New Jersey Institute of Technology
Digital Commons @ NJIT

Theses | Electronic Theses and Dissertations

Spring 5-31-2000

Gait development for use in dynamic gait optimization of qudrupedrobot walking

Mark Will
New Jersey Institute of Technology

Follow this and additional works at: https://digitalcommons.njit.edu/theses

Part of the Mechanical Engineering Commons

Recommended Citation
Will, Mark, "Gait development for use in dynamic gait optimization of qudrupedrobot walking" (2000). *Theses*. 805.
https://digitalcommons.njit.edu/theses/805

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

GAIT DEVELOPMENT FOR USE IN DYNAMIC GAIT OPTIMIZATION OF QUADRUPED ROBOT WALKING

by
Mark Will

The ability of walking robots to operate in areas that are inaccessible to wheeled robots has lead to significant research in the field of gait development and optimization for these robots. In this particular study, a catalog of gaits for use in a dynamic gait optimization system to optimize the walking speed of the quadruped Arturo robot on flat terrain is developed. This catalog of robot gaits was developed using a genetic algorithm formulation; various combinations of the selection, mutation, and crossover operators were analyzed.

The Arturo robot was modified so that physical verification of the developed gaits could be carried out. The performance of several gaits was analyzed to determine both robot performance and suitability of the gait for use in a dynamic gait optimization system.

The feasibility of using solely the position feedback from the joints for surface determination was examined. Piezoelectric crystals (Leybold Inficon 6 Mhz oscillators) were also examined for this application.

GAIT DEVELOPMENT FOR USE IN DYNAMIC GAIT OPTIMIZATION OF QUADRUPED ROBOT WALKING

by
Mark Will

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Department of Mechanical Engineering

May 2000

Blank Page

APPROVAL PAGE

GAIT DEVELOPMENT FOR USE IN DYNAMIC GAIT OPTIMIZATION OF QUADRUPED ROBOT WALKING

Mark Will

Dr. Michael Recce, Thesis Advisor — Date
Associate Professor of Computer and Information Science, NJIT

Dr. Zhiming Ji, Committee Member — Date
Associate Professor of Mechanical Engineering, NJIT

Mr. Bhaven Patel, Committee Member — Date
Project Manager Engineering, Servo Systems Company Industrial Robotics Division

BIOGRAPHICAL SKETCH

Author: Mark Will

Degree: Master of Science in Mechanical Engineering

Date: May 2000

Undergraduate and Graduate Education:

- Master of Science in Mechanical Engineering
  New Jersey Institute of Technology, Newark, NJ, 2000

- Bachelor of Science in Mechanical Engineering
  New Jersey Institute of Technology, Newark, NJ, 1999

Major: Mechanical Engineering

To my family and friends for all of their love and support

ACKNOWLEGMENT

I would like to express my appreciation to Dr. Michael Recce, my research advisor, for his support and encouragement throughout this project. I would also like to express my appreciation to Giorgio Grasso, on whose work this thesis is based, and Will Ferguson, my co-researcher, for his help and expertise. A special thanks goes to Dr. Zhiming Ji and Mr. Bhaven Patel for serving on the approval committee.

My fellow researchers in the Neural Computation and Robotics Laboratory also deserve thanks for their inspiration and help.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of this thesis is to develop a catalog of gaits suitable for use in a dynamic gait optimization system that would change the gait of a quadruped robot to maximize its speed on a primarily flat surface.

To develop the robot gaits we used a genetic algorithm and analyzed various combinations of two basic operators – mutation and crossover. The following combinations were analyzed: (1) constant mutation rate, (2) step change in mutation rate, (3) crossover, (4) crossover and constant mutation rate, and (5) crossover and constant mutation rate with the pocket algorithm.

For the verification of the gaits, the Arturo robot [Grasso 00] was modified to allow reliable testing on various surfaces. The performance of several gaits was analyzed to determine both robot performance and suitability of the gait for use in a dynamic gait optimization system.

The feasibility of using solely the position feedback from the joints for surface determination was examined. Piezoelectric crystals (Leybold Inficon 6 Mhz oscillators) were also examined for this application.

1.2 Background Information

Autonomous robots have been used for many tasks where the use of humans is either extremely dangerous or impossible. Such applications include interplanetary exploration (such as the Mars Sojourner Rover), hazardous material cleanup, and live explosive retrieval and destruction. Currently, the majority of these autonomous robots move on either wheels or tracks (which are effectively just extended wheels).

Legged robots have some significant advantages over their wheeled counterparts. First, legged robots can traverse much rougher terrain than wheeled robots since they have the ability to cross obstacles roughly their own size [Wettergreen 95]. Second, legged robots leave less evidence of their passing than wheeled robots. Since the wheeled robots rely on all wheels being firmly on the ground for propulsion, they leave a distinct set of tracks behind them. Legged robots, however, leave only a pattern of footsteps, which is preferable especially in environmentally sensitive areas. Third, legged robots have the ability to choose their footfalls to avoid dangerous areas (such as deep holes), while wheeled robots must maintain a continuous contact with the ground.

In the past 20 years, there has been a significant amount of research done on legged robots. The majority of these robots have walked with a statically stable gait – the center of gravity of the robot is always within the support polygon of the legs. Such gaits simplify the analysis and control of the legged robot because they are stable at all points during the robot movement.

These statically stable legged robots have been shown to autonomously adapt to irregular terrain. For example, Dante II [Wettergreen 95] (an 8-legged statically stable robot) walked into the volcanic crater of Mt. Spurr and sampled the volcanic gasses with minimal human direction. There have been a variety of other successes tested on simulated terrain under laboratory conditions. Many of these statically stable legged robots have been hexapods (6-legged robots), such as Terrapin I [Azarm, et.al. 87], AMBLER [Wettergreen 92], and even the virtual SimPod [Wettergreen 95]. Others have been quadrupeds (in which three legs are kept on the ground at all times), such as PV II [Hirose et.al. 83], TITAN III [Hirose et.al. 85], MENO [Sukhatme 97], MENO II [Sukhatme et.al. 97]. In addition, Mark Tilden has developed a series of legged robots (hexapods, quadrupeds, and bipeds) that appear to handle irregular terrain with no intervention [Trachtman 00], but they cannot be commanded to perform any useful function.

Besides the statically stable legged robots, a series of dynamically stable robots have been developed. These robots, like most mammals, use the inertia of the center of mass of the body to maintain stability during motion. In 1981, Mark Raibert demonstrated the principle of dynamic walking with a one-legged hopping machine [Raibert 85] and proved that this principle could be extended to robots with larger numbers of legs. In 1985, a quadruped robot named COLLIE moved utilizing a trotting gait (opposing legs move in phase) [Miura et.al. 85]. Later robots TITAN IV [Yoneda 94] and NINJA-1 [Nagakubo 94] realized trotting gaits while traversing uneven terrain (NINJA-1 was also capable of walking dynamically on vertical surfaces). In 1998, a quadruped robot controlled by a simple neural network was shown to be capable of traversing small steps without breaking its trotting gait [Kimura 98].

During the development of these dynamically stable legged robots, the main thrust of the research was to develop methods to handle highly irregular terrain. This thesis, on the other hand, will concentrate on the development of gaits necessary to optimize the speed of Arturo [Grasso 00], a dynamically stable quadruped robot, on relatively flat surfaces (perturbations in the surface less than one-fiftieth of the leg length). As far as the author is aware, such a catalog of gaits does not yet exist.

1.3 The Robot

Arturo is an eight-degree of freedom quadruped robot; there is an independently controlled hip and knee joint on each leg. Each of the eight joints is controlled by a National Semiconductor LM628N-6 precision motion controller (a complete PID controller on a single chip) and powered by a brush servomotor with a 50:1 gearbox. The position feedback for the PID controllers is obtained from a Hewlett Packard HEDS-9140-A00 rotary encoder mounted on the horizontal shaft for each joint. The hip joints are driven with a spur gear set with one gear fixed to the upper half of the leg; the knee joints are driven by a chain and sprocket, with one sprocket fixed to the lower half of the leg.

The feet of the robot are semi-spherical and constructed by cutting a squash ball in half. This foot construction eliminates the need for an ankle joint. The overall dimensions of Arturo are approximately 20 inches long by 11 inches wide by 14 inches tall. The robot weighs about 20 lbs. and is shown in Figure 1.

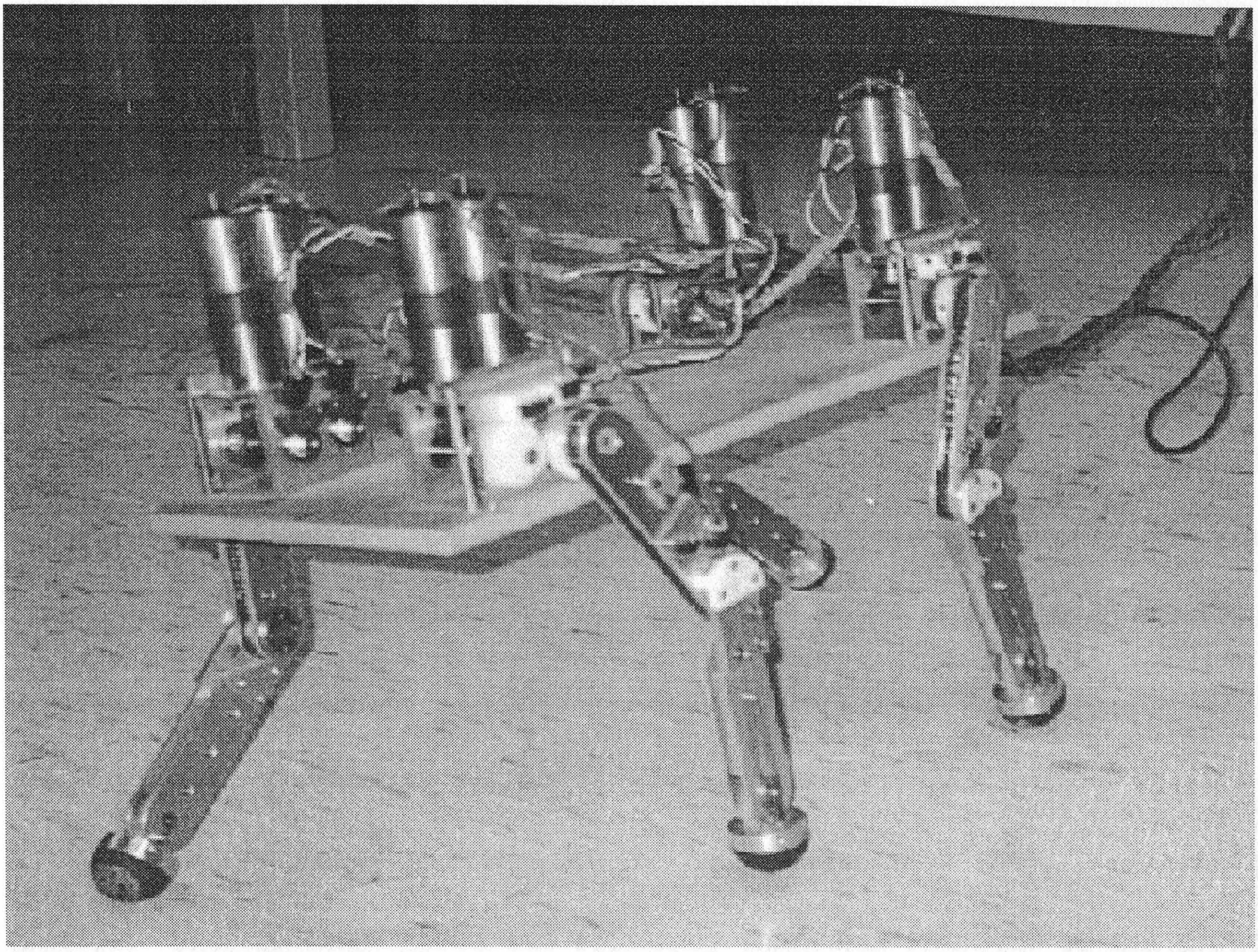

Figure 1 Arturo robot walking.

