The variables are defined in Section 2.1.

$$x_{max} = t_a v_o \sin \phi \tag{2.4}$$

For these equations, g is gravitational acceleration, ϕ is the launch angle as defined from the z axis, and v_o is the initial takeoff velocity.

2.2 The Role of the Ankle

In midair, conservation of angular momentum holds that no torques can be applied about the center of mass from within the robot. This means that if the legs of a robot have significant mass, like those of a human or crazy lamp, they cannot be swung into a correct position for landing without significantly changing the attitude of the body. Springboard divers, gymnasts and cats all have methods of ballistic reoriention without angular momentum [3] but this is rather complicated and does not seem feasible for the Luxo lamp considering the actuator limits and weight distribution. In order to incorporate this technique, more than one rotary joint must be able to affect the value of I_{yy}. Since our foot is considerably lighter than the rest of the leg, the ankle does not significantly affect I_{yy} and this makes reorientation difficult. So, in order to be oriented correctly at the landing, a torque must be applied at takeoff to impart angular momentum to carry the body through the correct airborne rotation.

Proper orientation for landing is defined as an orientation and posture which allows the lamp to absorb or dissipate the body's kinetic energy over time in order to lessen the impact forces and/or to recover energy for the next stride.

2.2.1 A Simplified Luxo Model

The Luxo lamp can be likened to a leg. It has a knee and ankle and three links which are analogous to a foot, shin and thigh. The knee and ankle together are capable of producing a linear force on the center of mass.

Let's consider the following 2-D simplified model that embodies a sort of worst case scenario in terms of swing leg to body weight ratio. There is some rigid body limited to motion within the x-z plane. It is equipped with two massless actuators connected to it. One actuator is a perfect linear force source which can apply forces only in line with the z-axis of the body. In addition there is a torque source which applies a moment about y axis. These sources must interact with the ground. These sources can also recover or dissipate energy like a spring or damper.

When a jump is part of a run, it is optimal for energy to be recovered during each landing for use on the next stride. If this is the case, then the body's z axis should be oriented parallel to the body's landing trajectory in order to absorb the maximum amount off energy upon landing. If it is

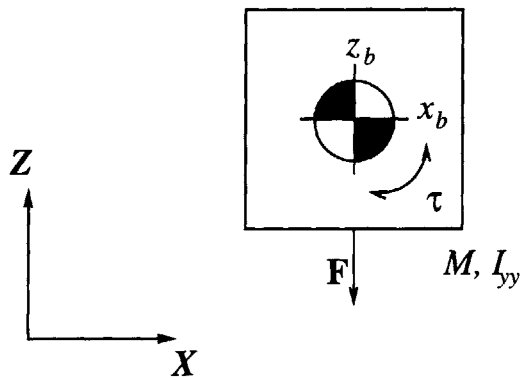


Figure 2-2: A simplified Luxo model. The model is a 2-D rigid body with mass, M, and rotational moment, I_{yy}. It has two perfect actuators attached to it. A perfect force source which exerts forces along the z_b axis and a torque source which acts about the y axis.

a single jump it is still optimal to be oriented this way at landing because it allows the linear force source to dissipate the kinetic energy and lessen the impact forces.

From projectile motion physics it is known that a rigid body's trajectory is parabolic and the angle of approach is complimentary to the angle of takeoff. If the angle of takeoff (ϕ) is defined with respect to the z-axis, first quadrant positive, the body must rotate about the y-axis an angle of 2ϕ while airborne. In order to accomplish this rotation, the body must rotate at a speed

$$w = \frac{2\phi}{t_a}, \tag{2.5}$$

where t_a is the time in the air as defined by Equation 2.3. The angular momentum, H, required to achieve this is

$$H = I_{yy} w \tag{2.6}$$

H has units of Nms. An impulse also has units of Nms so an impulse of value H at liftoff would be sufficient to provide the lamp with the required angular momentum for a successful jump.

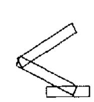
There are some major differences between this simplified model and an actual Luxo lamp. These include,

1. The lamp has the possibility for variable I_{yy} during flight.
2. The lamp has two imperfect rotary joints as opposed to the perfect linear and perfect torque sources of the model.
3. There are geometric differences.

It is well documented how to change linear forces and motions into joint torques and movements of robot arms using the Jacobian[1]. The generation of torque by the ankle is not as straightforward or well researched. The ankle is unlike any problem in manipulator robotics since the robot is not fixed to the ground and is actually leaving the ground when the torque needs to be applied. The difficulties associated with creating the correct torque are discussed in Section 2.2.3.

2.2.2 Variable I_{yy}

During flight, I_{yy} can be a function of t which allows the average value of the rotational moment to be much less than the value at takeoff. There are infinite possibilities for the value of $I_{yy}(t)$ during



