Design and construction of the Cornell Ranger, a world record distance walking robot.

Internship at Cornell University

J.G.Daniël Karssen
j.g.d.karssen@student.tudelft.nl

Cornell University
Department of Theoretical and Applied Mechanics
Biorobotics and Locomotion Lab
Ithaca, United States

Delft University of Technology
Faculty of Mechanical, Maritime and Materials Engineering
Section BioMechanical Engineering
Delft Biorobotics Lab
Delft, The Netherlands

version 2.2

Internship: Fall 2006
Report: January 26, 2007

Foreword

This report discusses my internship at Cornell University. The report has two purposes. The first is to inform the reader about the activities I have done during my internship. The second purpose is to be a guideline for people who will continue to work on the research project I worked on.

The project goal was to build a walking robot. People interested in the mechanical design of the robot can find this in chapter 2. The controller for this robot is explained in chapter 3.

I started the project as team leader of the mechanical team with the responsibility for the mechanical design, the fabrication and the assembly. For the fabrication and the assembly I had the help of 5 undergraduates who worked for about 10 hours a week on this project. After the mechanical part of the robot was finished I did the wiring of the robot and worked on the software. The main things I did on the software are the design and implementation of the walk controller and the torque controller.

On this project I worked with a lot of people. I would like to thank Professor Andy Ruina, lab manager Jason Cortell and all the other people for there support during this project. A complete list of people who worked on this project can be found in the acknowledgements (chapter 7).

Contents

Chapter 1

Introduction

Two-legged walking robots (bipeds) are an interesting research topic. The purposes of this research are to increase our understanding of human walking and to gain knowledge for building service and entertainment robots. In general there are two ways to control a biped. One way is to look at the biped as a static system and walk by balancing on one foot while placing the other foot in front. The other way is to look at the biped as a dynamical system and walk by falling forward, landing on the next foot and continue falling forward standing on this foot. This last method is called dynamic walking and has two advantages compared with the first method. These advantages are: high energy efficiency and human like movement.

The first practical demonstration of dynamic walking is the prototype built by McGeer[1]. This prototype is a passive dynamic walker, which means that the walker is completely actuated by gravity. Cornell University and Delft University of Technology have also built a couple of prototypes [2, 3, 4, 5]. One of the Cornell prototypes showed the good energy efficiency of dynamic walking by walking with only 11 watt [3]. This energy consumption is similar to humans when scaled with the mass and speed.

All the prototypes show that dynamic walking works in practice, but it has never been shown that this method is reliable for walking a long time. The maximal distance walked by one of the prototypes is approximately fifty meters. The goal of this project is to build a prototype that shows that a biped can walk reliable with the dynamic walking method, by walking fully autonomous more than one kilometer without falling. To do this the robot should have at least the following properties:

- reliable mechanics and electronics, so that the robot does not break down al the time,
- energy efficient movements, so that the whole distance can be walked with one

set of batteries,

- a way of steering, to not be dependent on a long straight surface.

This report presents the design of the robot. Chapter 2 shows the mechanical design of the robot and in chapter 3 the controller is discussed. The results of this project are shown in the next chapter (chapter 4). These results and future work are discussed in chapter 5. And finally in chapter 6 the conclusions are given.

Chapter 2

Mechanical design

The main principle used by the mechanical design of the robot is: keep it simple. The idea is to reduce the chance of failures by keeping the design as simple as possible. The 'keep it simple'-principal can be seen in the limited amount of degrees of freedom and actuators.

2.1 General lay-out

Figure 2.1 shows a CAD-drawing of the robot. The robot has a mass of 5 kilograms and a leg length of 1 meter measured from the bottom of the foot to the hip. There are three degrees of freedom, one at the hip and two at the ankles. Each of these degrees of freedom is actuated with a separate dc-motor.

The robot is a so called four-legged biped. This means that the robot has two pairs of legs, an inner and an outer pair. The legs of each pair are rigidly attached to each other and both pairs are connected at the hip with a hinge. Due to the four legs the motions of the robot are constrained to a 2d-plane.

The motors and the other heavy parts (i.e. batteries) are located around the hip, in order to get the center of mass of the legs as close as possible to the hip. This is done for two reasons: (1) to minimize the inertia of the legs with respect to the hip joint and (2) to get good symmetry between the two pairs of legs.

The low inertia of the legs (1) is important for the energy efficiency, because the hip force needed to accelerate the swing leg is almost linear with the inertia around the hip. Also a low inertia of the swing leg is useful for the control, because with this low inertia the movements of the swing leg do almost not influence the motion of the stance

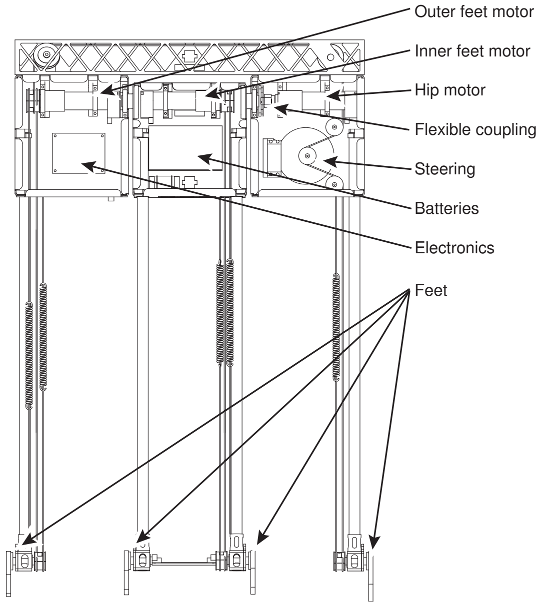

Figure 2.1: CAD-drawing of the Cornell Ranger and its most important parts

leg. And with this low influence between the legs the swing can be controlled without disturbing the motion of the stance leg.