CHAPTER 2

GAIT DEVELOPMENT

2.1 Problem Statement

The purpose of gait development for Arturo is to provide a catalog of gaits from which the dynamic gait optimization system can choose based on the surface condition. This catalog of gaits should range from high speed (very little weight given to stability) to high stability (very little weight given to speed). Due to the dynamic stability of Arturo's movement, such a catalog of suitable gaits cannot be developed analytically. Each prospective gait must be simulated in order to obtain a reasonable speed and stability value.

Due to the need for the simulation of each prospective gait, a genetic algorithm was chosen to implement the gait optimization process. Since each simulation produces a stability value and a fitness value, the relative weight of each can be varied to produce the necessary catalog of gaits. The genetic algorithm also allows for some optimization of speed for each set of relative weights.

For this gait development, a trotting gait was chosen for several reasons. First, it provides relatively high speed but constantly maintains two feet on the ground for stability. Second, the trotting gait can be started from a standing position, eliminating the need to switch types of gait; the pacing gait is not sufficiently stable to start from a standing position.

2.2 Genetic Algorithm Formulation

Initially described by John Holland in the 1970's, genetic algorithms have become a very important tool for function optimization [Rawlins 91]. Based loosely on the biological process of natural selection, genetic algorithms maintain a population of "organisms" (possible solutions) and utilize crossover and/or mutation to create the next generation of organisms. Increased probabilities of reproduction (generally in the form of a fitness function) are associated with the more optimal "organisms" which leads to a general optimization of the population as a whole.

2.2.1 Genome

In order to carry out the genetic algorithm, the prospective robot controllers must be encoded as a genome. The genome must be a string (loosely following the biological DNA code). Since Arturo will be walking in a periodic gait, the simplest way to represent the joint motions is with a Fourier series. This particular Fourier series is limited to five components due to the frequency response characteristics of the joint actuators [Grasso 00]. The genome for a robot controller is therefore represented as a string made from forty floating-point numbers – five Fourier components for each of eight joints. A sample robot controller is shown immediately below in Figure 2.

```
0.215701  0.235632  -0.239345  -0.058163  -0.327545
0.575739  -0.502497  0.244340  0.050798  -0.056467
-0.613145  -0.466973  0.210017  0.502510  0.055654
-0.130144  0.388856  -0.055264  -0.015177  -0.093770
-0.791787  0.223101  -0.165400  0.108819  0.365240
-1.000603  -0.047519  -0.111256  -0.126302  0.006345
0.868466  -0.261020  -0.423391  -0.121671  -0.133931
0.757606  0.019470  0.237614  -0.112963  0.184283
```

Figure 2 Sample robot controller file.

2.2.2 Selection Operator

In order to continue to improve the population of robot controllers, the better controllers must survive and the poorer controllers must die (the equivalent of the biological natural selection process). The determination of the best controller is implemented through the use of a fitness function. Since the "correct" fitness function is not very clearly defined by the problem, several different fitness functions were attempted. These functions are given in Equation 1, 2, and 3.

$$\mathbf{f} = \frac{1}{T}\int_0^T \frac{v}{1+s}\,dt \tag{1}$$

$$\mathbf{f} = \frac{1}{T}\int_0^T v\,dt \tag{2}$$

$$\mathbf{f} = \frac{\frac{1}{T}\int_0^T v\,dt}{\frac{1}{T}\int_0^T s\,dt} \tag{3}$$

Where

f = fitness function

v = velocity of robot

s = cosine of angle between vertical and robot body normal

t = time

T = total time of simulation (8 sec in this case)

The fitness function given by Equation 1 determines the average of the ratio of forward velocity to 1 plus the cosine of the angle between the robot body normal and the vertical (the addition of 1 in the denominator prevents division by zero). This fitness function will attempt to both maximize the controller velocity and minimize the deviation of the robot body from the horizontal. The fitness function given by Equation 2 just calculates the average velocity of the controller and is used to generate fast but not necessarily stable controllers. The fitness function given by Equation 3 is very similar to that in Equation 1, but calculates the ratio of the average velocity of the controller to the average value of the cosine of the angle between the robot body normal and the vertical. This particular fitness function was used because of a change in the simulator after the first set of trials; the simulator originally produced only a single fitness value (Equation 1), but was then modified to produce both a stability and velocity value.

The selection operator, which determines which of the robot controllers will actually reproduce, is probabilistic to better model the actual biological reproduction process. The actual selection function was implemented in a C code segment as follows [Grasso 00]:

1. Select a random number p on the interval [0, 1].
2. Compare p to the value of the fitness function f.
    - 2.1. If $f > p$ then the controller is reproduced.
    - 2.2. Otherwise, the controller is not reproduced.

2.2.3 Mutation Operator

In order to maintain the diversity of the population of controllers, some mutation must take place when the controllers are reproduced (similar to the biological process of mutation). A good discussion of mutation and its affect on genetic algorithm convergence can be found in [Tate 93]. In this particular case, the mutation operator was defined as follows [Grasso 00]:

1. Generate a random number a by utilizing Equation 4.

   $$\mathbf{a = p * 2 (r - ½)} \tag{4}$$

   Where

   p is the mutation probability (range [0,1])

   r is a random number on the range [0,1]

2. Generate another random number b on the range [0,1].
3. Compare the value of b to the value of p.

   3.1. If $b > p$, add a to the Fourier coefficient.

   3.2. Otherwise copy the coefficient directly.

4. Repeat the process for the other thirty-nine Fourier coefficients.

2.2.4 Crossover Operator

In sexual reproduction, the offspring has a combination of the genomes of the two parents; the crossover operator serves the same function in a genetic algorithm. This combination of parts of genomes tends to accelerate convergence of the genetic algorithm. A good discussion of the use of the crossover operator is found in [Eshelman 93]. In this particular case, the crossover operator was implemented as follows:

1. Generate a random number a on the range [0,1].
2. Compare a to ½.

   2.1 If $a > ½$, take the five Fourier coefficients (whole joint) from parent #1.

2.2 Otherwise take the five Fourier coefficients (whole joint) from parent #2.

3. Repeat for each of the other seven joints.

2.2.5 Implementation

Now that all of the necessary operators for the genetic algorithm have been described, the actual evolution process must be implemented. This process was implemented as a C shell script in Linux [Grasso 00]. An initial population of twenty controllers was created by applying 50% mutation to a basic controller file (a trotting gait with the legs moving in fifty hertz cycles). This population of twenty controllers was maintained for all generations during the evolution process. Each of the controllers was then simulated for 8 seconds (roughly 3-4 steps; 30-60 seconds computing time) to determine the velocity value v and the stability value s. Based on these two values and the particular fitness function (see Equation 1, 2 & 3 above), the next generation was developed using a combination of selection, mutation, and/or crossover. These combinations of operators are as follows:

1. Selection and Mutation
2. Selection and Crossover
3. Selection, Crossover, and Mutation

In the cases where crossover was used, the pocket algorithm was also implemented. This algorithm ensures that the best controller will never be lost; it guarantees a 100% probability of reproduction without mutation.

2.3 Results

2.3.1 Selection and Mutation

The first five evolutionary trials utilized only mutation and selection with the fitness function given by Equation 1. The only variable (besides the initial random population of controllers) was the mutation rate and the point at which this rate was changed (a step change). The results of these trials are shown in Figures 3-7.

There was a significant change in the fitness value increase rate (change in fitness value per generation) between an initial mutation rate of 7% (Figure 5) and an initial mutation rate of 10% (Figures 3, 4, and 6). This fitness value increase rate increased 2-3 times with the 3% change in mutation rate. Increasing the mutation rate from 10% (Figures 3, 4, and 6) to 15% (Figure 7) showed a negligible change in the fitness value increase rate. No significant changes in performance were noted when the location of the mutation rate change was varied from generation 30 (Figure 4) to generation 40 (Figure 6) to generation 50 (Figure 3). Figure 8 is identical to Figure 6 with the stability and velocity values separated. This figure is provided for comparison with the results of later evolutionary trials.

The next three trials utilized mutation and selection with the fitness function given by Equation 2 in an attempt to generate high velocity controllers. All three of these trials utilized an initial mutation rate of 10% followed by a mutation rate of 5% after generation 40, similar to the fourth trial above. Since no other parameters (with the exception of the initial random population) were varied, the results from all of these trials were very similar. A sample set of results is shown in Figure 9.