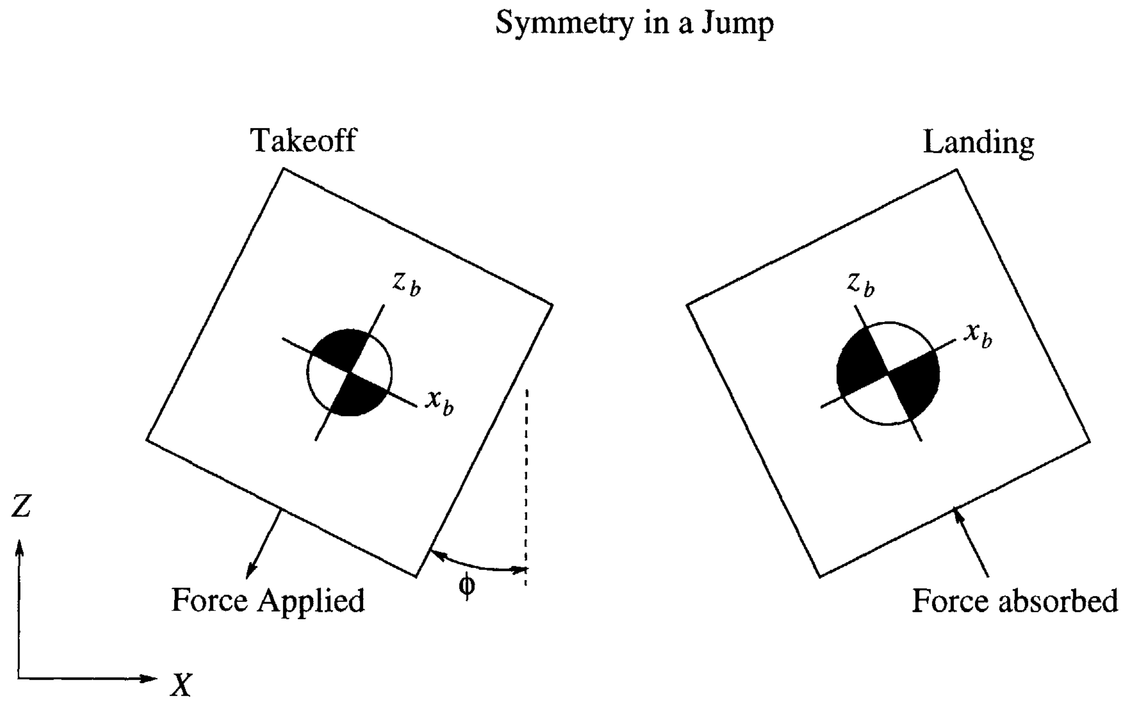


Figure 2-3: A simplified model demonstrating the symmetry in a jump. Since a rigid body lands at the angle it took off at, it is necessary to reorient in flight in order to absorb the impact.

flight. Figure 2-4 shows one possible curve. It is a linear curve that will produce an average value of

$$\overline{I_{yy}} = \frac{I_{yymax} + I_{yymin}}{2} \tag{2.7}$$

The two extreme cases of I_{yy} can be seen in Figure 2-5. I_{yy} is defined as

$$I_{yy} = \sum m_i r_i^2 \tag{2.8}$$

For simplicity let's assume the mass of the lamp is distributed evenly above and below the elbow and therefore, the entire mass can be idealized as two point masses, each located a distance l from the elbow. In this case I_{yymax} and I_{yymin} are defined as follows

$$I_{max} = 2ml^2 \tag{2.9}$$

$$I_{min} = 2ml^2 \sin^2 \psi \tag{2.10}$$

where ψ is half of the elbow angle as defined in Figure 2-5. If ψ is 15 deg then I_{min} is $0.12 I_{max}$ so $\overline{I_{yy}}$ will be $0.56 I_{max}$. By halving the value of $\overline{I_{yy}}$, the required momentum is halved. See Equation 2.6.

2.2.3 Problems of Quantifying Ankle Motion

The amount of angular momentum the ankle needs to provide to the lamp at takeoff is known from Equation 2.6. However, at this point it is not clear the best possible way to direct the ankle to provide this angular momentum. The motion of the ankle at takeoff is not in itself complex but the angular momentum it's providing to the center of mass is.

There are a number of inter-dependent nonlinear motions which complicate the quantification of

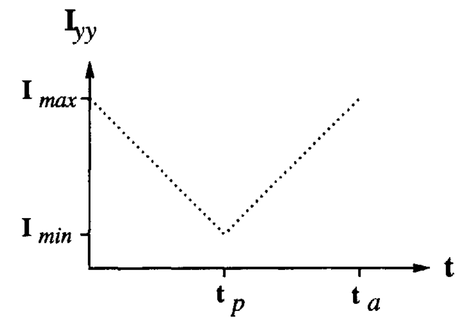


Figure 2-4: A proposed curve of I_{yy} during flight. t_p indicates the vertical peak in flight and t_a is the total flight time.

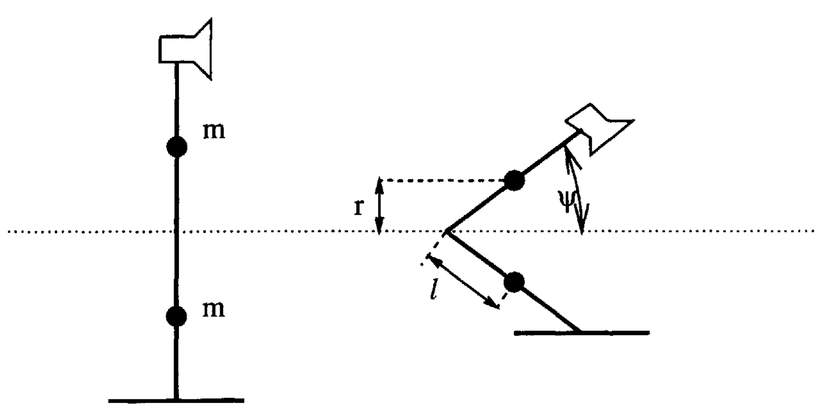
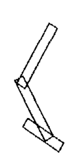


Figure 2-5: Different positions leading to different values of I_{yy}. The center of mass will lie along the dotted line assuming even weight distribution about the elbow. In this particular case, the mass distribution is idealized as two point masses each at a distance l from the elbow.



providing ankle torque.

1. The angle between the ankle and the ground is changing.
2. The distance between the contact point of the foot and the center of mass is changing.
3. The center of mass is moving, both with respect to the ground and with respect to the various lamp components.
4. The foot is coming off the ground.
5. There is always some component of the the reactive force from the ground which adds more linear momentum to the center of mass thereby changing the flight time and changing the angular velocity needed.

