Transient Control in Bipedal Robot Walking: The "Start-off" Problem

2.165 Project
Spring 1997
David W. Robinson

Contents

List of Figures

Figure 1: Spring Flamingo: A 7-link planar bipedal robot currently under construction.

1 Introduction

In the early 1980's, a serious work began in the study of legged robots. There were two paths pursued, one with static robots (e.g. hexapod), and the other with dynamically balanced robots (e.g. biped). When walking with dynamic balance, the projected center of mass is allowed outside of the area inscribed by the feet, and the robot may essentially be falling during parts of the gait cycle. A foot must be moved so as to catch the body at the proper instant, breaking the fall and achieving a desired net translational acceleration or deceleration. Included with problems of balance, legged locomotion has may challenges such as nonlinear dynamics and closed-loop chains which lead to unsolvable closed form dynamic equations.

Most research in dynamic robot locomotion focus on the steady state motion (e.g. walking or running in a limit cycle). In, a made for TV special about the current state of the art in robotics, the host, Alan Alda [1], said the following about robots built by Mark Raibert [15],

> [These robots] must keep moving to balance...[they] can't simply stand still and need a lot of computing power to constantly figure out what it needs to do to stay on it's feet.

Most dynamically balanced robots cannot stand still. They have been specially designed to perform actions while in motion. In order to start motion, they need to be helped by wranglers (people who hold the robot steady until it is in a steady limit cycle.) To say it precisely, dynamically balanced robots cannot easily transition from a stopped standing position to their steady cycle of motion. In fact any transient change in state (starting, stopping, turning) is a problem. More about this will be discussed in previous work.

For this class project, I specifically address the "start-off" problem. This is done in a planar simulation of a 7 link bipedal robot. Originally, the intent was to implement these results on a physical robot called *Spring Flamingo* (Figure 1). Unfortunately, this robot has yet to become fully operational. Nevertheless, with the simulated robot, I accomplish three major goals:

- The robot stands still in a fully upright dynamically balanced position.
- The robot initiates a walk in one stride while maintaining a constant height above the ground.
- The robot does this at full leg extension for maximum mechanical advantage and walking efficency.

The major extension of this work is to add active feet and ankles to the robot. Just like biological creatures, this robot goes up on its toes to help lengthen out its stride while maintaining its vertical height (Figure 2).

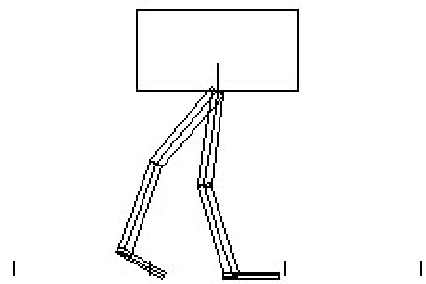

Figure 2: Simulation of a 7-link planar bipedal robot with active ankles toeing off.

The robot is over-seen by a state machine and the joint actuators are torque controlled. Since, the development of *Direct Drive Robots* [2] and in particular *Series Elastic Actuators* [13, 19] for non-colocated actuators, it is possible to assume that the command torques are actually received at the joints. With this assumption in place, then intuitive Cartesian control schemes such as *Transpose Jacobian Control* [3] and a major extension of that, *Virtual Model Control* [14] can be used to control non-linear multi body system. A force partitioning method [5, 14] is used to handle the constraints of the robot while it is in double support. The simulated robot can start from a standstill and begin walking. In fact, for two simulations shown, it enters a limit cycle after just one step. Not only is this work interesting, it is really fun too.

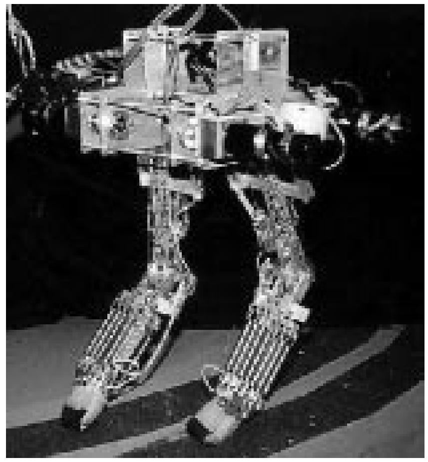

Figure 3: Spring Turkey: 1994-1996 Planar bipedal walking robot with point feet and Series Elastic Actuators

2 Previous Work

2.1 Virtual Model Control

Most of the results in the project are an expansion on the work of Pratt with virtual model control (VMC) [14]. Virtual model control uses simulations of virtual mechanical components to generate torques in a robot and thus give the robot the look and feel that it is actually connected to a real component. For example, a Cartesian PD control of the endpoint of a multi body robot arm could be thought of as a virtual spring/damper control system. It is stable because if a read spring and damper were connected to the robot the robot would come to a steady resting position. This has been shown by Slotine [17] for PD control on the joints of the robot using Lyapunov analysis. VMC is similarly stable in the sense of Lyapunov as shown by Pratt [14].