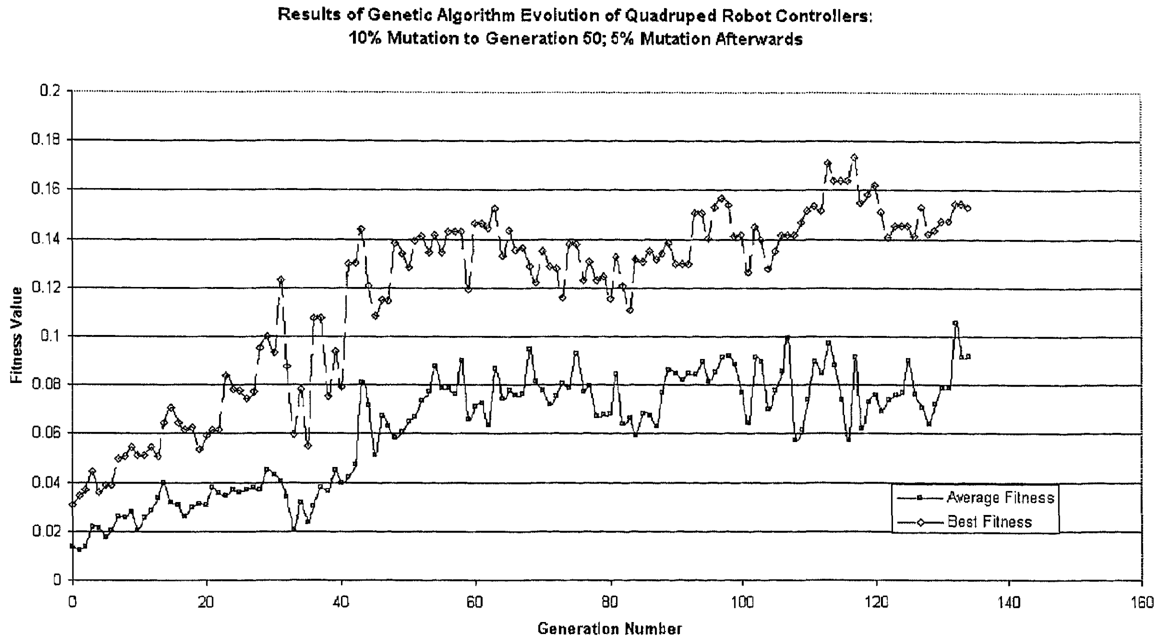

Figure 3 Results of first evolutionary trial.

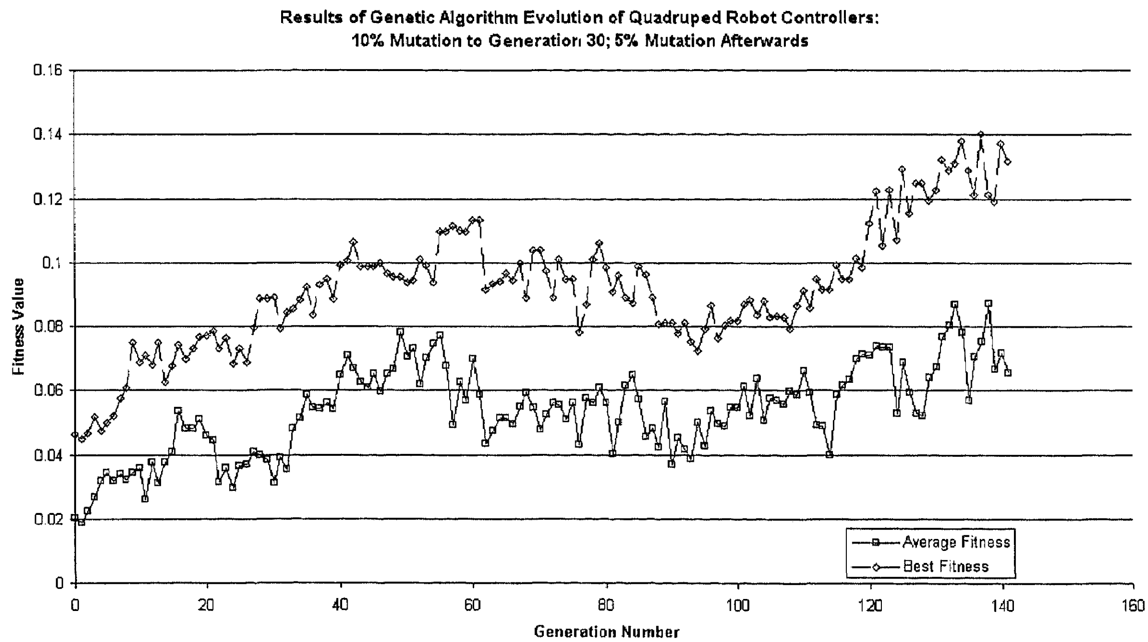

Figure 4 Results of second evolutionary trial.

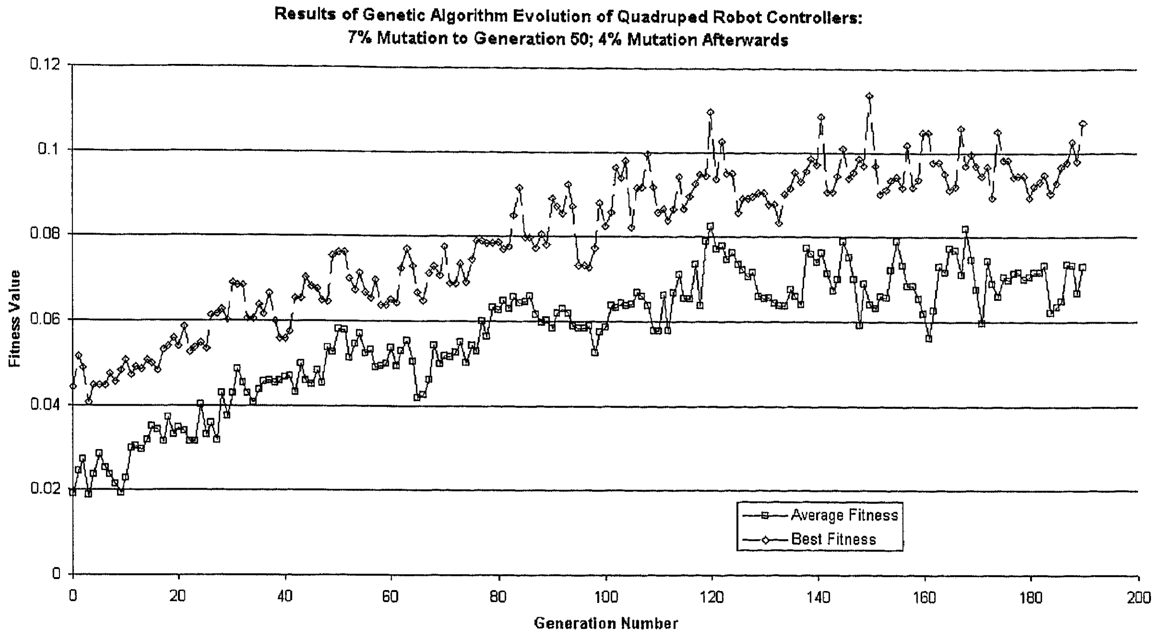

Figure 5 Results of third evolutionary trial.

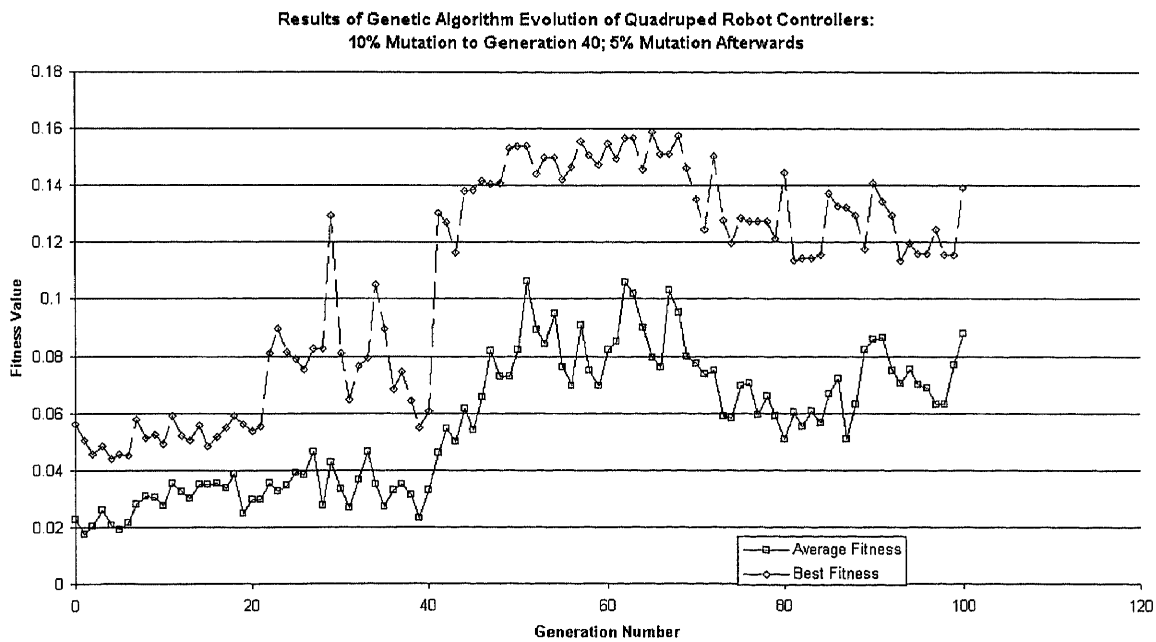

Figure 6 Results of fourth evolutionary trial.

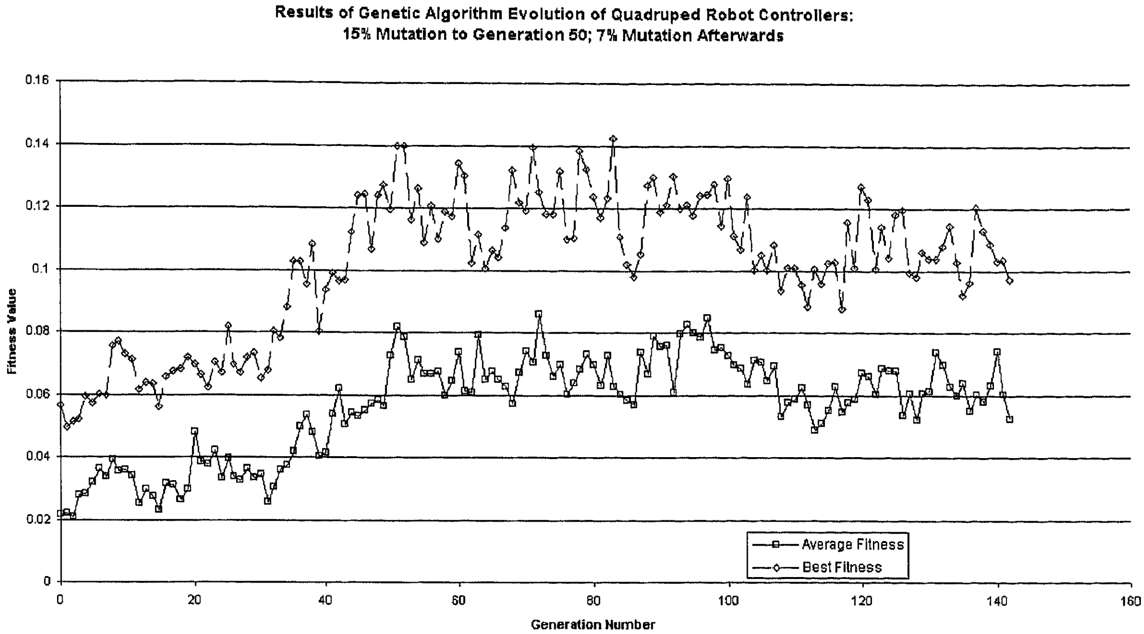

Figure 7 Results of fifth evolutionary trial.