Any force exerted by the ankle has two components. The component which is perpendicular to a radius from the center of mass to the toe and a component which is in line with the radius. The component which is perpendicular will provide the center of mass with angular momentum. The component in line with the radius will provide additional linear momentum to the center of mass.

One possible approach to solving this problem is to look at the way that nature has solved it, specifically, looking at the motion of a human ankle during a standing broad jump. Muybridge [5] has documented the motions of humans and other animals thoroughly through still photos. This, however, does not provide as much insight as hoped. There are a number of variables present which alter the role the ankle plays. The main factors that change the role of the ankle are the hips and the arms, each of which contributes a seeming significant amount of energy to the jump and therefore it is difficult to gain any insight from the human example. Other animals were not found to be particularly useful either.

There are a few things we know for sure that we will concentrate on. The necessary conditions for a successful jump are that

1. The center of mass has some linear momentum in positive x and z.
2. Angular momentum is counter-clockwise. Counter-clockwise is defined as rotation from the positive x-axis to the positive z-axis.

Figure 2-7 shows a build up to takeoff which results in these conditions. The right half of the figure shows three positions of the lamp as it approaches lift-off. The circles represent point masses at the head and ankle and the dashed lines indicate the trajectories of the point masses. The figure on the left hand side shows the location of the center of mass. The line between the two point masses represents the orientation of the center of mass. As the lamp goes from stage 2 to stage 3 the line experiences a counter-clockwise rotation which is equivalent to some angular momentum.

In the next chapter, we will discuss simple methods which can be used to achieve the initial conditions without solving any complex nonlinear equations.

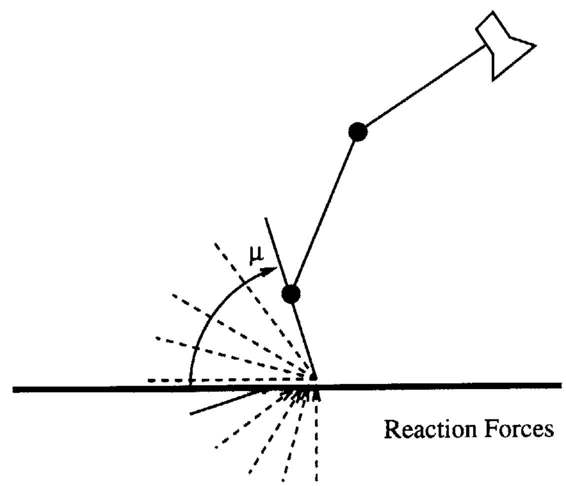


Figure 2-6: One possibility for the path of the ankle and the reactive forces exerted by the ground on the foot during liftoff.

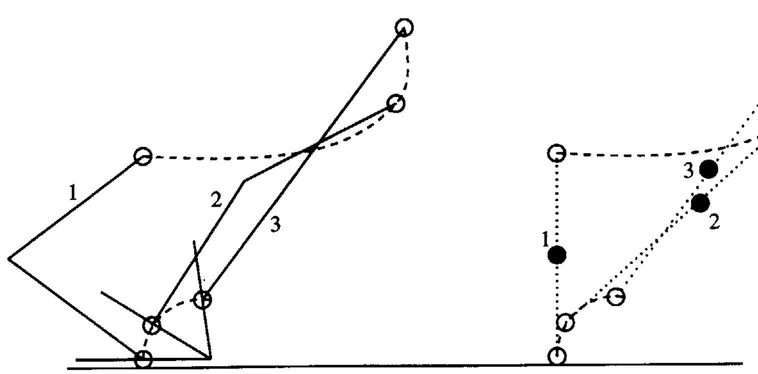


Figure 2-7: One possibility for achieving the desired initial conditions is dictating paths in space for the lamp head and ankle to follow. The circles in the figure represent point masses existing at the lamp head and the ankle. The dotted lines indicate their trajectory. Given a trajectory like the one shown above, the center of mass will have both linear and angular momentum at takeoff.

Chapter 3

Control Development

To get a good feel how an actual system might behave and how to control it, a computer simulation was done. Software previously developed in the Leg Lab was used to simulate the robot. The simulation was based upon estimated weight and sizes of various links and was used to determine and test a control scheme to make the lamp hop. A simple controller for the simulation was implemented to achieve the initial conditions for a successful jump laid out in Chapter 2. Since the initial control parameters were based on estimates they required iterative tuning to achieve the desired result. The lamp successfully performed simple hops with a simple virtual model controller. The control algorithm consisted of a finite state machine. Inside the individual states, virtual components were employed and their set points and component values were set to generate the correct conditions for a jump.

Part of the philosophy of many people past and present in the Leg Laboratory has been to avoid the use of dynamic inversion for control. It is assumed that there are ways of making the dynamics of a robot work for you. Rather than dictate extremely harsh and specific commands it is in the best interest of controlling the robot to give more general commands. These provide a general action and the specifics of the action will be dictated by the dynamics of the system. In legged locomotion, unlike manufacturing and some other robotics disciplines, precision placement is not the main metric. This makes dynamic inversion a bit costly to implement since it's benefits are not really utilized.

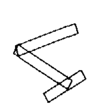
3.1 Virtual Model Control

Virtual model control is an intuitive control technique suggested by Gill Pratt and explored by Jerry Pratt in his thesis [8]. It exploits physical intuition of mechanical elements such as springs and dampers for simplifying the control development process. These linear components can be incorporated in a "virtual" sense to dictate the forces and motion that occur at the rotary joints. The mathematical framework for control of angular joints with linear force commands is detailed by Craig in his book *Introduction to Robotics* [1]. The idea behind virtual model control is that virtual components are 'attached' to the robot. In reality they are simply modules within the control algorithm which dictate forces that mimic those that would be experienced if the components were really there. An example of virtual components to help Luxo stand can be seen in Figure 3-1. The virtual components dictate forces which are to be applied to the lamp head with respect to ground. The position of the lamp head with respect to the ground is defined by the forward kinematic map.