2.2 Starting and Stopping

Starting and stopping have been addressed in a couple of manners. Miller [12, 9], who uses CMAC neural network to control a 3d bipedal robot, lets the robot learn to rock side to side and slowly starts increasing forward progress. The robot can take strides up to 6cm after increasing stride length 1cm at a time. Kajita, et.al. [7, 8], developed a concept of orbital energy that allows the robot to judge whether or not it has enough energy to get over the top of an inverted pendulum walking model. Their robot stands in a split stance and lifts up it front leg to initiate a forward force. This was a similar method tried by Pratt with his robot *Spring Turkey* (Figure 3) but it still needed a wrangler to steady the first few steps [14].

While not addressed directly in any publication, Fujimoto and Kawamura [4] have a simulation that begins from a stop to a walk.

2.3 Natural Dynamics and Limit Cycles

All robots have their own dynamics. Walking and running can be shown to be completely passive under certain conditions [10, 11]. These unpowered mechanical biped robots can walk down an inclined plane with a steady gait. The characteristics of the gait depend only on the geometry and the enertial properties of the robot and the slope of the plane. Goswami [6] has tried to anaylize the limit cycle of a double pendulum walker and has proposed ways of posibly enlarging the basin of attraction of the passive limit cycle. Since walking is passive, it might be possible to prove using Lyapunov theory [17] or Contraction Analysis [16] that a multi-link robot walker tends to a limit cycle from a transient disturbance such as starting.

3 Bipedal Walking

Many of the ideas in this section have been adapted from the work done by Pratt [14]. As mentioned before, he successfully made a 5-link planar bipedal robot with point feet walk.

3.1 Single Support with Ankle and Foot

As described in many robotic text books, a Jacobian J can relate the joint velocity of a robotic arm to the cartesian velocity at the end point. By assuming no slip and that the projected zero moment point of a robot leg lies within the area of the foot, a Jacobian can be written from the ankle to the hip.

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

where a, b, c, and d are trigonometric functions of the link lengths and joint angles. The relationship between cartesian velocities and joint velocities can be written as

$$\vec{x} = \begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\theta}_{body} \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_{ankle} \\ \dot{\theta}_{knee} \\ \dot{\theta}_{hip} \end{bmatrix} = J\vec{\theta}$$

The Jacobian transpose then relates forces in the cartisean frame to joint torques in the joint frame.

$$\vec{\tau} = J^T \vec{f}$$

With this relationship, any desired force vector applied at the hip can be realized as joint torques given the assumptions mentioned above.

3.2 Single Support Up On Toe

The previous section assumed that the robot foot would be flat on the ground. In part of the walk however, the robot will go up on its toe. This creates a new Jacobian

$$J = \begin{bmatrix} z & a & b & 0 \\ x & c & d & 0 \\ -1 & -1 & -1 & -1 \end{bmatrix}$$

The new elements in the Jacobian come from the fact that the reference frame on the ground is now measured from the toe rather than ankle. In addition to the new Jacobian, we have an additional constraint. Since the toe is not actuated, there is a zero moment constraint at the toe.

$$\begin{bmatrix} 0 \\ \tau_{ankle} \\ \tau_{knee} \\ \tau_{hip} \end{bmatrix} = \begin{bmatrix} z & x & -1 \\ a & c & -1 \\ b & d & -1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_x \\ f_z \\ f_\theta \end{bmatrix}$$

It is possible to assume that the forces in the vertical and rotational direction are more important then the horizontal force. Therefore specifying f_z and f_θ solving for f_x would give

$$f_x = \frac{f_\theta - x f_z}{z}$$

and plugging this back into our force/torque relationship

$$\begin{bmatrix} \tau_{ankle} \\ \tau_{knee} \\ \tau_{hip} \end{bmatrix} = \begin{bmatrix} \frac{zc-ax}{z} & \frac{a}{z} - 1 \\ \frac{zd-bx}{z} & \frac{b}{z} - 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} f_z \\ f_\theta \end{bmatrix}$$

We now have a simple method for determining joint torques for a given desired force at the body. If however, losing complete control over the force in the horizontal direction is undesirable, then defining a constraint matrix for a given desired force vector $\vec{f}$ creates a new force vector $\vec{f^*}$ such that

$$\vec{f} = C\vec{f^*}$$

where C is defined as

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ z & x & -1 \end{bmatrix}$$

(An extra zero must be added to the vector $\vec{f}$.)