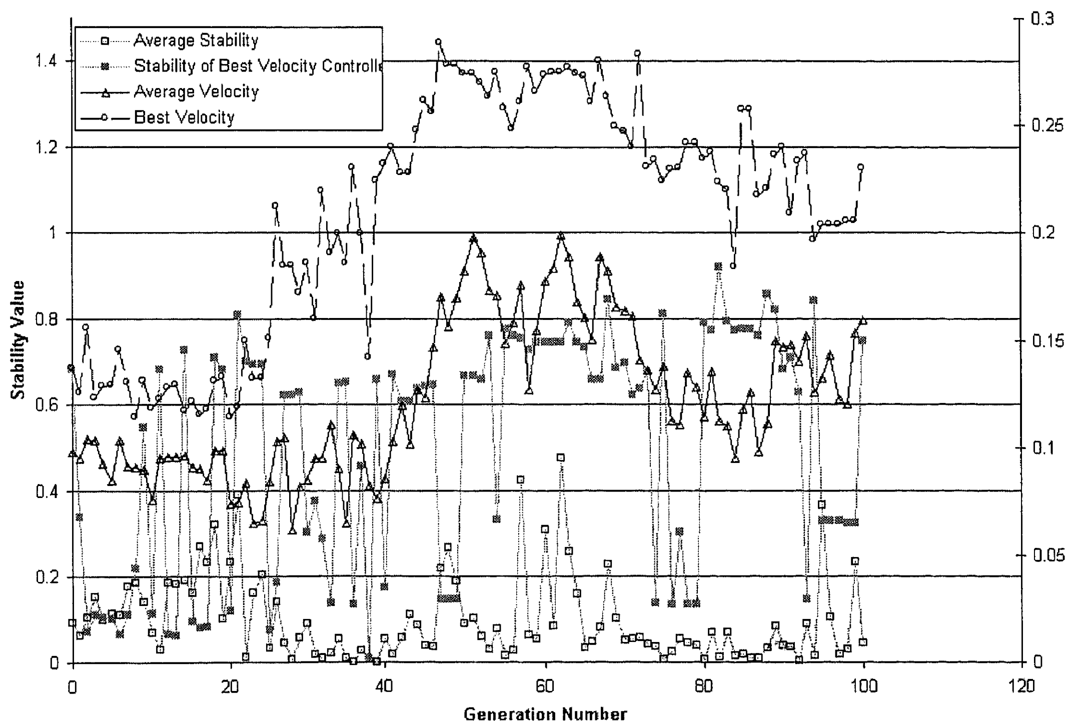

Figure 8 Results of fourth evolutionary trial with stability and velocity values separated.

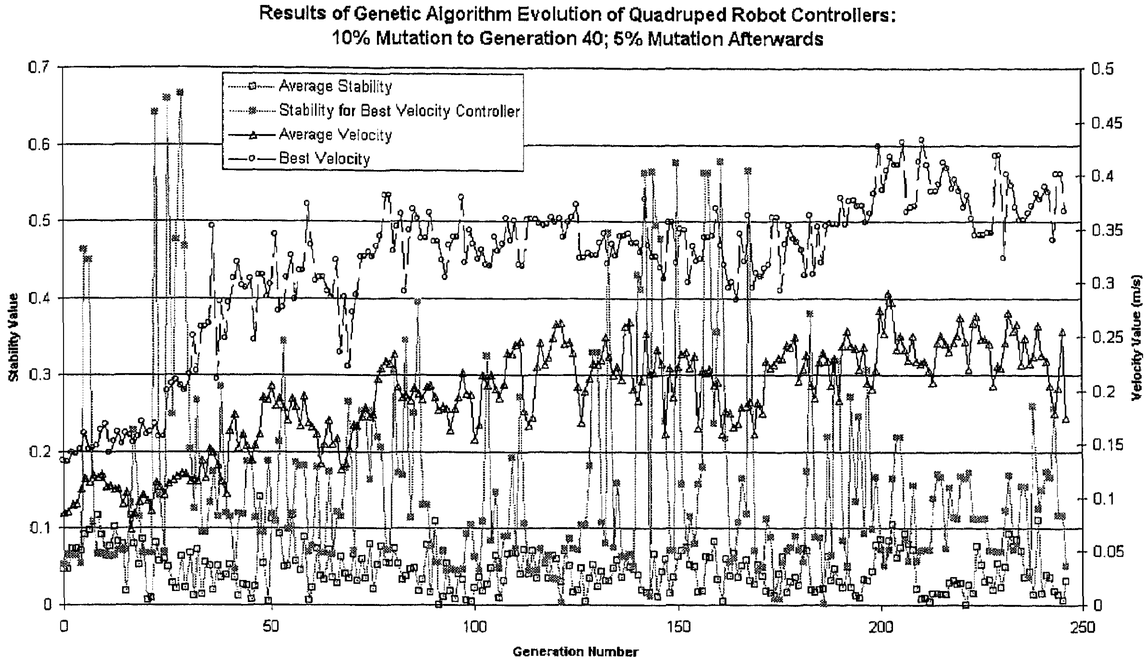

Figure 9 Results of evolutionary trial with mutation and selection with the fitness function given by Equation 2.

The fitness function given by Equation 2 developed controllers that were 33% faster than those developed with the fitness function given by Equation 1 (Figures 8 and 9). The velocity of the fastest controller increased from 0.33 m/s to 0.44 m/s with this change in the fitness function. The stability value of the best controller, however, was reduced by a factor of 2 with this change in the fitness function. Such a change is expected since the fitness function given by Equation 2 places no weight on the stability of the controller.

2.3.2 Selection and Crossover

Several evolutionary trials were conducted utilizing selection and crossover with the fitness function given by Equation 2. For these trials, the pocket algorithm was also implemented; this algorithm ensured that the best controller would appear in the next generation with no mutation. The best controller was also used as a parent for all of the controllers in the following generation. Unfortunately, this set of trials did not produce many useful results. It took about 9 generations for all of the diversity in the population to disappear – all of the controllers became the same as the best controller from the second or third generation.

Another implementation of the crossover operator was also attempted. This implementation allowed the sets of Fourier components for different types of joints (knee and hip joints) to be mixed. In other words, the Fourier coefficients for the knee joint of the offspring might have been the coefficients for a hip joint in one of the parents. This implementation of crossover produced dismal results – after about 7 generations, none of the controllers were capable of walking.

2.3.3 Selection, Mutation, and Crossover

Since crossover and selection alone did not produce useful results, these two operators were combined with mutation. Using the fitness function given by Equation 2, several evolutionary trials were run. The first used a constant mutation rate of 10%; the results of this trial are shown in Figure 10. The next used an initial mutation rate of 10% and a mutation rate given by Equation 5 after generation 40.

$$\textbf{mutation rate} = \frac{\textbf{400}}{\textbf{generation number}} \tag{5}$$

The results of this trial are shown in Figure 11.

The results of these two trials were very similar. In both of these trials, a controller with a velocity greater than 0.4 m/s was generated in about 50 generations.

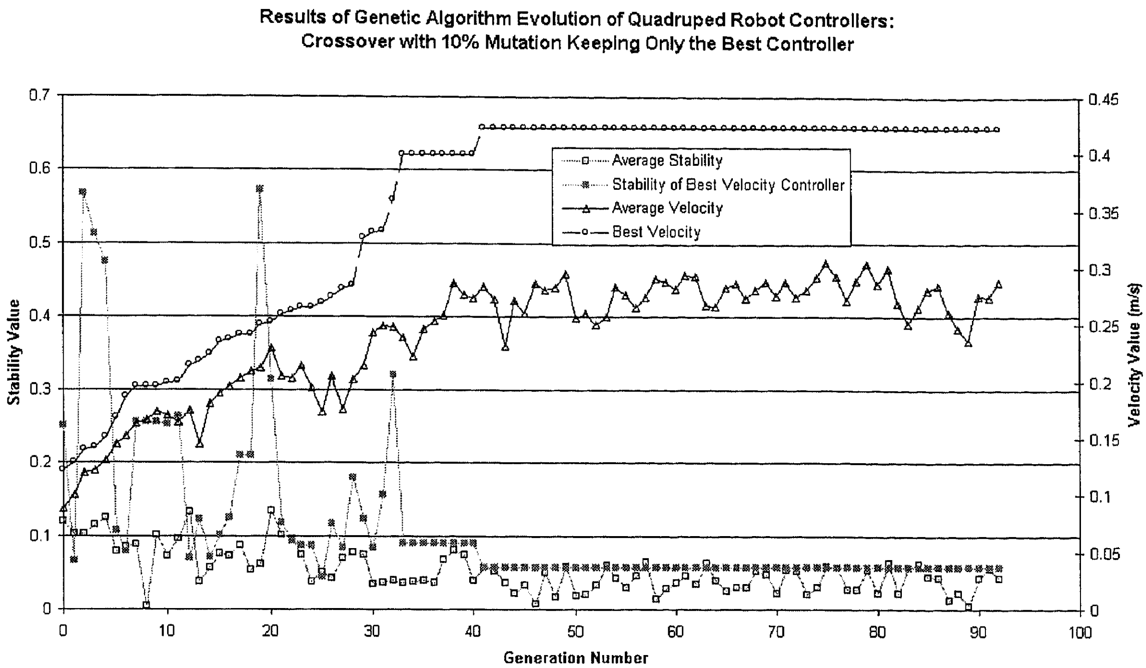

Figure 10 Results of evolutionary trial utilizing selection, mutation, and crossover with constant mutation rate.