The forward kinematic map is defined as follows:

$$ {}^{A}_{B}\vec{X} = \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} -L_1 \sin(\theta_a) - L_2 \sin(\theta_a + \theta_k) \\ L_1 \cos(\theta_a) + L_2 \cos(\theta_a + \theta_k) \end{bmatrix}. \tag{3.1} $$

Partial differentiation of the forward kinematic map produces the Jacobian.



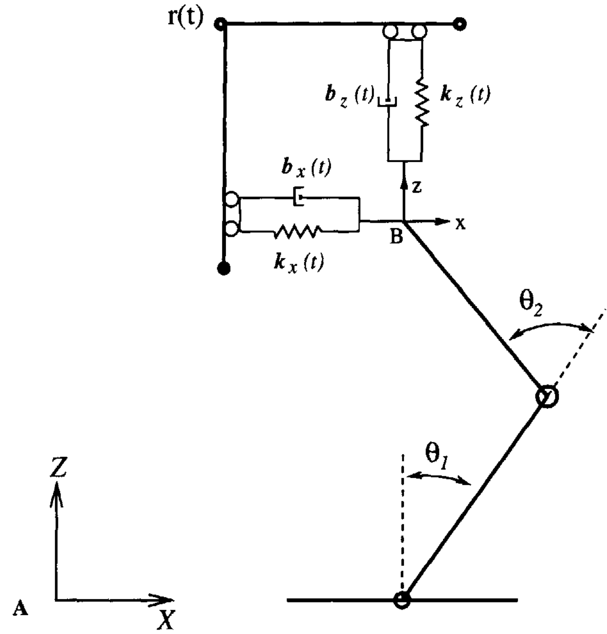


Figure 3-1: Virtual components to help Luxo stand.

$$ {}^{A}_{B}J = \begin{bmatrix} -L_1\ \cos(\theta_a) - L_2\ \cos(\theta_a + \theta_k) & -L_2\ \cos(\theta_a + \theta_k) \\ -L_1\ \sin(\theta_a) - L_2\ \sin(\theta_a + \theta_k) & -L_2\ \sin(\theta_a + \theta_k) \end{bmatrix} \tag{3.2} $$

The Jacobian is used to relate the velocity between frames A and B.

$$ {}^{A}_{B}\dot{\vec{X}} = {}^{A}_{B}J\ \dot{\vec{\Theta}} \tag{3.3} $$

The kinematic map is used to determine the position error which corresponds to a force in the virtual spring and the velocity is calculated using the Jacobian which corresponds to a force in the virtual damper. Once known, the virtual linear forces must be related to the joint torques. The virtual forces are related to the joint torques by the following equation:

$$ \vec{\tau} = ({}^{A}_{B}J)^T\ ({}^{A}_{B}\vec{F}). \tag{3.4} $$

In the case of the Luxo, the Jacobian is of full rank. This indicates that all virtual forces directions are admissible. This changes as the lamp foot lifts off. When this happens, the ability of the ankle to exert torques on the ground is limited and therefore, the virtual forces applicable to the lamp are limited. Virtual model control is also limited at very small angles. In simulation, virtual model control was not very effective in holding the lamp in a completely erect position. This can be explained by examining the Jacobian seen in Equation 3.2. At small angles the matrix becomes sparse and $|J| = 0$. Under these conditions certain forces have no effect[8].

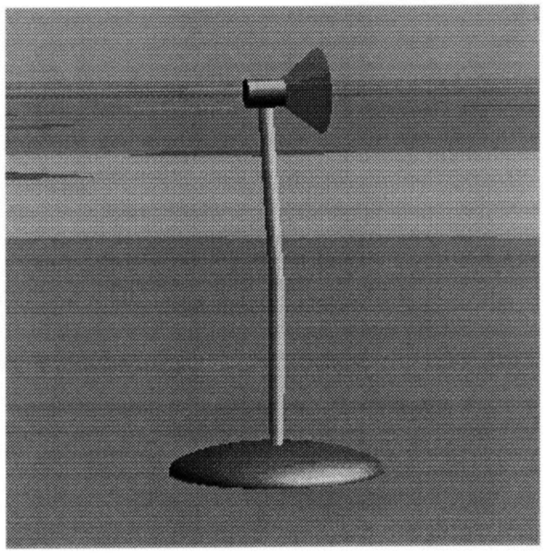


Figure 3-2: A picture of the simulated Luxo lamp.

3.2 Computer Simulation

3.2.1 The Software

The software used for simulation of the lamp is called the creature library. It was written by Robert Ringrose. It incorporates previous software written in the Leg Lab as well as a program known as SD/fast which was developed by Symbolic Dynamics. The simulation package allows you to specify links and joints and how they are connected. Each link must have a parent joint and each joint must be offset from the last joint. Once a robot or a "creature" has been created, a control algorithm must be written. The algorithm is contained in a section of C code which dictates what forces are to be applied at the joints according to the various parameters.

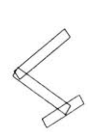
Once control code has been written a simulation can be done. In the simulation the computer follows the instructions laid out in the control code and applies torques at the joints and performs the physics calculation for the motion that will ensue. The simulation parameters can be varied by the user. Parameters such as timesteps can be set according to the accuracy desired. The results of simulation can be recorded and viewed.

3.2.2 The Simulated Robot

The first simulation was in 3-D but this proved to be a little tricky so it was scaled down to the planar case. The simulated robot consisted of a base and two links. The three members were connected by 2 pin joints. The simulated robot was based on the actual projected size and weight of the constructed robot described in Chapter 4. The simulated robot is shown in Figure 3-2. The actuator limits were also incorporated into the simulation. These limits are described in Chapter 4.