The symmetry meant in (2) is the dynamical symmetry. The legs are dynamical symmetric if they move the same when the same force is applied to both and if the complete robot will move the same irrespective of which leg is in front. In order to be symmetric the legs should have the following properties:

$$I_{outer,hip} = I_{inner,hip} \tag{2.1}$$

$$m_{outer}c_{outer} = m_{inner}c_{inner} \tag{2.2}$$

$$m_{outer}d_{outer} = m_{inner}d_{inner} \tag{2.3}$$

in which I_{hip} is the inertia around the hip, m is the mass, c is the vertical distance between center of mass and hip and d is the horizontal distance between center of mass and hip. For these equations the feet and legs are assumed to be one rigid body, neglecting the effect of the movement in the ankle joint. The equations show that placing point masses at the hip makes no difference for the symmetry, so that is why all of the heavy parts are placed as close to the hip as possible.

2.2 Hip and feet actuation

The hip and both pair of feet are actuated with 46 watt dc-motors (Faulhaber #2657CR012). In section 3.4.2 you can find motor test results of these motors. The hip motor is equipped with a 1:66 planetary gearbox (Faulhaber series 30/1) and both feet motors have a 1:14 planetary gearbox (Faulhaber series 30/1). The hip motor is located in the outer leg and is attached to the inner leg with a flexible coupling. This flexible coupling allows for some misalignment between the hip bearing mounts and the motor mount.

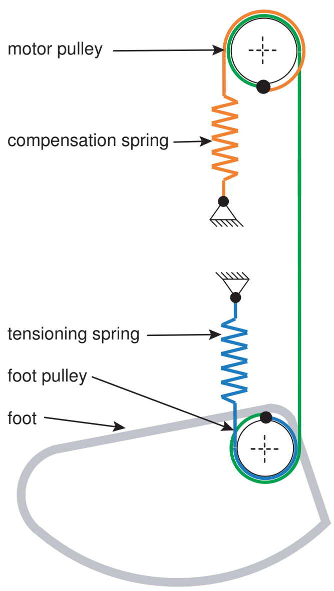

Figure 2.2: Cable system used to actuate the feet

The feet are connected with a cable system to the motors (figure 2.2). In this cable system there is a parallel spring to keep tension on the cable and another parallel spring to reduce the spring force on the motor. The springs have a low spring constant and a high pretension, so that they behave almost like constant force springs. Therefore the spring force on the motor is minimal.

The inner feet are rigidly connected so that they can be actuated by one cable. For the outer feet two cables are necessary to actuate the feet (figure 2.3). These two cables make it possible to change the angle of the feet separately, which can be used for steering (section 2.3).

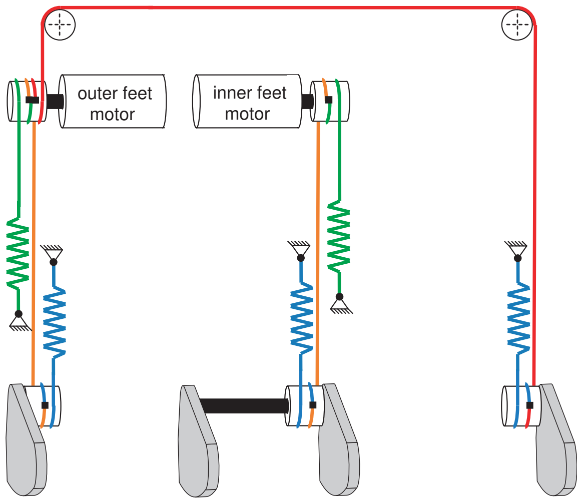

Figure 2.3: Cable system for the inner and outer pair of feet

Pulleys are used to connect the cables to the motors. These pulleys are made of two halves and clamp on to the shafts of the gearbox (figure 2.4). One of the halves has a flat spot to lock the D-shape shafts. There are two clamps on the pulley for clamping the cables. To reduce the force on the clamps the cables are wrapped a couple of times around to pulley. The pulley has a cut out so that it can be placed close to the gearbox to reduce the moment load on the bearings of the gearbox.

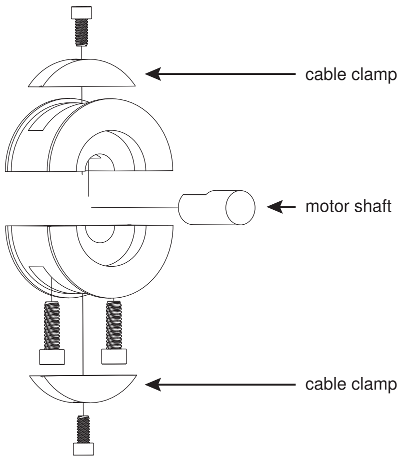

Figure 2.4: Motor pulley

2.3 Steering

Steering for the robot is necessary in order to walk a long distance without a long straight track. The simplest way of making the robot steer is to make an asymmetry in the feet actuation. This can be done for the outer feet by changing the length of one of the cables between the motor pulley and a foot pulley (red or orange cable in figure 2.3).

There are two important requirements for the steering: (1) the control of the steering should be separated from the main computer and (2) the power consumption should be low (maximal 1 watt continues). The first requirement can be meet by using an rc-controller. An rc-system has also the advantage of being reliable and easy to get, because it is used a lot in model planes and model cars. To get low power consumption (2) a non back drivable can be used, so that only energy is required to change the steering angle and not for holding it in place.

Using the two ideas above the concept of figure 2.5 was selected. This concept uses two gears. A small gear is mounted on an rc-servo motor and a large gear is mounted on the robot. On the large gear a pulley is eccentrically mounted, so that by rotating the large gear the length of the cable can be adjusted. The gear ratio between the large and the small gear makes the system non back drivable.