By taking the pseudo-inverse of C [18], $\vec{f^*}$ will become the least squares fit of $\vec{f}$ due to the constraint $zf_x + xf_z - 1 = 0$.

$$\vec{f^*} = (C^T C)^{-1} C^T \vec{f}$$

$$\vec{f^*} = C^{\#} \vec{f}$$

where

$$C^{\#} = \begin{bmatrix} \dfrac{2+x^2}{2+x^2+z^2} & -\dfrac{z\,x}{2+x^2+z^2} & \dfrac{z}{2+x^2+z^2} \\ -\dfrac{z\,x}{2+x^2+z^2} & \dfrac{2+z^2}{2+x^2+z^2} & \dfrac{x}{2+x^2+z^2} \\ \dfrac{z}{2+x^2+z^2} & \dfrac{x}{2+x^2+z^2} & \dfrac{1+x^2+z^2}{2+x^2+z^2} \end{bmatrix}$$

The joint torques $\vec{\tau}$ are now defined by

$$\vec{\tau} = J^T \vec{f^*} = J^T C^{\#} \vec{f}$$

This is solved symbolically using maple and used in the control code.

Another method for solving the zero moment point at the toe that should be explored is Lagrange multipliers.

3.3 Double Support

Approximately 20-25% of the stance cycle is in double support. Using a force partitioning method [5, 14], the force vector specified at the body can be split to the two legs. We know that

$$\vec{f} = \vec{f_l} + \vec{f_r}$$

Since there are six joints and only three controlled directions of force, three design constraints are needed. Many different constraints can be chosen but I choose the following:

- The back foot is on its toe so there is no torque at the toe.
- Let the front foot have no torque at the ankle.
- Match the hip torques.

This will give a relationship

$$\begin{bmatrix} \vec{f} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ z & x & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & e & g & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \vec{f_l} \\ \vec{f_r} \end{bmatrix}$$

The fourth, fifth, and sixth rows correspond with the three design constraints respectively. The z, x, and -1 correspond to the top row of J^T for a leg described in the second section. The e, g, and -1 correspond to the top row of J^T for a leg as mentioned in the first section only it is using different variables to keep things separate. The last row matches the hip torques.

Inverting this matrix defines the force partition.

$$\begin{bmatrix} \vec{f_l} \\ \vec{f_r} \end{bmatrix} = \begin{bmatrix} -\frac{e\,x}{-e\,x+z\,g} & -\frac{g\,x}{-e\,x+z\,g} & \frac{1}{2}\frac{g+x}{-e\,x+z\,g} \\ \frac{e\,z}{-e\,x+z\,g} & \frac{z\,g}{-e\,x+z\,g} & -\frac{1}{2}\frac{z+e}{-e\,x+z\,g} \\ 0 & 0 & \frac{1}{2} \\ \frac{z\,g}{-e\,x+z\,g} & \frac{g\,x}{-e\,x+z\,g} & -\frac{1}{2}\frac{g+x}{-e\,x+z\,g} \\ -\frac{e\,z}{-e\,x+z\,g} & -\frac{e\,x}{-e\,x+z\,g} & \frac{1}{2}\frac{z+e}{-e\,x+z\,g} \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \vec{f}$$

These respective forces can now be multiplied by their Jacobians to specify each joint torque in the robot.

3.4 Stopped Double Support

The force partition just described only works for the robot while it is walking. In order for it to stand stopped with its feet together another partition must be defined. Since our robot is standing still with its feet together, by requiring the torques in each leg to be identical then our robot becomes like a single leg described in the first section. The matrix inversion, as expected, divides the required forces by two and gives each leg one half.

3.5 Starting

The maximum velocity attainable in one step of the robot from a stopped position can be estimated by developing a very simple model. Let the robot be an inertia (m) with mass less legs that travels only in the horizontal or x direction. We know that by conservation of energy

$$\frac{1}{2}mv^2 = \int_0^x f_x dx$$

Assuming that the robot starts off on its toe (try it, people start off on their toes too), then we know from the zero moment constraint that

$$af_x + cf_z - f_\theta = 0$$

By inspection of the formulation of a and c, we find that they represent respectively minus the standing height, $-z$, and forward distance traveled, x, with respect to the toe. Also, for steady walking we know that $f_\theta/(af_x+cf_z) << 1$ and that $f_z \approx mg$. Therefore, assuming a constant standing height, $\bar{z}$,substituting back into the zero moment constraint and solving for f_x gives