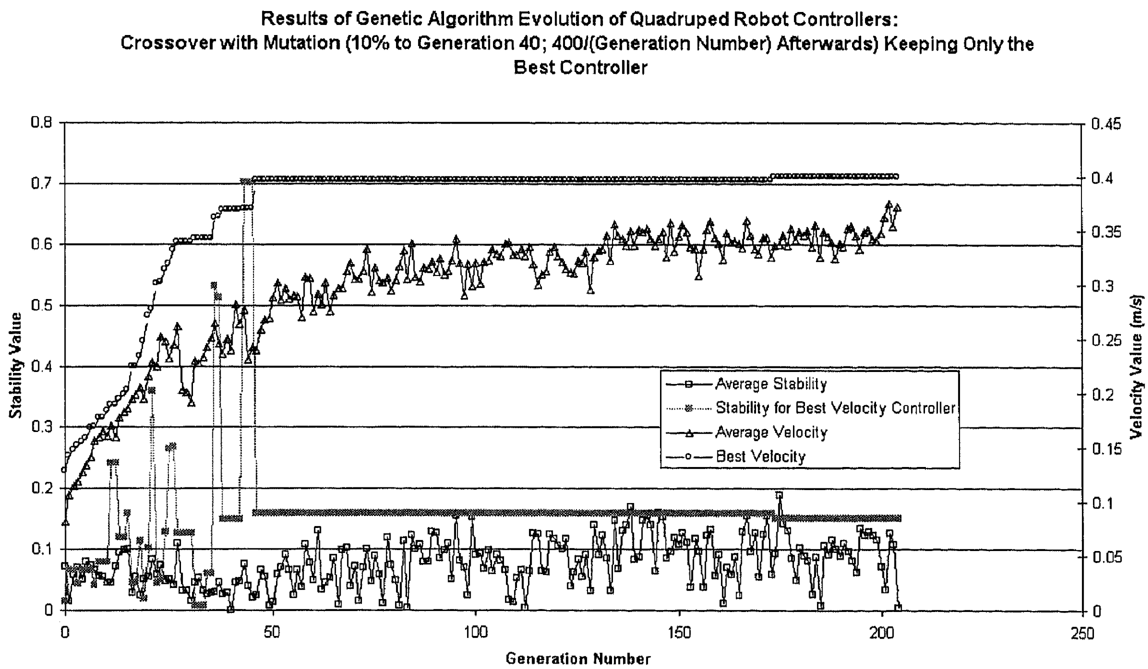

Figure 11 Results of evolutionary trial utilizing selection, mutation, and crossover with variable mutation rate.

The trial with the constant mutation rate appears to develop this controller more quickly; this difference can be attributed to the random initial population.

Another evolutionary trial was conducted which utilized selection, mutation, and crossover with the fitness function defined by Equation 3. This trial started with an initial mutation rate of 10% and changed to the mutation rate defined by Equation 5 after generation 40. The results of this trial are shown in Figure 12.

The fitness function given by Equation 3 produces a best controller with the same velocity as on developed with the fitness function given by Equation 1. The best controller in both cases had a velocity of 0.33 m/s. The fitness functions given by Equations 2 and 3, however, produce stability values for the highest velocity controllers which are about half of those for the fitness function given by Equation 1.

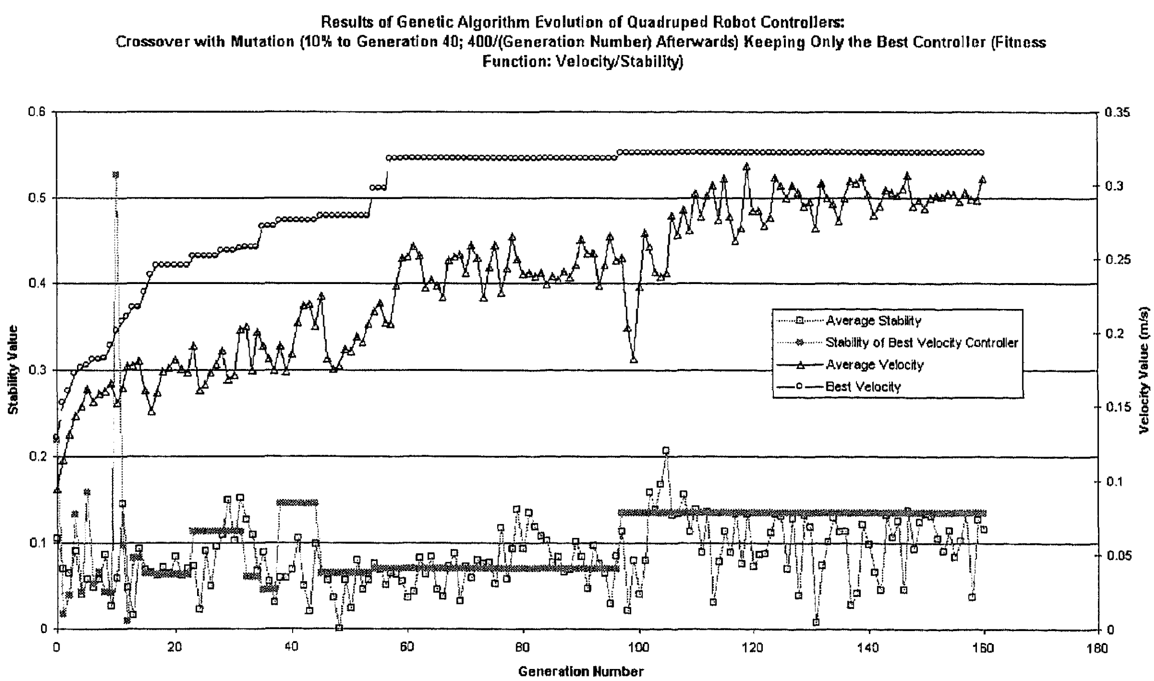

Figure 12 Results of evolutionary trial using the fitness function defined by Equation 3.

2.3.4 Comparison of Operator Combinations

The three above combinations of genetic algorithm operators provided noticeably different fitness function increase rates regardless of the fitness function. These fitness function increase rates are compared graphically in Figure 13. The addition of crossover to mutation and selection doubles or triples the fitness value increase rate.

An even better comparison of the three combinations of genetic algorithm operators can be seen in Figure 14. The best evolutionary trial for each of the three combinations is shown; all of the combinations utilize the fitness function given by Equation 2. The combination of selection, mutation, and crossover achieves the same velocity controller as the combination of selection and mutation; it just produces this controller in about 50 generations rather than 200. The addition of crossover provides a reduction of about 75% in the time required to develop optimal controllers.

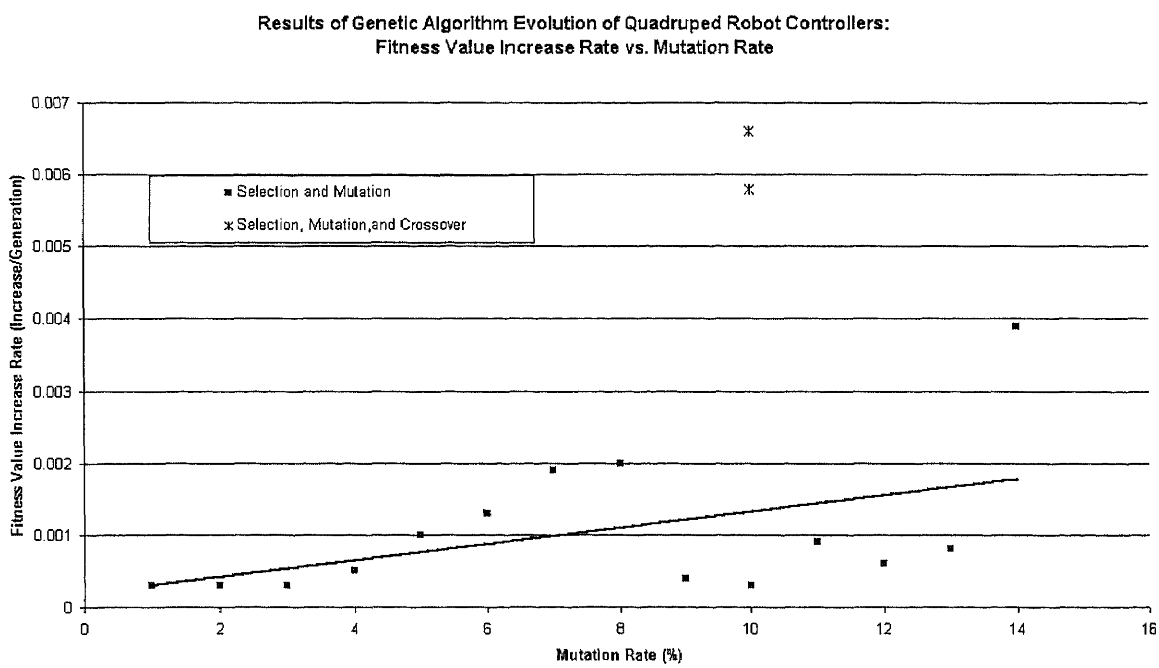

Figure 13 Fitness value increase rate vs. mutation rate.

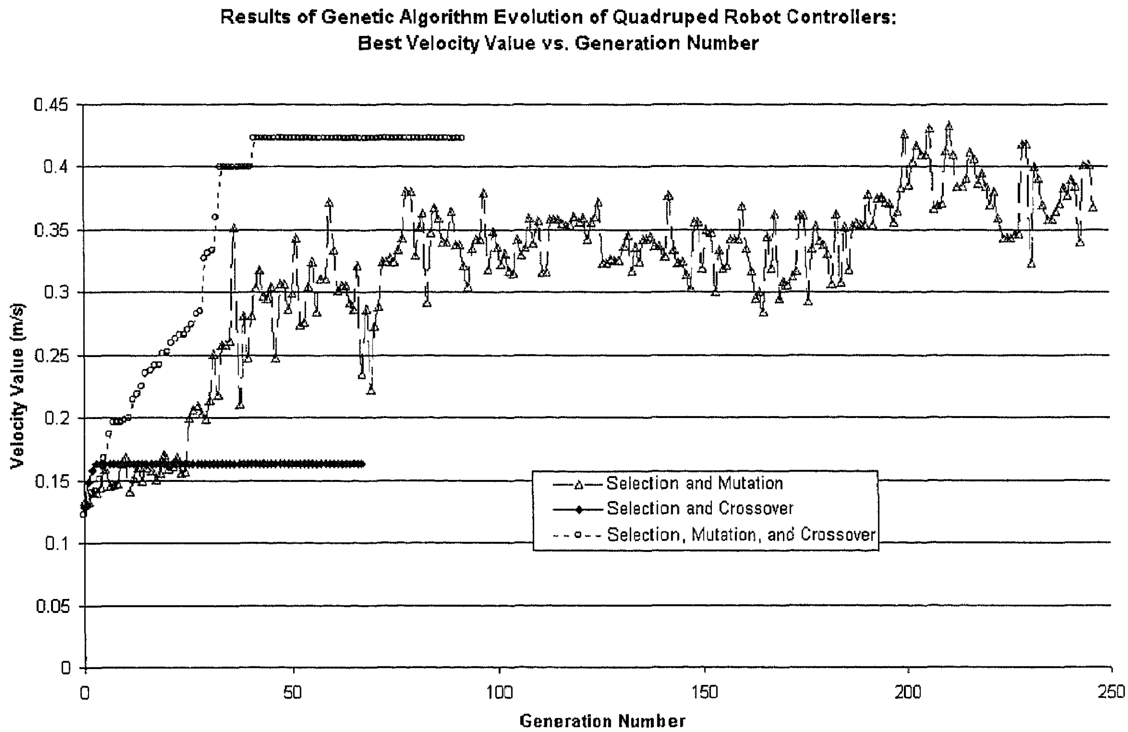

Figure 14 Comparison of results for the three combinations of genetic algorithm operators.