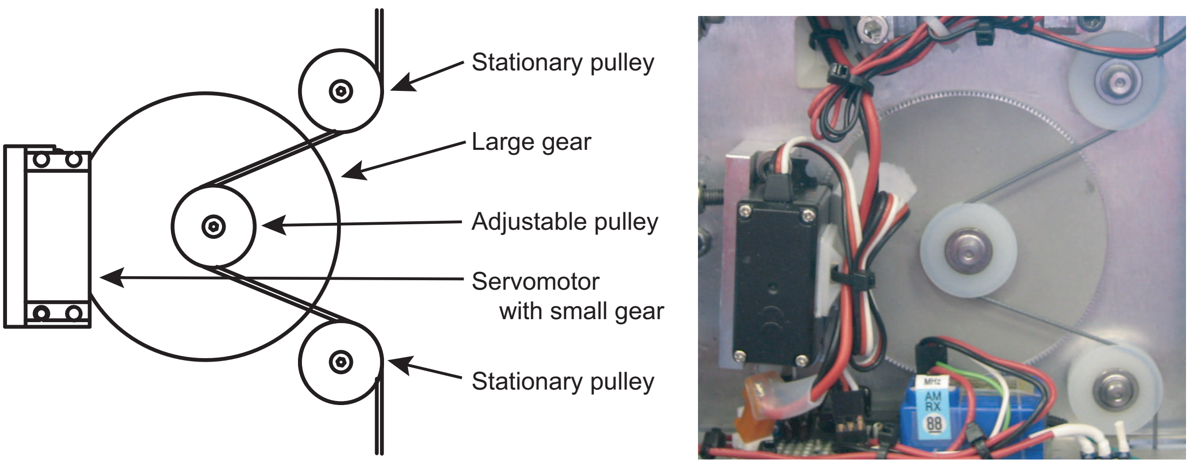

Figure 2.5: Steering system

2.4 Ankle joint

The ankle joint for this robot was a challenging part to design, because it has a tight weight limit (maximal 100 gram) and it has a complex function. The ankle has the following functions:

- a rotational joint between the leg tube and the foot,
- a stop to limit the motion of the foot,
- a guide for the sensor wire from the foot to the leg.

Figure 2.6 shows a half section of the ankle design. The shaft is parietal hollow and has a opening in the middle so that the sensor wire can go from the foot into the hollow leg tub. The sensor wire is wrapped around the shaft one time to allow for rotation of the shaft. The shaft is supported by two flanged bearings. These bearings are flanged to reduce the number of parts needed for holding the bearings. Attached to the shaft is a pulley for the actuation of foot. This pulley has the same design as the pulleys at the motors (section 2.2). A shoulder bolt is used as a stop for the foot to prevent it from rotating more than one rotation.

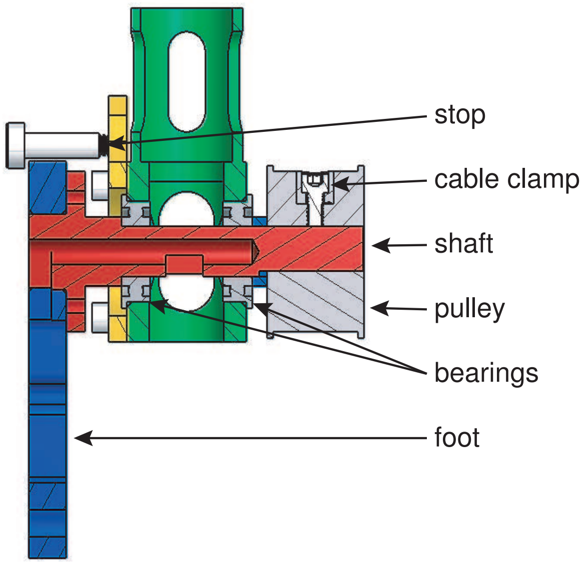

Figure 2.6: Cross-section of the ankle

2.5 Supporting structure

The outer legs are constructed out of two sheet metal boxes and a top bar connecting the two boxes. The inner leg is constructed out of one box (figure 2.7). Boxes are used because they are light weight and relatively stiff. There is only one problem with using boxes: they lose their strength when the cover is removed. This can cause problems during maintenance for which the covers have to be removed. To reduce this problem an extra cover is placed over the motor in the inner leg to increase the strength of the inner leg box. For the outer leg this is a smaller problem because the top bar gives these boxes extra strength.

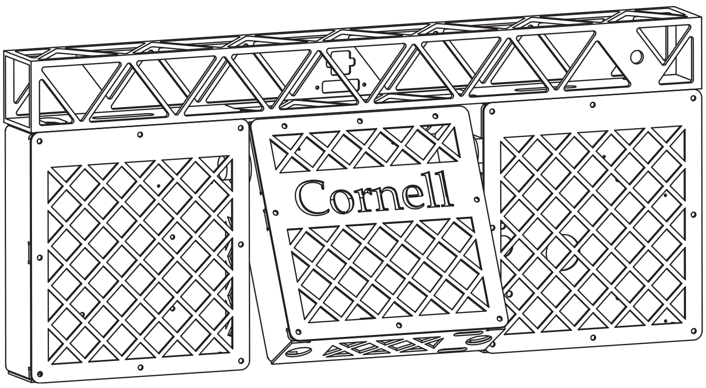

Figure 2.7: Supporting structure made of three boxes and a top bar

Chapter 3

Controller

In order to let the robot walk it has to have a controller telling what to do when. This controller has as inputs the sensors and as outputs the motor controllers. The controller can use the following sensors:

- encoders on all three motors,
- gyroscopes for all three axis,
- linear accelerometers in all three axis,
- contact sensor in each foot,
- battery voltage sensor,
- current sensors for battery current and for motor currents.

The controller is divided in several functions in order to make each function more clear. In this chapter the different functions are discussed and as a start an overview of all the functions is given.