3.2.3 The Control Code

The control developed was a discrete state controller. There are a series of states which are each designed for a particular portion of the jump. The states are represented visually in Figure 3-3 and are described below.



Stand Sets the Luxo to stand still in a nearly erect position using virtual components like those shown in Figure 3-1. A timer is employed to dictate when it is time to switch to *Crouch.*

Crouch The setpoint of a virtual linear spring and damper set are moved slowly to coax the Luxo down to a crouched position. Once the crouch is complete a timer is used to determine when to switch to *Launch.*

Launch This is the most critical state. A virtual linear spring and damper are used to extend the lamp rapidly at a given angle to gain momentum for takeoff. Just when the elbow is nearing full extension a virtual rotary spring provides a torque at the ankle and a path of motion similar to that shown in Figure 2-7. Footswitches are used to trigger transition to the next state.

Takeoff In this state the Luxo is completely airborne and a virtual linear spring and damper combo connected from head to ankle is used to bring the Luxo into a crouched position. The foot is PD position controlled(virtual rotary spring-damper) to a desired angle. A zero velocity in z direction is used to trigger the state machine to switch to *Approach.*

Approach This state is triggered when the Luxo starts its descent. The Luxo extend it's base out in order to absorb the landing. Footswitches are employed to tell the state machine when it is appropriate to switch to the state *Landing.*

Landing The Luxo lands and crouches down to minimize impact forces. When the lamp head position is near that of the desired crouch position and the foot is steady on the floor the state is switched to *Crouch.*

The majority of the control within the various states is accomplished through virtual model control. Setups such as the one seen in Figure 3-1 are used to coax the lamp into various positions. Also virtual rotary springs and dampers were incorporated at the joints. This is equivalent to PD position control of the joint angle. However, at any given time, only one set of linear or rotational components was implemented. Although it is possible to use both simultaneously it becomes a bit more confusing and unintuitive. For some states the control is simply a matter of moving the zero point of the virtual spring-damper sets and changing the k and b values of the components. When the robot is airborne, virtual components can only be connected between two points on the robot. Since forces can not be applied against the ground, virtual components attached to ground provide the controller with useless directions for impossible forces! A virtual spring-damper from the lamp head to ankle and position control servoing of the foot was employed in the air.

The simulation software allows you to monitor all the aspects of the motion of the robot: joint angles, position in x,z plane, and velocity, linear and rotational. The values of these parameters are employed in the control code to determine when it is an appropriate time to switch state controllers.

The k and b values for the virtual components were first estimated using simple lumped parameter analysis and then tuned iteratively to obtain the desired result.

In the simulation, state transitions were easy due to the fact that the simulator is aware of the values of all of the lamp parameters and the locations and velocities of all it's individual components. Knowledge of these is limited in the actual robot and as a result of the control code needs to be altered. The possible alterations are discussed in Chapter 4.

3.2.4 A Sample State

This is a small section of the code used to control the robot. The code is written in C. The compiler has been customized slightly and certain operations can be carried out using an alternate syntax. Variables are simply defined where they occur by entering them as **`ls.variablename`**. The code is a large switch-case routine which has a case for each state controller.

These few subroutines precede the case statement and are called every control cycle.

```
jacobian();
```

ɜfile= /home/r2/leinad/Thesis/MAT/cartoon/LUX'cartoon'79.ps hscale=25 vscale=9 hoffset=-150 voffset=-36" "fi "fi

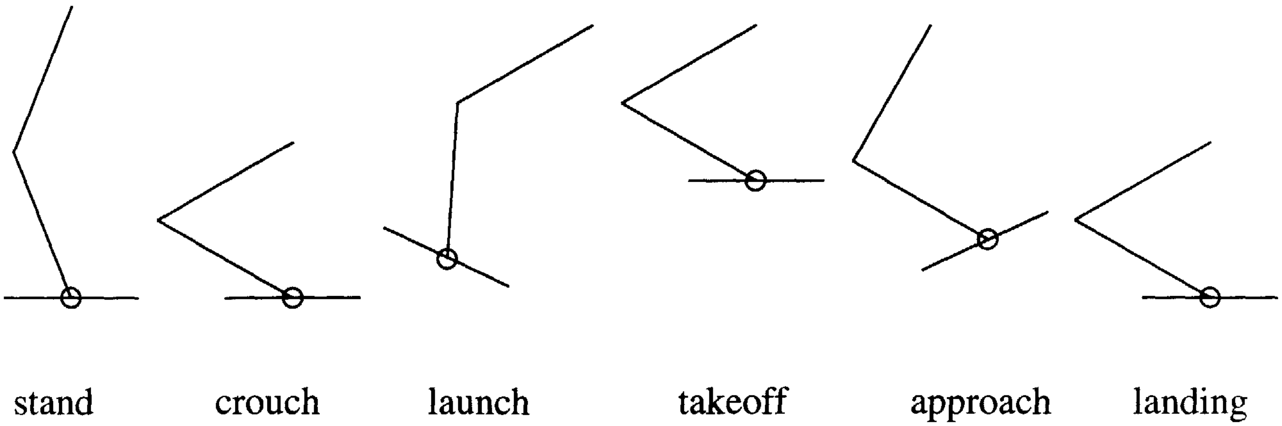


Figure 3-3: These are the control states. The figures indicate probable final state conditions.

```
calculate_pos();
calculate_vel();
check_footswitches();
```

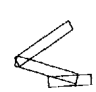
After it calls those lines it checks the state variable to see which state routine to run. The following is the section of code for the state *stand*.

```
case (int) STAND:
   /* STAND - 0 */
   set_standing_params();
   actuate_joints();
   ls.last_t_stand = ls.t;
   break;
```