$$f_x \approx \frac{xmg}{\bar{z}}$$

Substituting this result into the energy equation gives,

$$\frac{1}{2}mv^2 = \int_0^x \frac{xmg}{\bar{z}} dx = \frac{mg}{2\bar{z}}x^2$$

Solving for v,

$$v_{max} \approx x\sqrt{\frac{g}{\bar{z}}}$$

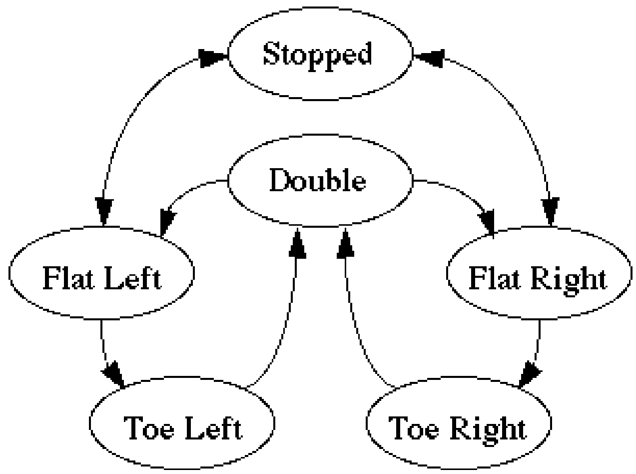

Figure 4: Walking State Machine: Overview of the different states for a walking bipedal robot.

By estimating values for the variable that are typical to my robot, $x = 0.15m$, $\bar{z} = 0.65m$, I get $v_{max} > 0.5$ m/sec. 0.5 m/sec is my desired walking speed so in one step the robot can reach steady walking speed. Then it is up to the steady state walking algorithm to stabilize into a walking limit cycle.

3.6 State Machine

The controller for this robot is a six state machine. The states are

- Stopped.
- Double.
- Flat Right.
- Toe Right.
- Flat Left.
- Toe Left.

The robot begins in the Stopped state. Then went it starts, it moves to Flat Right or Flat Left. The robot changes state as it walks. It changes state due to positions of the body with respect to the feet or due to foot switches contacting the ground. Figure 4, shows the relationship between the states. Notice that the stopped state goes directly to the flat right or flat left to enter the walking limit cycle. The flat states are also the states from which the robot returns to the stop state.

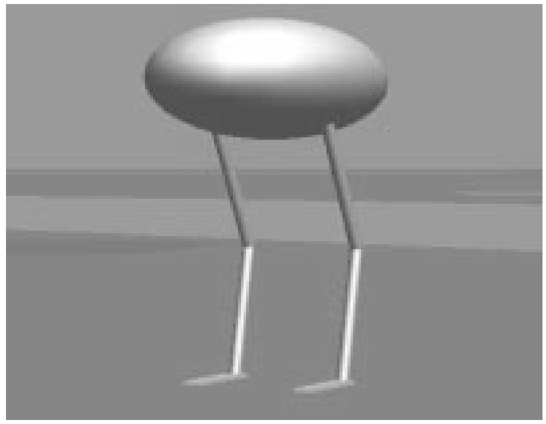

Figure 5: Picture of the simulated robot stopped and ready to begin walk.

4 Results

The parameters of the simulated robot are as follows:

- Body Mass 7kg.
- Leg Link Mass 0.5 kg.
- Foot Mass 0.15 kg.
- Leg Link Length 0.35 m.
- Foot Length 0.17 m.

The robot can stand at a maximum height of 0.69 m because of knee joint stops which limit the leg from extending completely. There are no other joint stops and hips and ankles can rotate freely. The lengths and masses are approximately the same as a ratite called a Rhea.

The simulation is done in an environment called the creature library developed at the MIT Leg Lab. The creature library is a front end to a commercial dynamics simulation software package called SDfast. The stick figure simulations are displayed in Leg Plot and the video is created with a program called Anim (figure 5) both of which were developed at the MIT Leg Lab.

4.1 Standing

Standing with zero velocity is no difficulty for the robot. Since there is no velocity it is like a robot arm rigidly attached to the ground. It can go onto one foot and move that foot around. It is also possible to move the projection of the robot's center of mass all around the footprint.

The simulation is able to go almost to a full upright position with the legs straight. Like in biological creatures, I put a nonlinear joint stop (virtual kneecap) so that the knee will not hyper-extend. Because of that, it never reaches the singularity at full extension but it does come very close. Since each of the walking simulations started from this stopped position more will be mentioned about it later in the next section.

4.2 Starting

Figure 6 shows one of the results of the start off problem.

The simulation is 22 seconds long. The first two seconds are with the robot standing with zero velocity. Then it starts to walk for the next 20 seconds. This is the same for each of the simulations results shown (figures 7,9).