3.1 Overview of the controller

Figure 3.1 gives a block scheme of the functions of the controller and how they interact. The signal processing blocks translate the output of the AD-converters to usable sensor readings, such as rad/s for the gyroscopes. These sensor readings are the input of the state estimator which gives as output the estimated state. This state is used by the walk controller to give the desired torque patterns. The torque controller calculates the input for the motor controllers based on the desired torque patterns and the angular velocities of the motors. And finally the motor controllers translate this signal to a

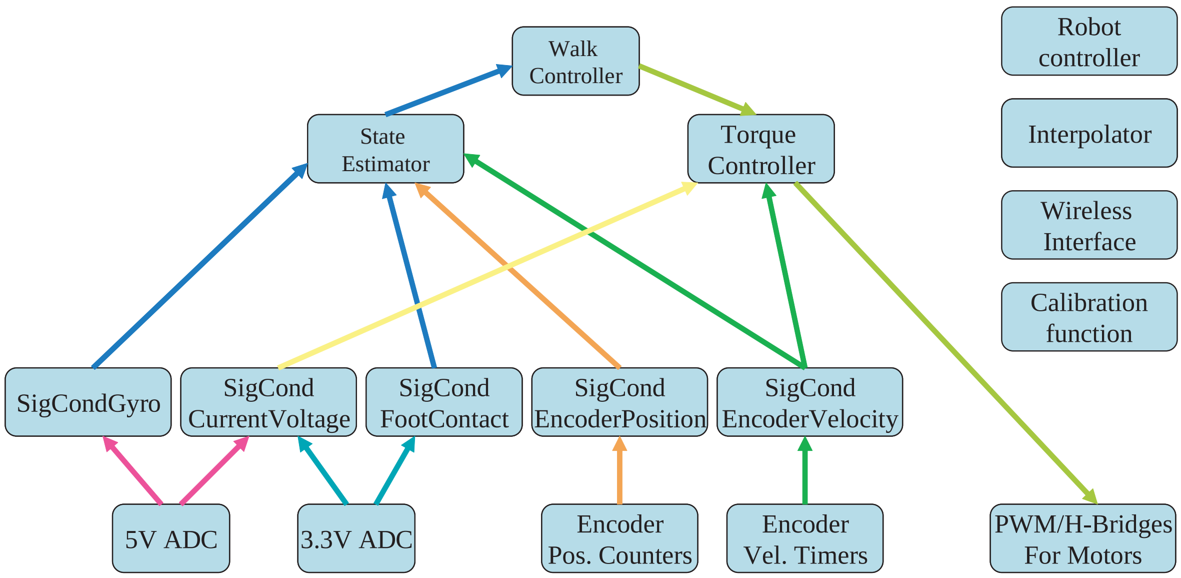

Figure 3.1: Block scheme of the software

pulse-width-modulated voltage for the motors.

Beside these basic functions there are a couple supporting functions. These supporting functions are:

- robot controller, for the overall control of the robot, e.g. switching between stand-by and walk mode,
- interpolator, used to imitate complex mathematical functions, such as sin functions, by interpolation of a table, to reduce the number of clock cycles,
- wireless interface, for communication between the robot and a laptop,
- calibration function, for calibration of the encoders, gyroscopes and linear accelerometers.

3.2 Walk controller

The walk controller is the most important part of the controller and it decides how the robot moves by setting desired torques for the motors. The desired torques are selected by two state-machines, one for the hip and one for the feet (figure 3.2).

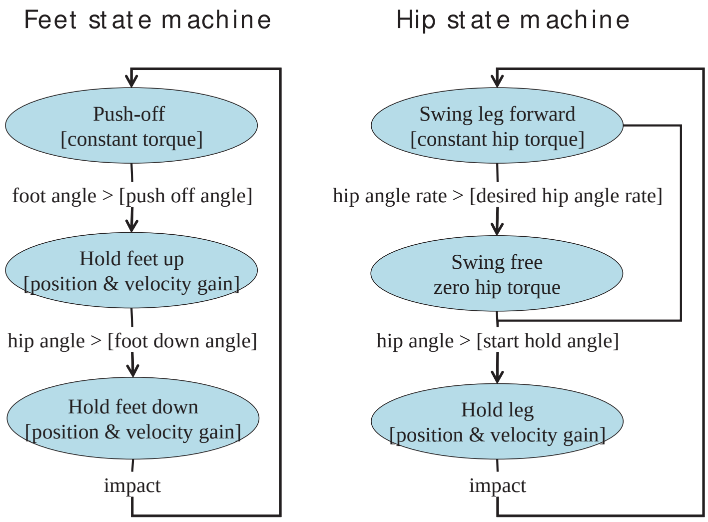

Figure 3.2: State machines for the hip and feet control. The words between the squared brackets are parameters that can be tuned

The swing feet start with push-off directly after impact. This is a push-off with a constant torque for a desired displacement, to get a fixed amount of energy in the system. After the push-off the swing feet are hold in an upright position to get ground clearance to swing the leg forward. After the swing leg is swung far enough to the front the swing feet are lowered and held in place. The feet are now ready for impact and when impact occurs the state machine starts all over again, but now with the other feet as swing feet.

The state machine for the hip starts after impact with swinging the swing leg forward with a constant torque until the hip angle rate reaches the desired rate. Then the leg swings forward under zero torque until it reaches a certain hip angle. At this angle a pd-controller starts to hold the hip at a desired angle. The pd-controller keeps the leg in place for the impact and at impact the state machine starts over. The hip state machine has a by-pass for the ’free swing’ state, so that if the leg does not reach the desired angle rate it can still go to the ’hip hold’ state.

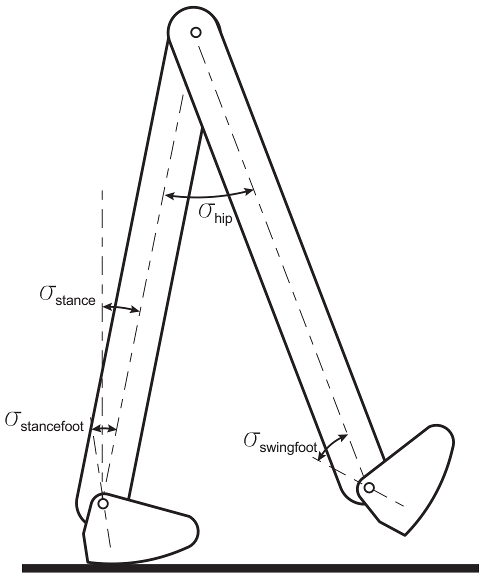

Figure 3.3: The angles of the state

3.3 State estimator

The state estimator takes all the sensory inputs to determine the state of the robot. The state consisted of all three internal angles and angle rates, the angle and angle rate of the stance leg with respect to the ground and a binary state which of the legs is the stance leg (figure 3.3). The internal angles and angle rates are trivial to determine because they are the angles that the encoders are measuring. The angle rate of the stance leg is the same as the gyro rate if the outer leg is the stance leg. If the inner leg is the stance leg then the stance leg rate is the gyro rate plus the hip angle rate.