CHAPTER 3

GAIT VERIFICATION

3.1 Problem Statement

The next step in the development of a catalog of gaits for use in a dynamic gait optimization system is to verify that the gaits are suitable when applied to the physical robot. The suitability of the gait involves not only the stability, but also the robot's ability to step over small obstacles without changing gaits. The most stable gait and the highest velocity gait from each of the evolutionary trials were chosen for use in the dynamic gait optimization system, for a total of 22 gaits. Of these 22 gaits, 8 were randomly chosen for physical verification.

3.2 Robot Modifications/Repairs

Various modifications and repairs to the Arturo robot were required before the data collection necessary for gait verification could even be attempted. The first of these modifications was the extension of the Arturo robot's umbilical cord.

In order to verify the previously developed gaits, the Arturo robot needed to be capable of walking for at least 20 seconds so that the dynamics of the gait would stabilize (the first two steps are required to start the motion to ensure dynamic stability). In its original configuration, the Arturo robot had an umbilical cord about 6 feet in length, which allowed only 2-3 steps (4-6 seconds of walking time). The motor power cables were replaced with 18 foot cables made from 16 AWG wire. This rather large wire gauge was chosen to minimize electrical resistance between the servomotors and the power amplifiers. The signal cables (for the encoder signals) were replaced with 18 foot cables made from 20 AWG wire. At this point, the umbilical cord for the Arturo robot allowed 20 seconds of walking time for the fastest controllers.

The extension of the umbilical cord also required another modifications to the electrical system of the Arturo robot. It was discovered that the encoders used on each of the eight joints had a limit on the wire length of 6 feet. This limit needed to be extended in order to avoid burning out the electronics, so pull-up resistors needed to be added to all three encoder signal wires for each of the eight joints.

Repairs were also necessary on the mechanical drive system (Figure 15) for each joint. The gears and sprockets are all held in place by a single setscrew tightened against a flat in the shaft. Unfortunately, all of these setscrews were loose and in many cases had

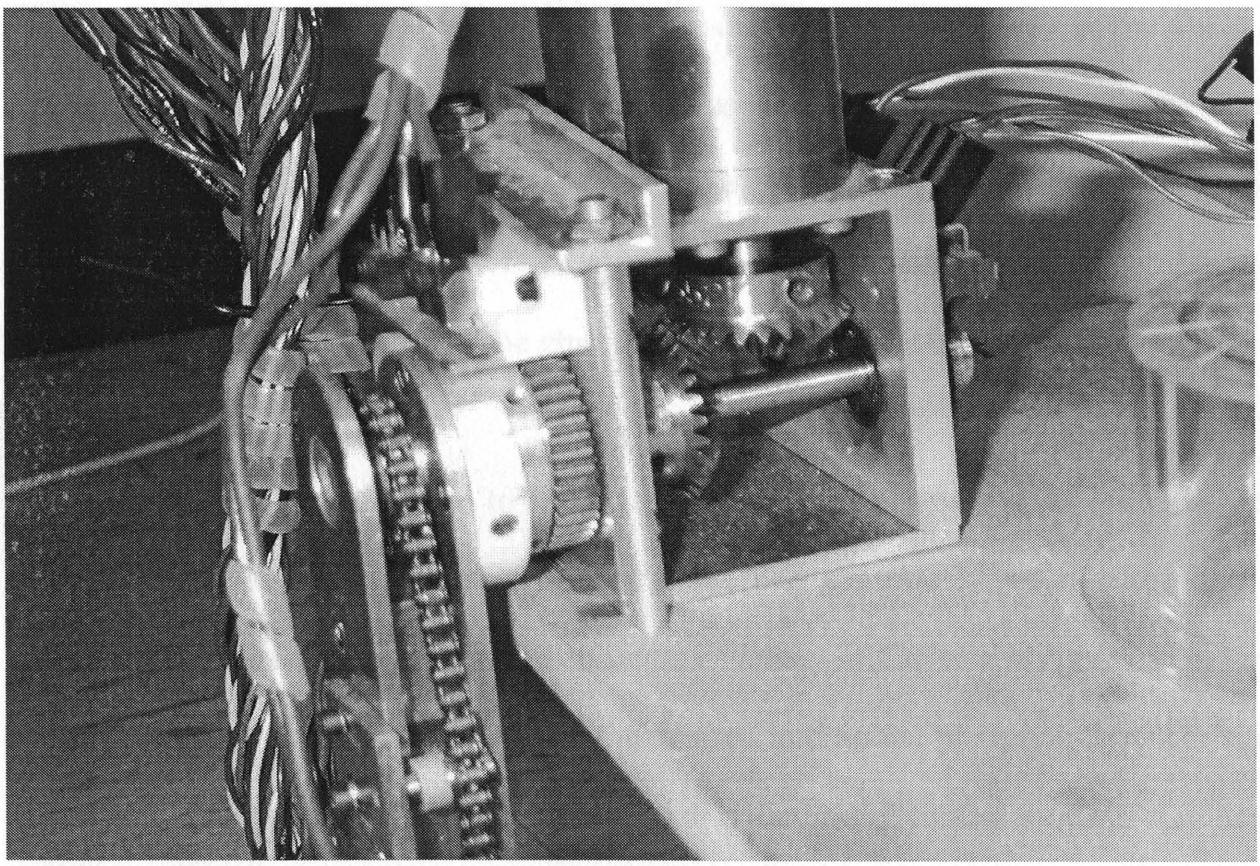

Figure 15 Mechanical drive system for Arturo robot.

allowed the gears to rotate with respect to the shaft. In most cases, tightening the setscrew while rotating the shaft (but not the gear) realigned the setscrew with the flat in the shaft. The setscrews were then tightened into place with a coating of Locktite thread sealant to prevent further loosening. On the left rear hip joint, however, the gear had rotated with respect to the bushing between the gear and the shaft. Since this bushing contained only a single hole, the gear had to be removed from the shaft and realigned with the hole in the bushing. This setscrew was also tightened into place with a liberal coating of Locktite. The right rear knee joint could not be tightened since the setscrew was stripped; fixing this problem would have required rebuilding the leg and was deemed impractical.

Another problem encountered with the drive system was slack in the chain drives for the knee joints. With use, the steel chains used to drive these joints tend to stretch and thence the knee joints become rather loose. The existing chain tensioners were too small to tighten the chain any further. A new set of chain tensioners was made, which removed virtually all of the looseness in the chain drive and allowed for further tensioning of the chain.

The final modification to the Arturo robot was the tuning of the PID filters for each LM628N-6 controller chip. The robot had previously been using the default parameters for each filter with questionable results. Initially, a program was written to move each joint about 15° and record the desired and actual positions from the controller chip for tuning purposes. Unfortunately, such a method of tuning requires that the Arturo robot be supported so that it does not fall over; this support changes the loads on the motors and therefore makes the tuning process ineffective. Better results were obtained when tuning based on data collected during an actual walking trial. The final PID filter values were:

$K_p = 700$

$K_d = 0$

$K_i = 100$

Integration Limit = 100

These values for the PID filter produced a very small following error; a sample of one of the graphs used for tuning these parameters is shown in Figure 16.

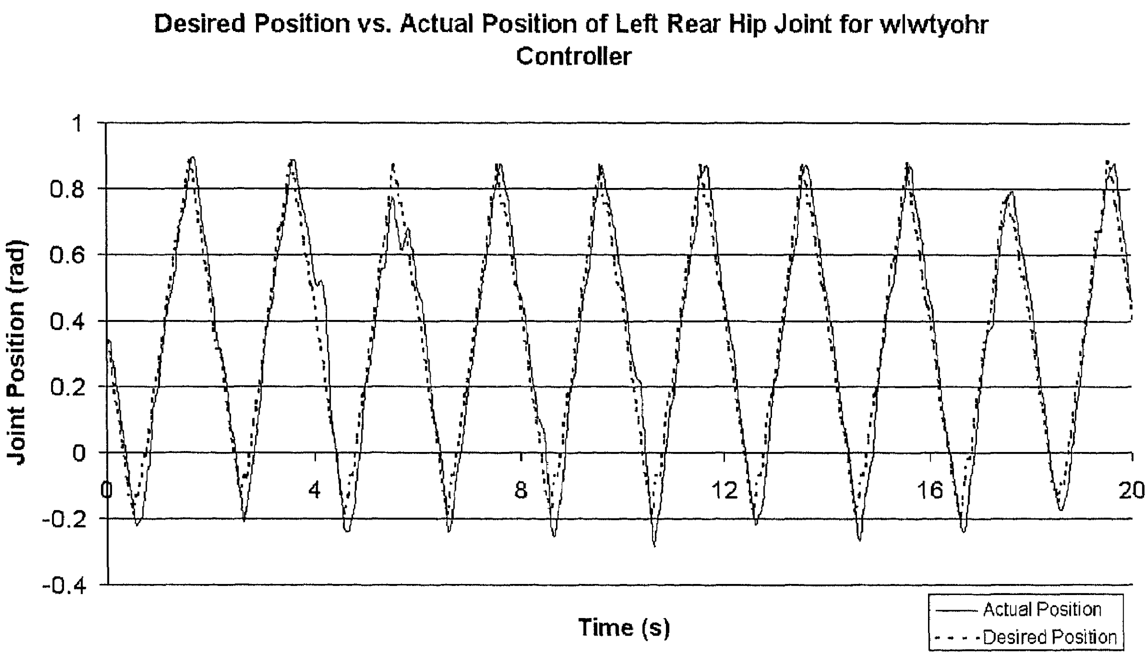

Figure 16 Sample tuning results for Arturo robot PID controllers.

3.3 Verification Trials

Now that the robot was in working order, the data collection for the verification trials could begin. The verification trial for each of the 8 controllers consisted of the following:

1. 3-5 runs across the linoleum floor of the lab.
2. At least one run across a piece of indoor-outdoor carpeting.
3. One run across the linoleum floor of the lab strewn with scraps of ¼" thick plywood (to simulate small obstacles).

For each of these trials, the following data was recorded at a rate of about 40Hz:

1. Current time
2. Calculated position of joint (from controller program)
3. Desired position of joint (from LM628N-6 chip)
4. Actual position of joint (from LM628N-6 chip)

Due to the rather large number of controllers to verify, the results of only a single controller are presented here. The results for the other trials (other than the gait itself) are very similar and are summarized in Table 1 at the end of this chapter. The robot controller that will be presented here is wl1beqxw, which has a stability value of 1.055 and a velocity value of 0.181548 m/s. The ideal movement of each of the eight joints is shown in Figure 17.