This is the code for one of the subroutines which assigns values to the virtual components and defines the zero point.

```
set_standing_params()
{
  ls.v_z_k = ls.v_z_k_0;
  ls.v_z_b = ls.v_z_b_0;
  ls.v_x_k = ls.v_x_k_0;
  ls.v_x_b = ls.v_x_b_0;
  zero_joint_k_b();
  ls.x_d = 0.0;  /*desired position of head */
  /* since Jacobian doesn't work  when it's totally upright. */
  ls.z_d = 0.9*(UPPER_ARM_HEIGHT + FOREARM_HEIGHT);
  if (ls.time_to_crouch)
ls.state++;
}
```

The first four lines of the above subroutine assign values to the virtual linear components seen in Figure 3-1. The subroutine `zero_joint_k_b` sets the virtual rotary springs and dampers to zero. `ls.x_d` and `ls.z_d` are the zero points for the virtual spring-dampers sets. At all states in the code, virtual components are present in linear and rotational form. Their component values are simply turned on and off as needed. From the simulation interface any of the `ls.variables` can be set. This allows the values of the k and b components to be tuned without recompiling the code.



3.3 Simulation Results

The code was able to be kept rather simple. With iterative tuning of virtual model components, the lamp was able to complete successful jumps. One successful jump can be seen at the bottom of the pages of this thesis. The jump has been elongated in the x-axis to better illustrate it. The actual jump travel along the x-axis was only about 1.5 times the foot length.

The previously described code was able to generate the proper takeoff conditions that led to successful Luxo jumps. The controller was tuned for a particular takeoff angle and velocity. There is a very large state space of successful jumps and it has not been searched thoroughly. With more time, the controller can possibly be maximized for energy expended per unit traveled. Also a more robust controller can be developed which can generate the proper takeoff conditions in a broader range of cases.

Chapter 4

Physical Model Implementation

In order to test the accuracy of the model, and learn a bit more about the system a robotic model was made. Keeping the design simple, the robot consists of three links and two planar pin joints. The foot is large enough to ensure side to side stability so for all intents and purposes it limits the motion and control problem to the planar case. The size and shape of the model was designed around force controlled series elastic actuators which were designed by Mike Wittig and Jerry Pratt. One actuator was used for each pin joint. The parts of the robot will be referred to as follows: base or foot, link1 or shin, link 2 or thigh, and lamp head. These parts are easily seen in Figure 1-1. The actual physical robot is shown in Figure 4-1.

4.1 Series Elastic Actuators

Most robotic actuators for arm manipulators consist of a motor connected to gear box that boosts the torque and lowers the speed. This is very effective in position control applications but problems arise in areas where force control is more important [7]. Series elastic actuators are actuators which incorporate a spring in series with the motor after the gears in order to isolate the motor from shock loads and monitor the force output. By adding a spring to the end of the gear train it turns force control into a position control problem. Using a PD controller with additional feedforward terms the spring can be displaced to a reasonable accuracy in order to produce a known force. In addition, the spring isolates the gears from possible shock therefore making it possible to use less expensive gears. It also lessens the stress on the motor, gears, and other components of the robot.

4.1.1 Actuator Mechanics

As of yet a good two directional torsional spring has not been found so the actuator designed by Wittig and Pratt turns the rotary motion of the motor into linear. Linear springs are then incorporated and attached in series to a linear slider. The actuator specs are shown in Table 4.1.

Since the actuator is linear, it's motion needs to be converted to rotational. This is accomplished

Table 4.1: Linear series elastic actuator specifications.

Maximum Linear Force	350 N
Maximum Operating Frequency	20 Hz
Weight	920g 2.2lbs
Overall Length Required	18 inches
Width	3.21 inches
Max Stroke	2.6 in