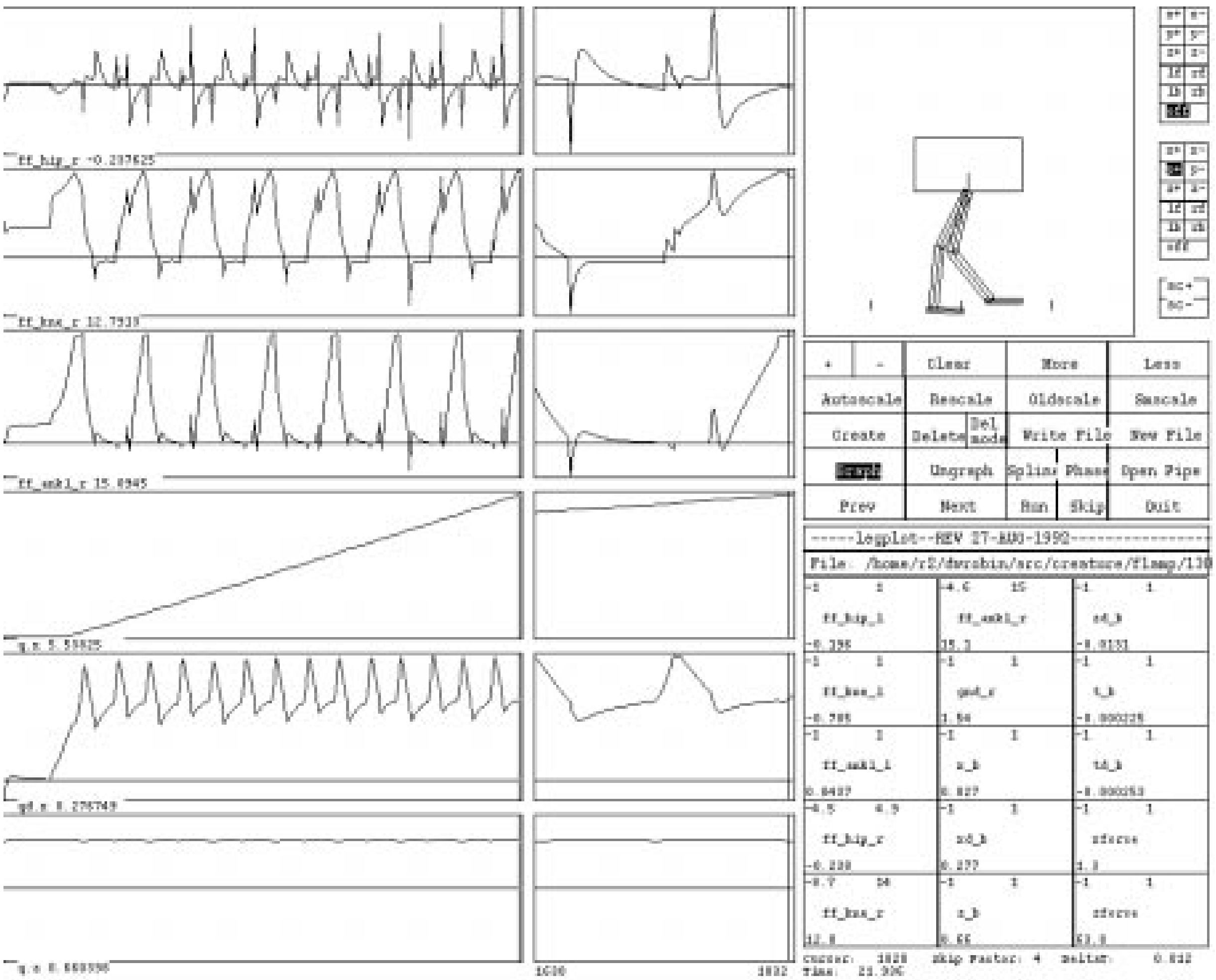

Figure 6: 22 seconds of simulation results of a start to steady state walking using horizontal velocity control during flat and double support with a gain of 7Ns/m.

Figure 6 uses the toe controller with no f_x control while figures 7 and 9 show results from the pseudo f_x control for the toe state.

The figures are a screen dump of the simulations. On the left side of the figures, six variables are displayed. The top three are the joint torques for the three joints in the right leg: hip, knee, and ankle. The next two are the position and velocity of the robot. The bottom one is the height of the robot above the ground.

For the x-direction, proportional velocity control is used during walking and proportional position control is used in the stopped state. In the first simulation (figure 6), the average velocity is 0.275 m/s while the command velocity is 0.5 m/s. This velocity and gain are very comfortable for the robot in simulation. However, if either the velocity or gain are increased too much more the robot will eventually fall over its front. The limitation is due to the speed at which the swing leg can move forward ready to take the next step. Since in biological creatures, the swing leg motion is mostly passive, making it swing forward too fast would make the robot look unnatural.

The average speed of the simulation in figure 7 is 0.214 m/s. In order for this robot to maintain stability and enter the walking limit cycle, the proportional velocity gain had to be reduced by almost 30%. The simulation in figure 9 show the results if the gain is only reduced by 15%. It is easy to see that for a few steps the limit cycle seems to have been reached with and average velocity of 0.34 m/s. However, right at the end of the simulation the speed of the robot starts increasing exponentially. If the simulation was allowed to finish the robot would have fallen over. Once again this is a swing leg problem. The swing leg cannot get out in front far enough to help control the next step. Perhaps the gain of the swing leg and the velocity gain should be coupled. When one is increased the other should as well.