Figure 18 shows a trace of the actual and desired positions for the right front knee joint for the second trial on the lab floor. The ideal position curve for the joint is also superimposed on this graph. The actual and desired position curves are almost identical, showing that the motors are tuned correctly. The ideal position curve is slightly farther off, but still shows a very close physical reproduction of the ideal position curve. This close reproduction of the ideal position curve, however, is not consistent with all eight joints. For example, a similar curve for the right rear knee (Figure 19) shows some significant differences between the ideal and actual position curves. There are several

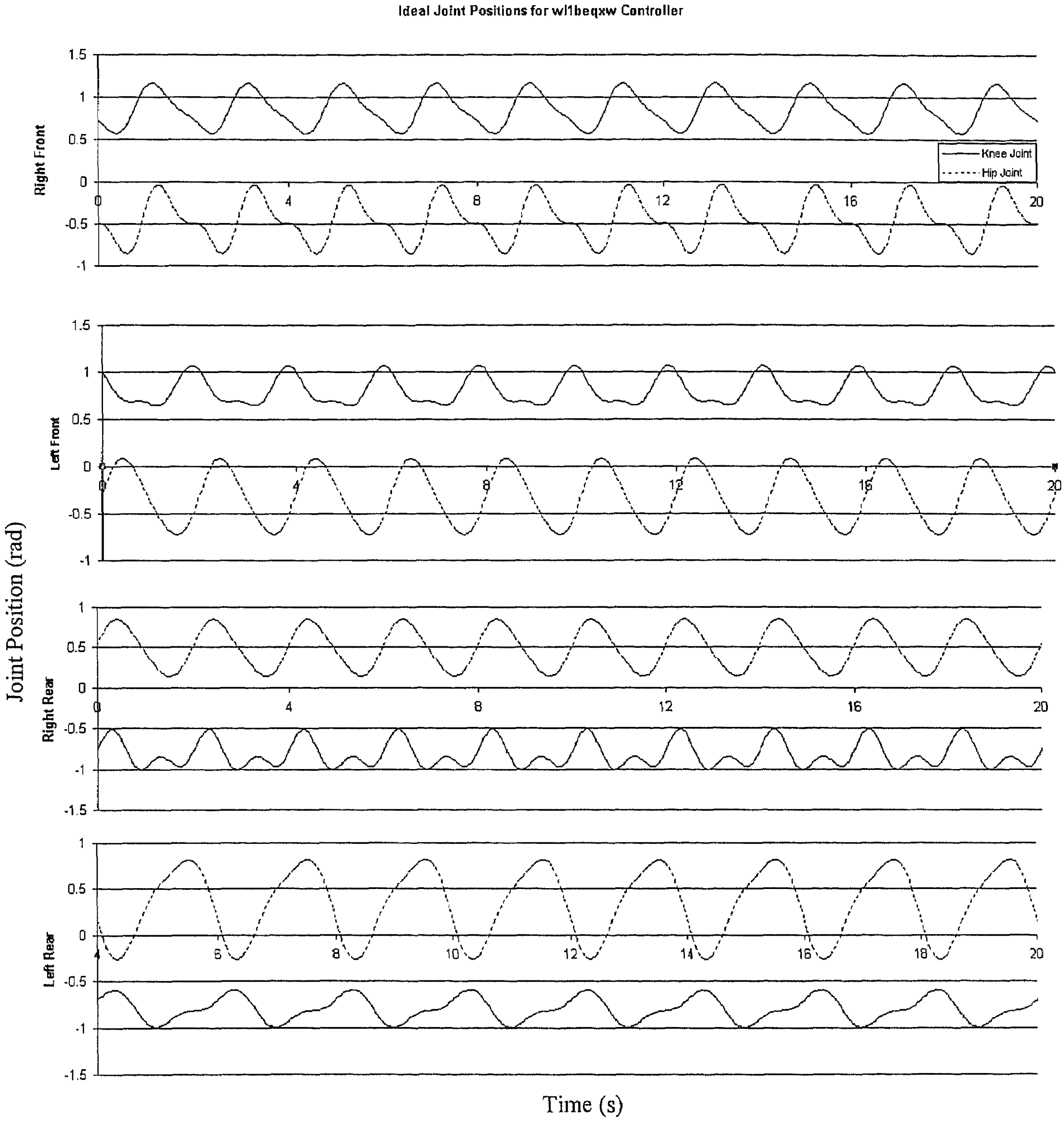

Figure 17 Ideal joint positions for wl1beqxw controller.

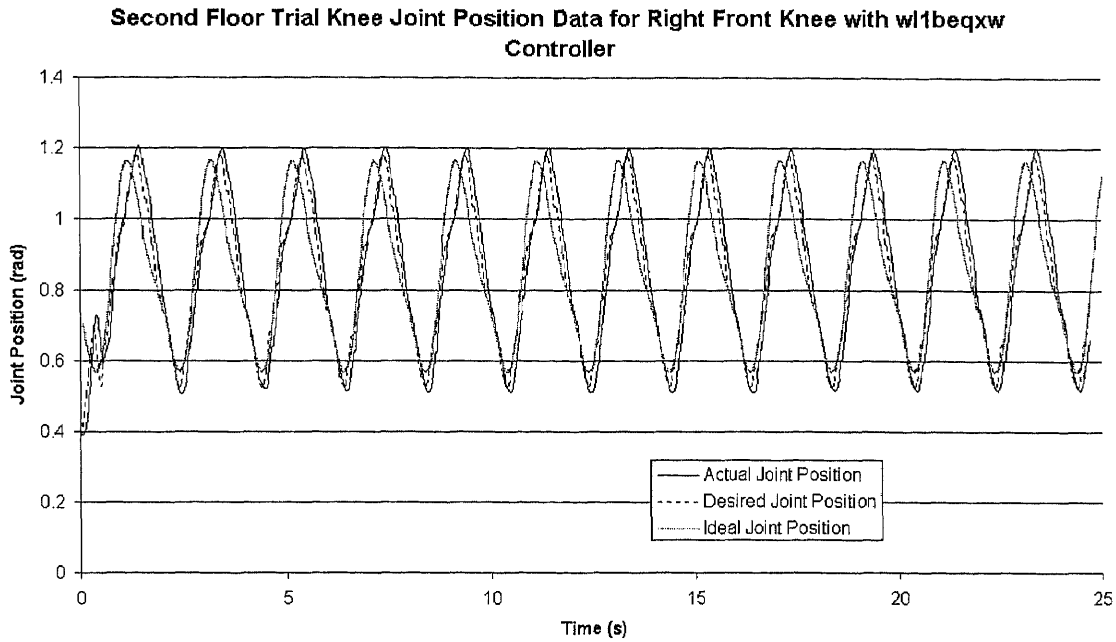

Figure 18 Right front knee position data for second floor trial.

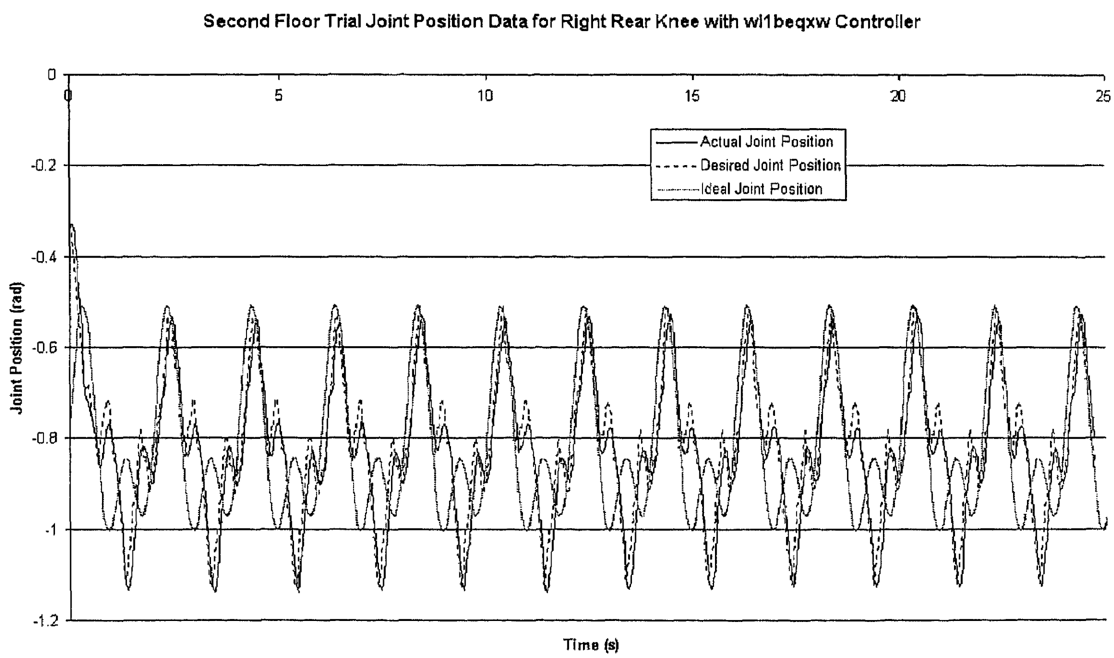

Figure 19 Right rear knee position data for second floor trial.

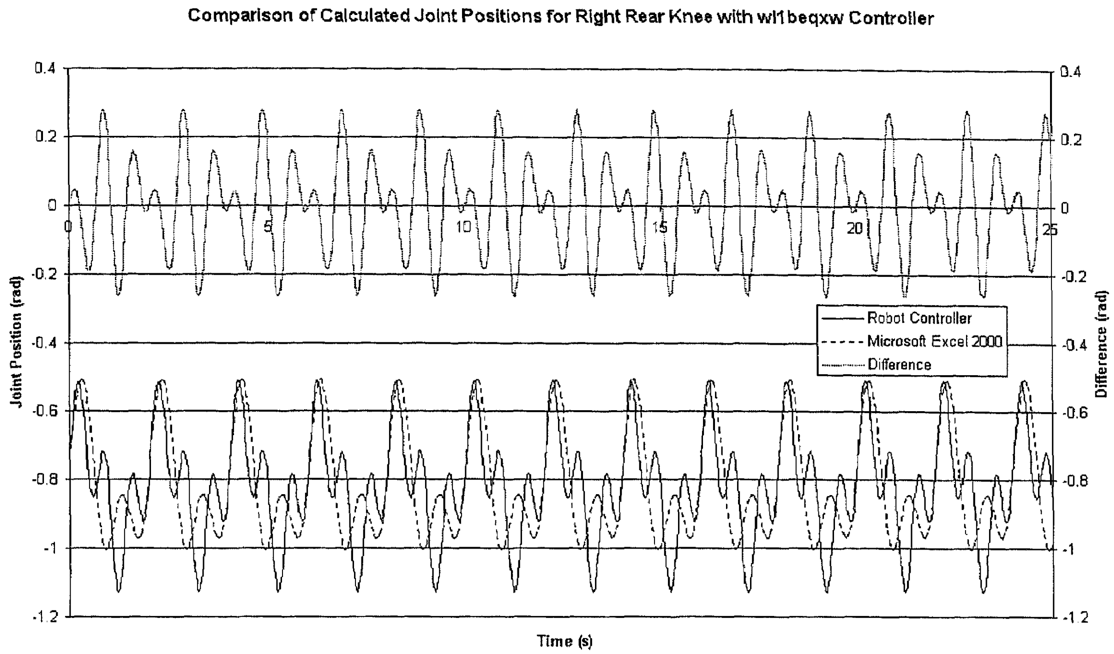

Figure 20 Difference in calculated joint positions for right rear knee.