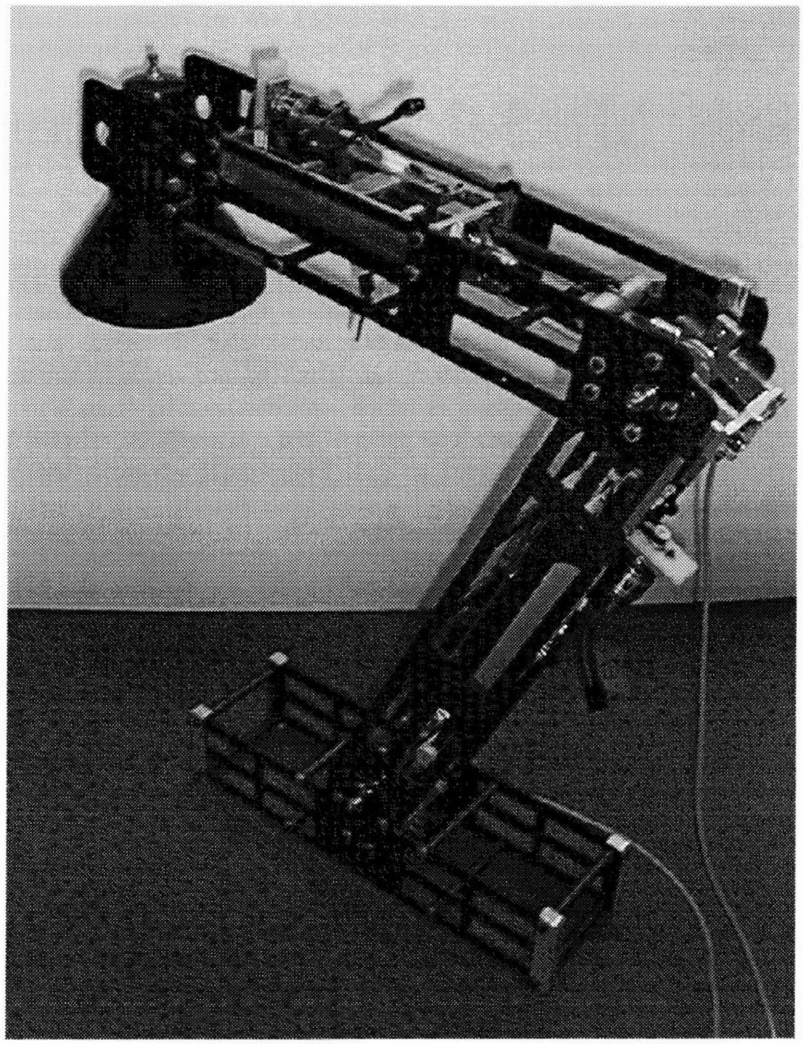


Figure 4-1: A photograph of the actual prototype Luxo lamp. The length of the lower link from joint axis to joint axis is 19.5 inches.

through a cable and pulley system. The design is adopted from a design currently in use on the biped robot Spring Flamingo. The actuator is rigidly mounted in one link which is connected to another link via a pin joint. A cable from the actuator is wrapped around a pulley at the joint and the cable is fixed to the other link. The diameter of the pulley determines the amount of radial displacement the actuator is capable of producing. The radius also affects the maximum torque applicable at the joint. To balance these effects the radius was chosen to be 0.75". This leads to a max torque of approximately 12Nm and a radial displacement of greater than 150 deg. A potentiometer is also mounted on the joint to monitor the angle.

4.1.2 Actuator Control

There is a PD circuit board designed by Jerry Pratt in use with the actuator. It reads the linear potentiometer sensor and drives the motor accordingly. A diagram describing the connections of data and power can be seen in Figure 4-3. The DSP tells the controller circuit the desired force. The circuit compares this with the linear pot reading, which represents actual force, and produces an error signal. The error is amplified through a proportional gain and a derivative gain and sent out to the amp which in turn delivers a current to the motor to correct the error.

4.2 Design and Construction

4.2.1 Component Design

Considering the linear slider clearance needed on each side, the actuator is of considerable size. This constrained the placement options heavily. The actuators were placed along their longest axis in each link. Clearance for the linear slider on the actuator dictated the length of each link and the position of the actuator within the link. A foot size was then chosen a bit smaller than the link. The desired characteristics of the foot are that it is large enough to be stable in standing mode, and that it's symmetric and lightweight.

Each component was modeled individual using the CAD package PRO/Engineer. The parts were then assembled into an assembly drawing to make sure everything would fit. A view of the assembly is seen in Figure 4-2.

4.2.2 Material Selection

The actuators have limited force output so in order to keep from saturating them it was necessary to design the rest of the robot to be as light as possible. From simulation it was established that is was within reason to make an 8 pound robot hop. Since the actuators themselves already weigh 2 lbs a piece, the rest of the components had to be as light as possible. The actuators themselves could have certainly been optimized and made to weigh less but they were already assembled so it was decided to leave them heavy and spend time keeping the other components light. In order to achieve this, carbon fiber was chosen as the main material for the links and base. Additional aluminum supports where incorporated at the joints to hold the bearings and distribute the forces on the carbon fiber in a manner such that the carbon fiber would only be under serious stress in it's strongest orientation. Additional nylon supports where added as horizontal spacers along the length of the link in order to help prevent buckling of the carbon fiber. The pulleys and cable tensioners were made from aluminum for strength. The actual robot ended up weighing about 10 pounds.

4.2.3 Construction

Since it is difficult and hazardous to machine, the carbon fiber was sent out to be machined. DXF files from PRO/Engineer where sent to the machine shop via email. All the aluminum and plastic parts were machined in house. Shafts, bearings and spacers were ordered from standard catalog suppliers. Joint cabling and electrical wiring was available in the Leg Lab from other projects. The robot was assembled and can be seen in Figure 4-1.

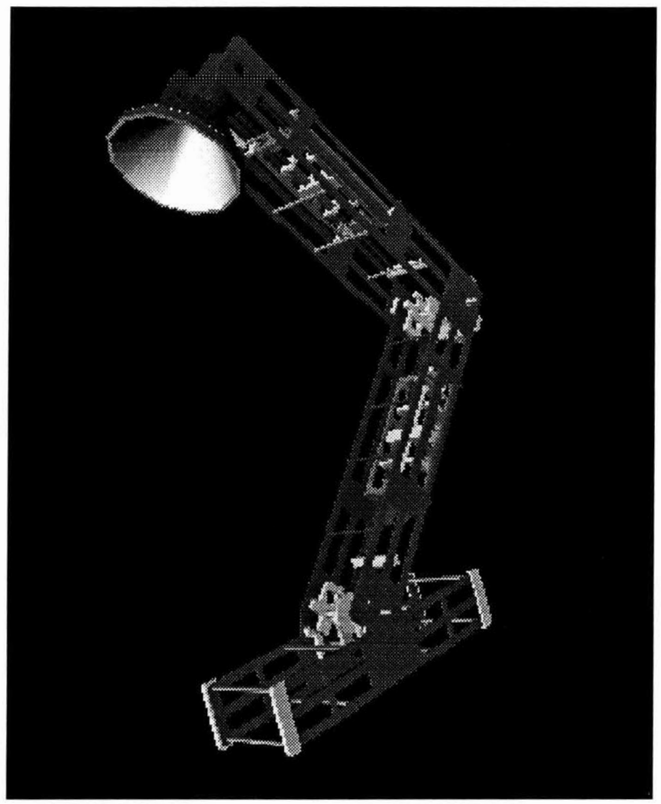


Figure 4-2: A PRO/Engineer assembly drawing of the lamp. The assembly incorporates all of the mechanical pieces of the lamp. The actuator assembly was modeled by Jerry Pratt.