In all three simulations shown, it is easy to see that the height of the robot did not change much at all during walking. In fact a close examination of the walking height will show that it doesn't vary more than 1.5 cm from the nominal. In figure 8, the robot is in full stride. The ankle definitely helps to maintain the nominal height above the ground.

Another point of interest is the periodic nature of the torques seen in the simulations. As a minimum a frequency analysis of the torque signals should be done to make sure that the simulated torques are well within the 20Hz bandwidth of the robot *Spring Flamingo* (figure 1).

Since the torque signals are periodic, this gives the impression that it may be possible to put a pattern generator into the robot control structure that mimics the natural dynamics of the system. It may also be possible to do learning because walking is so periodic that in each stride could be considered a learning cycle. This is left for future work.

In a comparison of the two toe control methods, the pseudo control seems not to work as well as other. I do not have an answer for this. I thought that it would work better. It should be noted however, that the *tweaking* in these simulations was not exhaustive. It should be possible to use something such as a genetic algorithm or other optimization tool to better tune the parameters of this simulation. In doing so, I feel that the pseudo controller will perform better.

4.3 Stopping

Stopping is not part of the original plan for the project but it was easy to try. Unfortunately, the short time spent in trying to get the robot to stop was unsuccessful. There didn't seem to be enough time after entering one of the flat states to provide the necessary backward stopping force. The robot would just topple over because of the residual forward velocity. I feel confident that with a little more work though the robot would be able to stop. Perhaps if in the previous double support state it would start slowing down, then the flat state would be able to take it to a full stop. This is left for future work.