To get the angle of the stance leg is more work. The angle of the stance leg can be found by integrating the gyro rate. But the integration of the gyro rate can cause a drift of the stance leg angle, because the gyro rate has always a small offset. To prevent this error from growing the stance leg angle is calibrated every step. This is done when the robot is in the double stance phase. With all the feet on the ground, the stance leg angle is calibrated by solving a geometrical equation with the assumption that the floor is level.

3.4 Torque controller

The torque controller sets the voltages for the motors so that the motors supply the same torques as the desired torques. To get the relation between motor voltage and motor torque a model is used.

3.4.1 Motor model

A dc-motor can be modeled as follows:

$$UI = I^2R + T_m\omega_m + L\frac{dI}{dt} \tag{3.1}$$

in which U is the dc-voltage, I is the current, R is the electrical resistance, T_m is the motor torque, ω_m is the angular speed and L is the electrical induction. The electrical inductions term can be neglected because the time scale of this is much faster then the time scale of the controller. With this simplification and with the linear relationship between the torque and the current, $T = Ik$, the motor equation can be simplified to:

$$U = \frac{R}{k}T_m + k\omega_m \tag{3.2}$$

in which k is the motor constant. Beside the motor the gearbox has to be modeled. The gearbox can be modeled as a scaling of the torque and angular speed with the gearbox ratio G (equations 3.4 and 3.5). In the gearbox model a torque term T_f is subtracted to model the fiction in the motor and gearbox. The fiction is assumed to be a combination of coulomb and linear fiction(equation 3.5).

$$\omega_g = \omega_m G \tag{3.3}$$

$$T_g = \frac{T_m}{G} - T_f \tag{3.4}$$

$$T_f = c_1\omega_g + c_0 sign(\omega_g) \tag{3.5}$$

The motor controller should control the voltage U so that the output torque of the gearbox T_g is equal to the desired torque T_{des}. Combining equations 3.2 to 3.5 results in the following relation between U and T_{des}:

$$U = \frac{GR}{k}T_{des} + \frac{c_1GR + k^2}{k}\omega_m + \frac{c_0GR}{k}sign(\omega_g) \tag{3.6}$$

In this equation there is, beside the desired torque T_{des}, the angular velocity ω_g and five constants. The angular velocity ω_g can be measured by the encoders on the motors. Of the five constant there are three, the gearbox ratio G, electrical resistance R, motor constant k, that can be found in the spec sheets and the other two constants, linear friction constant c_1 and coulomb friction constant c_0, have to be found by doing experiments.

For the feet motors the torque controller has one more addition, a compensation for the springs in the cable system. The torque due to the springs T_s is linear with the angle of the gearbox θ_g. Applying this to the motor equations results in:

$$T_m = (T_{des} + T_f + T_s)G \quad (3.7)$$

$$T_s = c_s(\theta_g - \theta_0) \quad (3.8)$$

$$U = \frac{GR}{k}T_{des} + \frac{c_1GR + k^2}{k}\omega_m + \frac{c_0GR}{k}sign(\omega_g) + \frac{c_sGR}{k}(\theta_g - \theta_0) \quad (3.9)$$

in which c_s is the spring constant and θ_0 the rest point of the springs.

3.4.2 Motor testing

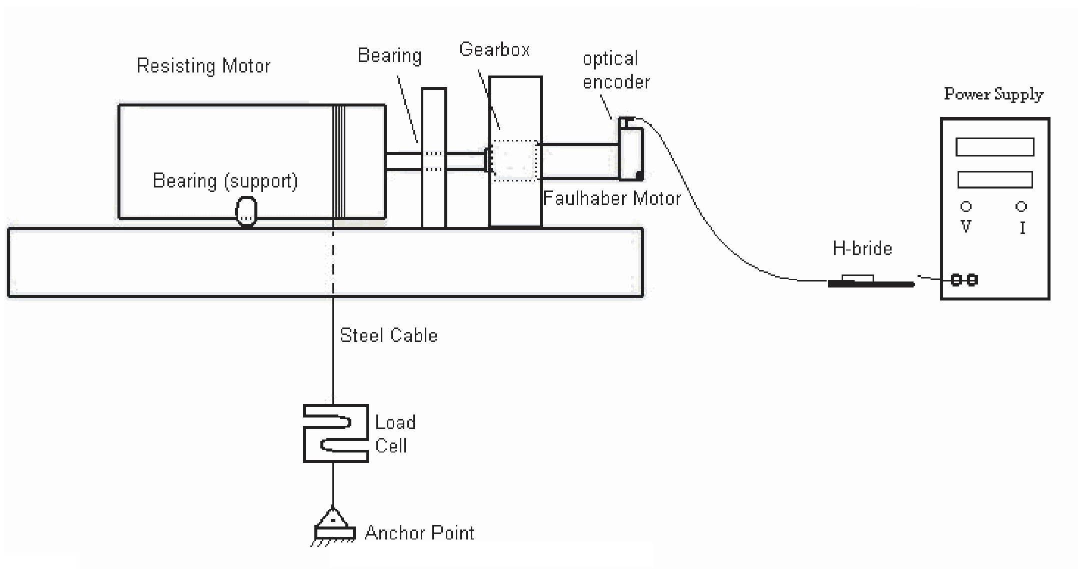

Figure 3.4: Setup for testing the motor. Image by Carlos Arango

The motors used in the robot where tested in order to find the friction properties and to check the motor constant and the resistance from the spec sheet. The motor is tested by attaching it to another motor (figure 3.4). This resisting motor is supported in such way that it can rotate freely. Attached to and wrapped around the resisting motor is a cable. The other end of the cable is attached with load cell to the ground. With this construction the torque of the test motor can be measured with the load cell, because this is the only connection that keeps the motor shaft from rotating with respect to the fixed world. Beside the torque, the motor velocity and current are measured. The velocity is measured with an optical encoder and the current is measured with a current meter.