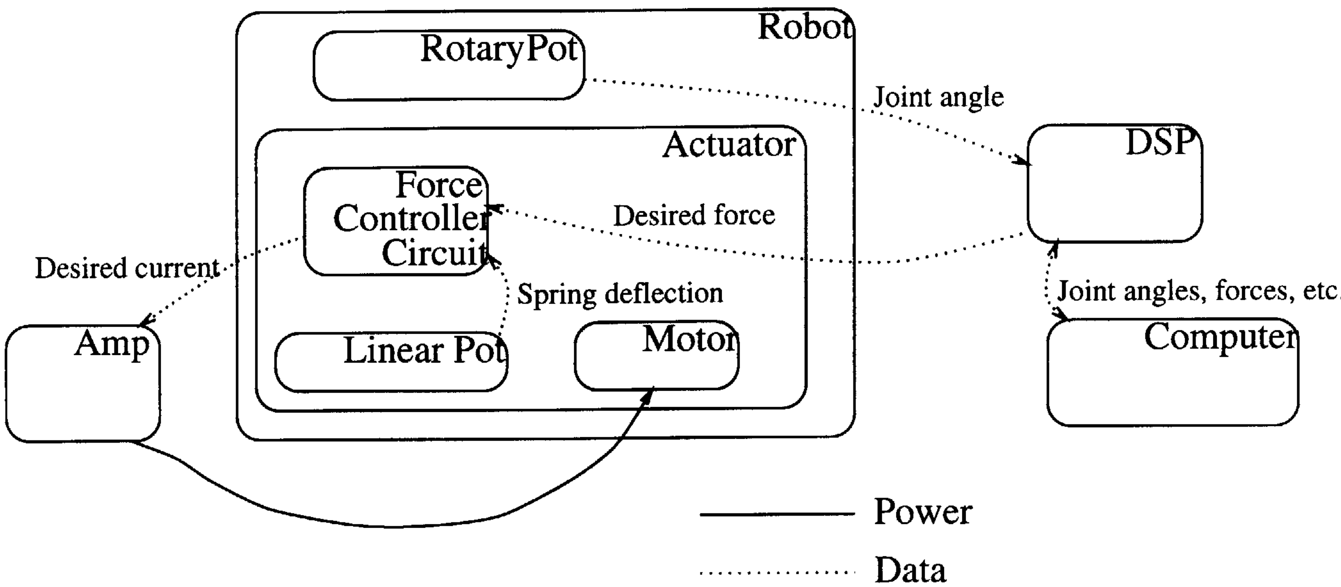


Figure 4-3: Electrical connections of data and power on physical robot.

4.3 Control Implementation

The control was developed through computer simulation as detailed in Chapter 3. The control software was able to be implemented using a DSP designed for the biped robot Spring Flamingo. The DSP interface software was written by Jerry Pratt and Robert Ringrose. The lines of communication and power between the DSP, the actuator controller circuits, and the computer can be seen in Figure 4-3.

Certain differences between the actual robot and the simulated robot affect the control implementation. One large difference is the lack of sensors on the actual robot. The actual robot is equipped with only 2 rotary potentiometers measuring joint angles. The simulation is aware of all the conditions of the lamps motion and this information was used to switch control states in the simulation. Some of the control state switching was done using the knowledge of parameters other than the motion of the two joints. Modifications to the control code must be made in order to account for the lack of sensing capabilities. There are several options for overcoming this lack of information. Two of the methods that will be implemented are listed below.

1. Run some simulations and record the times that the controller switches control states. Use these time as triggers for switching on the actual robot.
2. Use simple physics calculations to estimate the appropriate times for various states.

Both of the methods mentioned above will most definitely require iterative fine tuning.

4.4 Testing

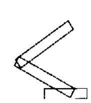
Once the lamp has been adapted, the physical model can be tested. The validity of the computer model, the controller robustness, and alternate control schemes can be addressed.

Chapter 5

Conclusions

A set of initial conditions, angular and linear momentum at takeoff, was established that lead to a successful jump. These conditions were successfully achieved within a physically realistic computer simulation using a simple virtual model controller. A physical model was designed and constructed in order to further test the control. Due to time constraints the evaluation of the physical model is incomplete. However, early analysis seems to suggest it will be suitable for testing the control developed through the simulation. Analysis of the lamp has definitely provided plenty of insight into the role of the ankle in a jump but there is more to be learned. The physical lamp should prove to be a useful tool in analyzing the problems further and testing possible control solutions.

Some of the future work will lie in experimentation using the physical model. The complexities of ankle motion presented in Chapter 2 should be able to be addressed in a variety of ways. There is also further work to be done through the computer simulation. The state space of possible control solutions is rather large and further research will hopefully shed more light into the subtleties of various approaches.



Bibliography

[1] John J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley, 1989.

[2] Michiel van de Panne Eugene Fiume and Zvonko Vranesic. Reusable motion synthesis using state-space controllers. *SIGGRAPH Proceedings*, 24(4), 1990.

[3] Cliff Frohlich. Do springboard divers violate angular momentum conservation? *American Journal of Physics*, 47(7), 1979.

[4] Michael Kass and Andrew Witkin. Spacetime constraints. *SIGGRAPH proceedings*, 22(4), 1988.

[5] Eadweard Muybridge. *The Human Figure in Motion.* Dover Publications, New York, 1955.

[6] Pixar. Luxo, jr. (Film), 1986.

[7] Gill A. Pratt and Matthew M. Williamson. Series elastic actuators. *IEEE International Conference on Intelligent Robots and Systems*, 1:399–406, 1995.

[8] Jerry E. Pratt. Virtual model control of a biped walking robot. Master's thesis, Massachusetts Institute of Technology, May 1995.

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