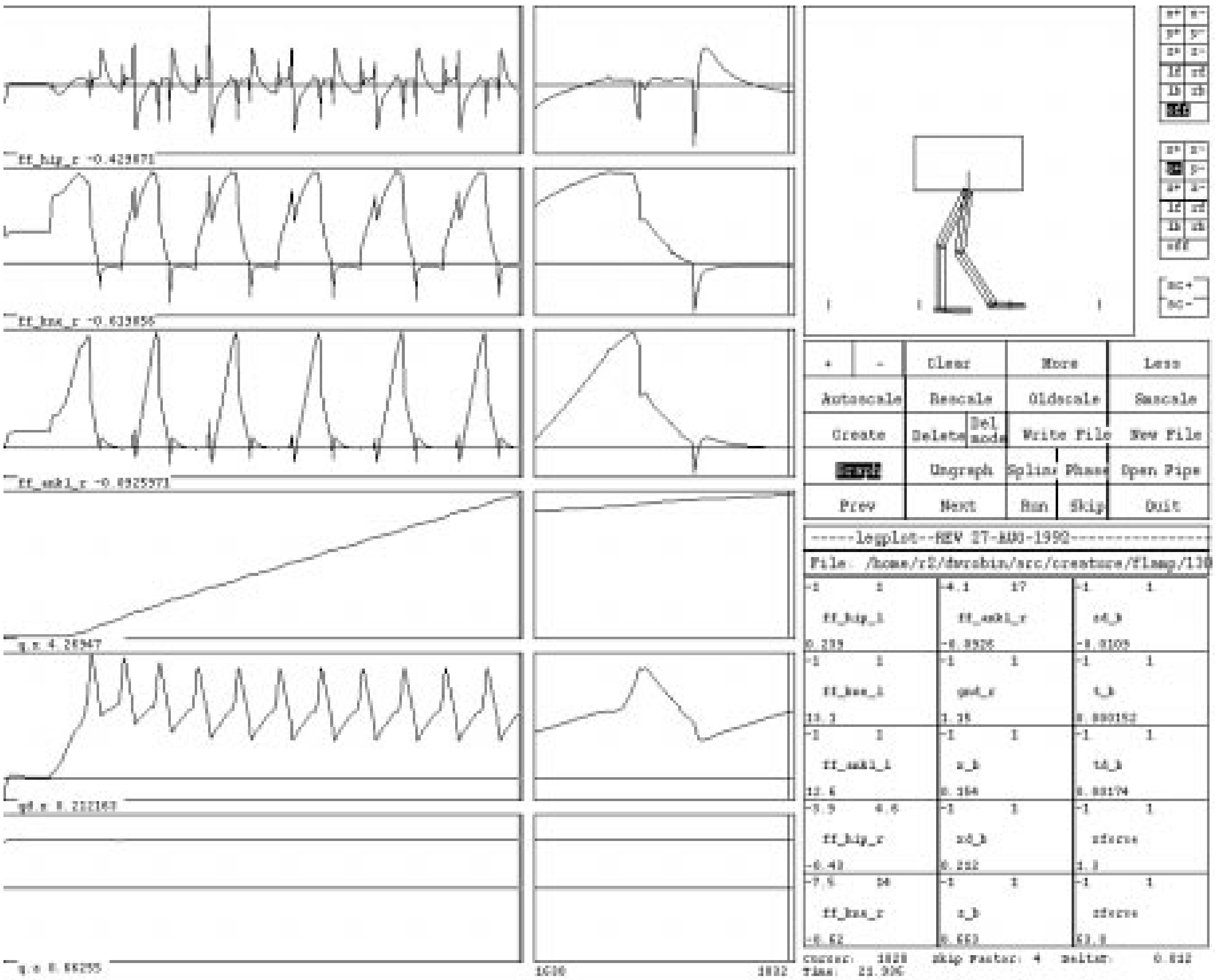

Figure 7: 22 seconds of simulation results of a start to steady state walking using horizontal velocity control during flat and double support and pseudo control during toe support with a gain of 5Ns/m.

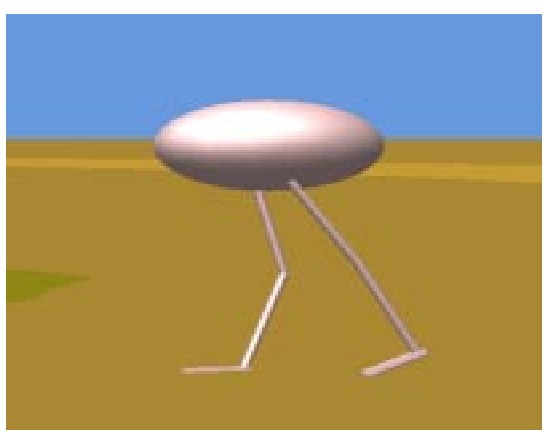

Figure 8: Picture of the simulated robot in full stride.