The voltage to the test motor is set with a h-bridge. This h-bridge reduces the voltage by means of pulse-width-modulation. For the tests the voltage of the test motor and the current to the resisting motor is varied, to get motor data by a variety of torques and speeds. Figure 3.5 shows a graph of the measurements. The PWM values, in this graph, give the voltage to the test motor, where 1 means 12 volt and 0 means 0 volt.

The motor model is fitted on the motor data and the result is shown in figure 3.5. The torque-velocity curves are straight lines except for a jump by zero velocity due to the coulomb friction. There is a deviation from the model in the measured torque-velocity curves for torques above 0.95. This is caused by the limited power supply, that has a maximal current of 5A.

Table 3.1 gives the motor model constants found by the experiments and the values from the spec sheet. The value found for the motor resistance is surprising, because it is almost twice as high as on the spec sheet. This results in four times as much electrical loss for a given torque.

Table 3.1: Motor model constants

	from experiments	from spec sheet	
Motor constant k	0.0162	0.0173	Nm/A
Motor resistance R	1.2	0.71	Ω
Linear friction constant c_1	0.01	–	Nms/rad
Coulomb friction constant c_0	0.08	–	Nm

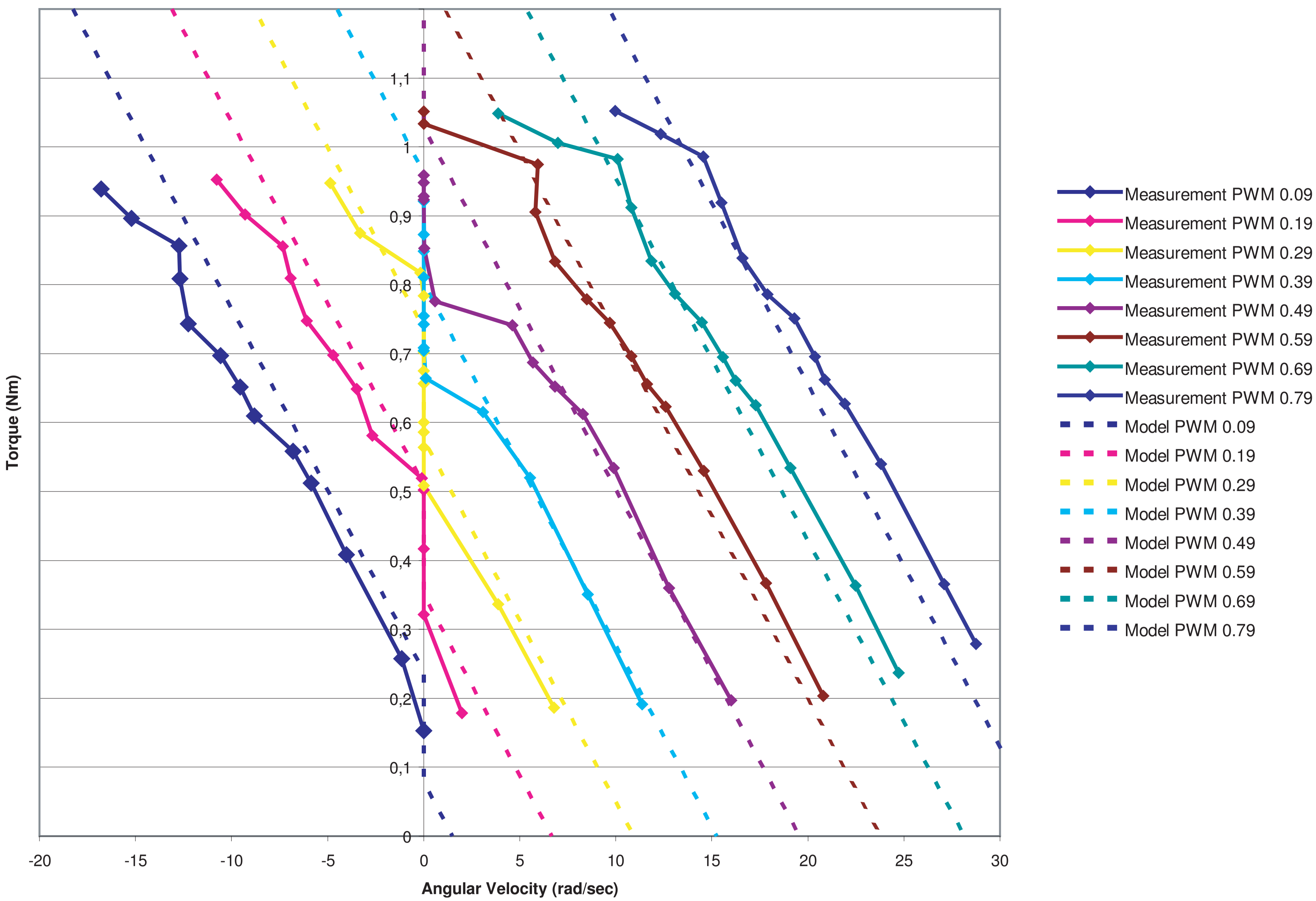

Figure 3.5: Motor torque versus motor velocity for test data and motor model

Chapter 4

Results

On December 3, 2006, the robot reached its goal by autonomously walking just over one kilometer (1003 meters). After walking 40 minutes the robot fell by tripping over a sand-pit cover. When the robot fell it still had battery power left. The energy consumption is approximately 40 watt and the batteries have a capacity of about 80 watt-hours, so the robot can walk for about two hours. With an average speed of about 1.5 km/h the maximal traveling distance is about 3 kilometers.

The record was set on a rubber indoor running track. The surface of the track was important for the performance of the robot, because the surface is an important factor in the friction between the feet and the floor (see section 5.1).

With the steering the robot was able to make turns with a radius of 25m. The record was set by walking in circles of approximated 100 meter and using the rc-steering for minor corrections.

Figure 4.1: The Cornell Ranger during the one kilometer run

Chapter 5

Discussion & future work

5.1 Discussion

Before the goal of walking 1 kilometer was reached we encountered a variety of problems, from gearbox failure to electronic meltdown. The three major problems will be discussed below.

- The first major problem was that at start-up the h-bridges (motor amplifiers) had a random output. If one of the outputs is randomly high it will cause a motor to start rotating at full power. These sudden bursts of power caused a couple of broken springs, but luckily no serious damage. To overcome this problem a watchdog timer was installed. The timer makes sure that the h-bridges can only be high if the software is running.

- A second problem was that, due to a design mistake, the push-off torque was to small. This small torque limited the amount of energy that could be put into the system during push-off and it resulted in less stable walking. The problem was partly solved by applying a forward hip torque during push-off. The hip torque causes the leg to move forward and this extends the leg foot combination causing an additional push-off force. The hip torque solves the push-off problem, but it is also one of the causes of the third major problem.

- The third problem was that for steering low friction is necessary and for push-off high friction. The steering works by a asymmetry in the outer feet, but there are always at least two feet on the ground, so steering only works if a foot can slip. During push-off there is a forward hip torque and this torque causes the feet to slip forward if there is not enough friction. Slippery tape was applied on the heels to get low friction for the steering and for high friction during push-off a spike was placed in the toe (figure 5.1).

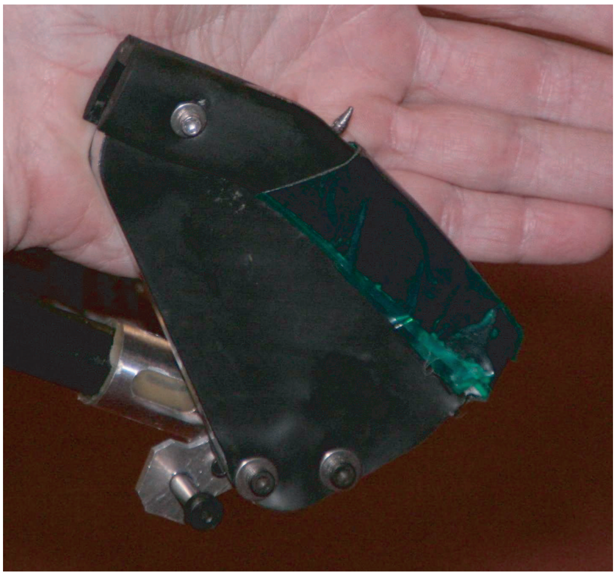

Figure 5.1: Foot with slippery tape on the heel for low friction during steering and spike in the toe for high friction during push-off

5.2 Future work

Now the robot reached its goal that does not mean that no work can be done with it. Some improvements can be made and a lot more experiments can be done. The first improvement will be the replacement of the gearboxes of the feet motors. The gear ratio will be lowered from 1:14 to 1:44, to increase the push-off torque. With a higher push-off torque there is no need anymore for a hip torque during push-off and therefore the chance of slipping during push-off will decrease.

There is a lot of room for improvement in the walk controller. The energy efficiency as well as the robustness can be improved. The following things can be done to improve the walk controller:

- At the moment the controller is not using any of the data from the gyroscopes. With this data it is possible to see during a step if the robot goes to slow or to fast and react by for example a change in step size. We already did some experiments with changing the step size linear with the rate of the stance leg at mid stance. The results were promising. With this controller the robot was able to overcome a big push in the back by taking some bigger steps. For a push in the other direction it was less effective because with small steps the push-off becomes less effective.

- Another way of improving the walk controller can be to add feedback over one step. This controller looks at the last step to see if it should adjust something for the next step. With this kind of controller it is possible to make the eigenvalues over two steps zero. This means that if a small disturbance occurs it is detected at the end of that step and at the end of the next step the robot is back to the steady walking cycle. In simulation we showed that this works, but make it work on a prototype requires more work.

- The current walk controller is optimized by hand. To improve the controller the optimizing could be done with a optimizing routine. To do this you need a clear measure of the performance of the robot. This performance measure should at least consist of data about energy efficiency and robustness. To optimize the controller, without running the robot over and over, a simulation is needed. This simulation should have the dynamics of the robot, including the motor properties.

Besides making the robot walk better, it can be improved by increasing the versatility. Things that can be added are for example: starting and stopping, balancing and walking backwards.

Chapter 6

Conclusion

This report describes the design of the Cornell Ranger. A robot build to show that it is possible to make a reliable walking robot using the dynamic walking method. To show that the robot is reliable it should walk fully autonomous more than one kilometer without falling. To reach this goal the following requirements were set:

- reliable mechanics and electronics,
- energy efficient movements,
- a way of steering the robot.

The mechanical design was kept as simple as possible in order to get reliable mechanics. All the heavy parts were concentrated around the hip to reduce the work needed to swing the leg and to get a good symmetry between the legs. The robot was designed with three internal degrees of freedom (one hip and two ankles). All these degrees of freedom are actuated with dc-motors. For the ankles a cable system is used to transfer the power from the motors to the ankles.

The robot is steered by making an asymmetry in the feet actuation of the outer feet. The steering system is made up by a non back drivable gear system, an rc-motor and an rc-controller. The system is energy efficient, because it only uses energy for changing the steer angle and almost no energy for holding it in place. The control of the steering is also reliable, because of the use of a widely used and reliable rc-controller.

The control of the robot is done by a controller in software. The four most important parts of this controller are: the signal processor, the state estimator, the walk controller and the torque controller. The signal processor translates the signals from the sensors to usable units. The state estimator estimates the state based on the sensor data.

This state is used by the walk controller to determine the desired torques. The torque controller makes sure that the motors output torques are equal to the desired torques.

After 3 months of hard work, the robot reached its goal by walking 1003 meter. The robot had the followings characteristics: speed 1.5 km/h, power consumption 40 watt, battery capacity 80 watt-hours, step length 36 centimeter, mass 5 kg and leg length 1 m. Now the robot reached its goal there is still room for improvement, i.e. the push-off can be improved and the walk controller can be optimized.