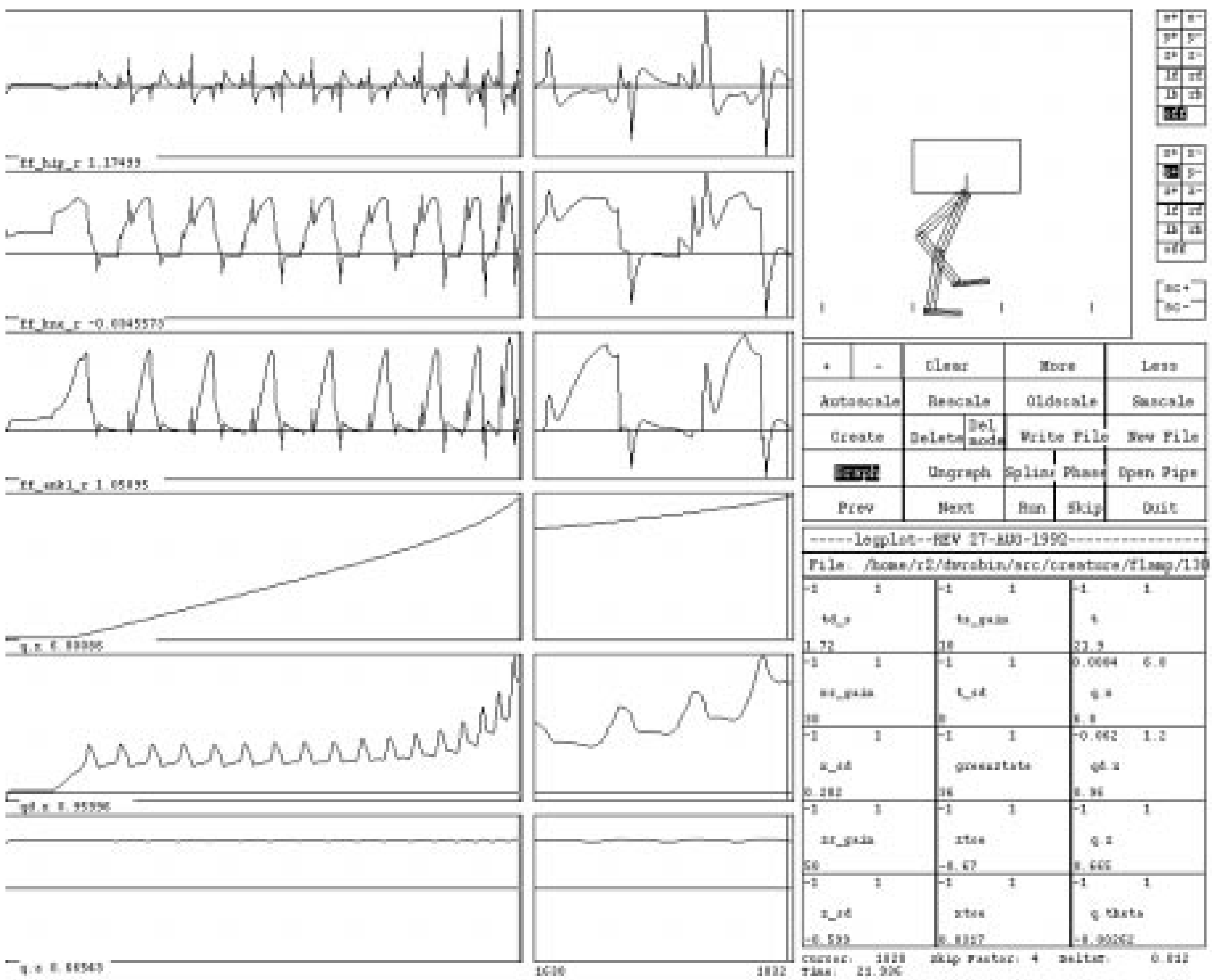

Figure 9: 22 seconds of simulation results of a start to unstable steady state walking using horizontal velocity control during flat and double support and pseudo control during toe support with a gain of 6Ns/m. Notice the increasing velocity at the end of the simulation. In a few more steps the robot will fall over forward.

5 Conclusions and Future Work

As mentioned in the introduction, the simulated robot can walk. It can start from a dead stop and start into a walking limit cycle in one stride. The robot maintains a steady height above the ground for each stride. The main reason for this is the addition of an active ankle and foot to the two link leg of the robot.

Future work may include:

- Characterization of the joint torques and incorporate them as a pattern generator.
- Use optimal engineering techniques to increase the walking speed of the robot.
- Use contraction analysis to develop a concrete theory of stability.
- *This is the most important* Incorporate the theory and tools developed here and implement it all on a real walking robot.

References

[1] Alan Alda. *Robots Alive* Discovery Channel, 1997.

[2] Haruhiko Asada, Kamal Youcef-Toumi. *Direct Drive Robots. Theory and Practice.*, The MIT Press, 1987.

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

[4] Yasutaka Fujimoto, Atsuo Kawamura. Proposal of Biped Walking Control Based on Robust Hybrid Position/Force Control. *IEEE International Conference on Robotics and Automation, 1996*

[5] J. Gardner, K. Srinivasan, K. Waldron. A Solution for the force Distribution Problem in Redundantly Actuated Closed Kinematic Chains. *Journal of Dynamic Systems, Measurement, and Control. Vol.112, September 1990.*

[6] A. Goswami, B. Espiau, A. Keramane. Limit Cycles and Their Stability in a Passive Bipedal Gait. *IEEE International Conference on Robotics and Automation, 1996*

[7] S. Kajita, T. Yamaura, A. Kobayashi. Dynamic Walking Control of a Biped Robot Along a Potential Energy Conserving Orbit. *IEEE International Conference on Robotics and Automation, 1992*

[8] Shuuji Kajita, Kazuo Tani. Experimental Study of Biped Dynamic Walking in the Linear Inverted Pendulum Mode. *IEEE International Conference on Robotics and Automation, 1995*

[9] Andrija Kun, W. Thomas Miller, III. Adaptive Dynamic Balance of a Biped Robot Using Neural Networks. *IEEE International Conference on Robotics and Automation, 1996*

[10] Tad McGeer. Passive Bipedal Running. *Simon Fraser University, Centre for Systems Science, CSSS-IS TR 89-02*, 1989.

[11] Tad McGeer. Passive Bipedal Walking. *The International Journal of Robotics Research, Vol.9, No.2, April 1990.*,

[12] W. Thomas Miller, III. Real-Time Neural Network Control of a Biped Walking Robot. *IEEE Control Systems, February 1994.*

[13] Gill A. Pratt, Matthew M. Williamson. Series Elastic Actuators. *IEEE International Conference on Intelligent Robots and Systems,1995*

[14] Jerry Pratt. Virtual Model Control of a Bipedal Walking Robot. *M.S. Thesis, MIT Department of Electrical Engineering and Computer Science*, 1995.

[15] Marc H. Raibert. *Legged Robots That Balance.*, The MIT Press, 1986.

[16] Mechanical Engineering 2.165, Applied Nonlinear Control. *Massachusetts Institute of Technology.*, Spring 1997

[17] Jean-Jacques E. Slotine, Weiping Li *Applied Nonlinear Control.*, Prentice Hall, 1991.

[18] Gilbert Strang *Introduction to Applied Mathematics.*, Wellesley-Cambridge Press, 1986.

[19] Matthew M. Williamson. Series Elastic Actuators. *M.S. Thesis, MIT Department of Electrical Engineering and Computer Science*, 1995.