So the robot reached its goal and showed that a walking robot can walk reliable with using the dynamic walking method. And there is still room for improvement, so there is a promising future for this robot and dynamic walking robots!

Chapter 7

Acknowledgements

This project was a real team effort. A team of more than 13 people worked very hard to make this project a success. Below you can find a list of the team members who where involved on this project during my visit and the topics they worked on.

- Staff:

  professor Andy Ruina management, advisor, optimizing walk controller, ...

  lab manager Jason Cortell management, mechanical design, electronics, computer interface, floating point, software, ...

- Graduate students:

  Gregg Stiesberg simulations, interpolator, optimizing walk controller, ...

  Pranav Bhounsule torque controller, clock cycle count, swing optimizing, ...

- Undergraduates:

  Carlos Arango fabrication, motor testing, motor modeling, ...

  Megan Berry drawings, design and fabrication of the steering , ...

  Alex Gates top bar fabrication, implementation of the feet sensors, ...

  Matt Haberland fabrication, state estimator, display, ...

  Sam Lee electronics, gyroscopes, current sensors, ...

  Andrew Mui motor testing, floating point, voltage sensors, ...

  Andrew Spielberg fabrication, motor testing, swing optimizing, ...

- High school student:

  Ben Oswald CAD modeling, fabrication, CNC programming, ...

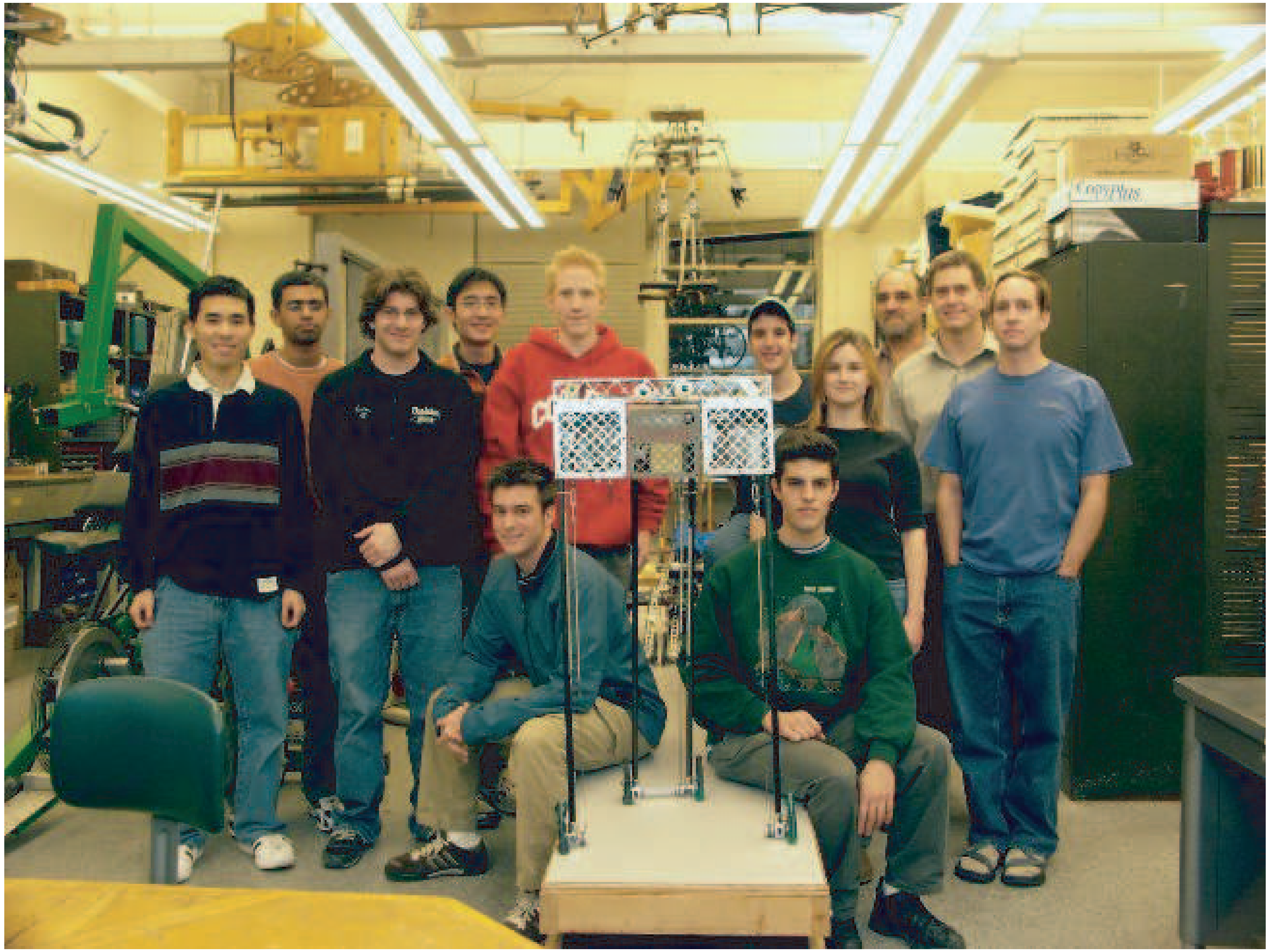

Figure 7.1: Group photo, with from left to right: Andrew Mui, Pranav Bhounsule, Alex Gates, Sam Lee, Matt Haberland, Daniël Karssen, Cornell Ranger, Andrew Spielberg, Carlos Arango, Megan Berry, Andy Ruina, Jason Cortell, Gregg Stiesberg

Bibliography

[1] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.

[2] S. H. Collins, M. Wisse, and A. Ruina. A 3-d passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607–615, 2001.

[3] S. H. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

[4] R. Q. Linde. Passive bipedal walking with phasic muscle contraction. *Biological Cybernetics*, 81:227–237, 1999.

[5] M. Wisse and J. v. Frankenhuyzen. Design and construction of mike; a 2d autonomous biped based on passive dynamic walking. In *Proceedings of 2nd International Symposium on Adaptive Motion of Animals and Machines*, 2003.

List of Figures