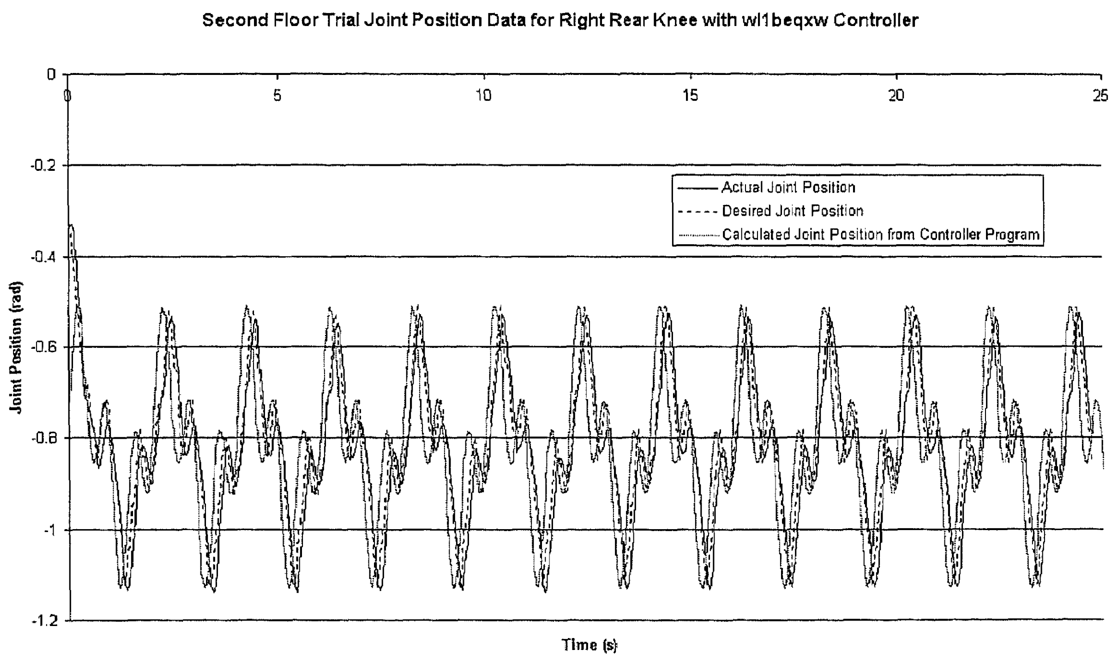

Figure 21 Actual and calculated joint positions from controller program for right rear knee.

sources for these errors. First, the velocity and acceleration limits on the servomotors prevent the actual robot joints from exactly following the higher frequency modes of the ideal position curve.

Second, and more significantly, is the difference in the values of the sine and cosine functions in Microsoft Excel 2000 (where the ideal position curves were calculated) and the math.h header file in Red Hat Linux 5.0 which was used to compile the controller program. The difference between these two calculated joint positions is shown rather clearly in Figure 20. The difference curve shown in Figure 20 has the same basic shape for all sets of calculated joint position curves regardless of joint or controller. If the actual joint position curves are plotted against the calculated joint positions from the controller program rather than the ideal joint positions calculated by Microsoft Excel 2000 (Figure 21), the only significant error is a slight phase shift. This phase shift is due to the delay inherent in the signal transmission from the controller program to the motor.

The wl1beqxw controller's relative stability and ability to step over small obstacles were also examined qualitatively. This particular controller had no problems with stability, even when stepping down from a height of ¼", but encountered problems when trying to step over obstacles of the same size. The front feet almost drag on the ground (which helps to maintain stability), which required the robot to make several attempts before stepping over the obstacle. The results from the other 8 verification trials are given in Table 1.

Table 1 Results summary for verification trials

Controller	wla14144	wla23930	wleyP4jj	wlkPI5sb	wl9sj2fa	wlwTyoHr	wlapxCuW	wl1Beqxw
Velocity Value (m/s)	0.295359	0.259935	0.324142	0.288323	0.171105	0.281075	0.303277	0.181548
Stability Value	0.59706	0.07301	0.58603	0.14818	0.94689	0.09877	0.37378	1.05518
Actual Velocity (m/s)	0.07	0.11	0.03	0	0.02	0.22	0.05	0.15
Actual Stability	Fair	Fair	Poor	None	Poor	Very good	Fair	Very good
Sensitivity to Surface Perturbations	High	High	High	N/A	Medium	Minimal	Medium	Minimal
Obstacle Handling	Fair	Good	Poor	N/A	Poor	Fair	Poor	Fair
Other Notes	Right rear hip very close to limit	Very bouncy	Very unstable	Does not walk	Right rear knee very close to limit	Tends towards right	Right rear knee very close to limit	Tends towards left

CHAPTER 4

SENSOR FEASIBILITY

4.1 Problem Statement

The next step in the implementation of a dynamic gait optimization system is the determination of the surface conditions so that the correct gait can be selected. The feasibility of two sensing methods was examined. The first of these methods involved the use of the position feedback from the servomotors on the robot, which would require no external sensors. The second method involved mounting piezoelectric crystals (specifically Leybold Inficon 6Mhz crystals) to the bottoms of the robot feet.

4.2 Position Feedback

The ability to sense the surface conditions using just the position feedback from the servomotors on the robot would be very advantageous – no external sensors would need to be mounted or read. Ideally, however, this method of sensing floor conditions should not be expected to be feasible – the controller for the joints is independent of the floor surface. The motors, since they are tuned correctly, should very closely follow the controller's calculated joint positions, regardless of the floor surface. It is, however, good research practice to verify such assumptions.

Using the data from the verification trial of the wl1beqxw controller in the previous chapter, the actual joint position curves for the different surfaces were compared. The graphical comparison for the right rear knee joint is shown in Figure 22. The actual joint position curves for the carpeted and obstacle-strewn surfaces practically overlay those curves for the linoleum floor surface. There are greater variances between the three trials on the linoleum floor than between these trials and those on carpeted and obstacle-strewn surfaces. Therefore, surface condition determination by the joint position feedback is not feasible.

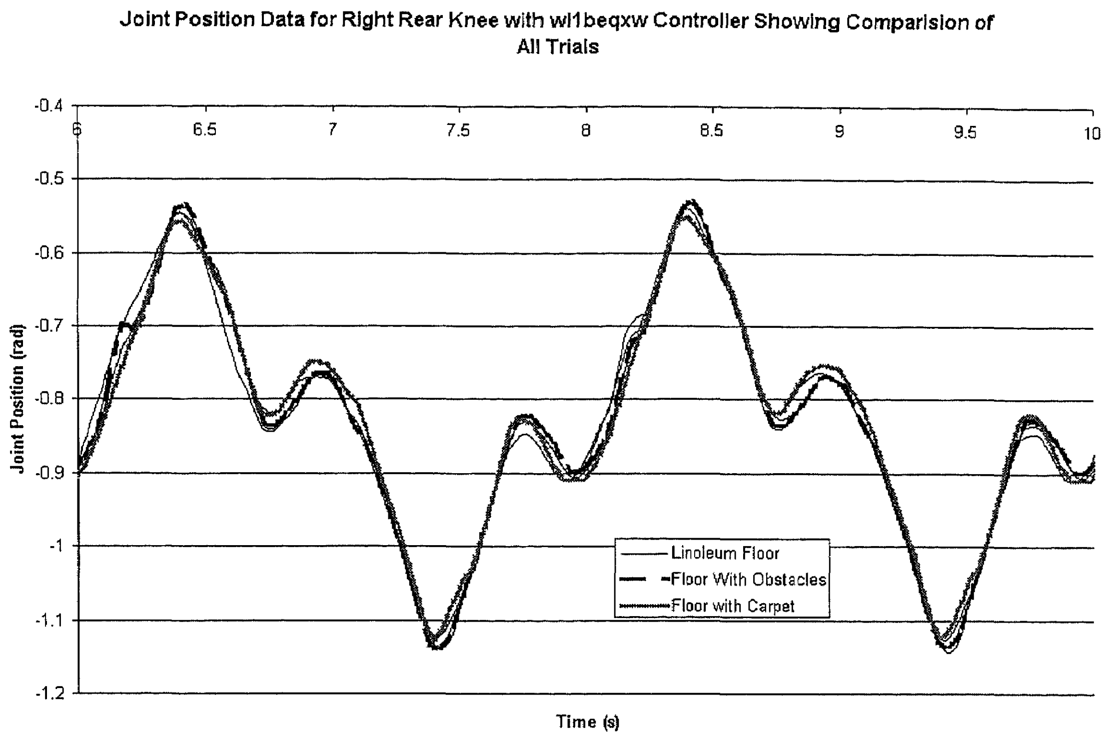

Figure 22 Comparison of actual joint position curves for different surfaces.

4.3 Piezoelectric Crystal

Piezoelectric crystals mounted on the robot feet would provide a pressure-time curve for each foot. Such a curve could conceivably be used to determine the differences between surface conditions; the curve for a soft surface such as sand would be significantly different than that for a hard surface such as the linoleum lab floor. Other phase shifts in the pressure-time could indicate obstacles or slipping on the floor.

The piezoelectric crystals chosen for this application were Leybold Inficon 6 Mhz quartz oscillators since they were currently being investigated for a similar application. These crystals are ½" in diameter and about 0.03" thick with gold electrodes on both planar surfaces. Since the piezoelectric properties of the quartz crystals were being utilized, a broken crystal would still produce an output (assuming that there are still leads connected to both electrodes) and provide a measure of fault-tolerance in the sensor system. Unfortunately, the crystals were also very fragile and shattered under minimal pressure if not placed on a perfectly flat surface.

The very fragile nature of the crystals basically deemed them infeasible for surface detection on the Arturo robot. The crystals could not be mounted to the outside of the rubber robot foot, since they would shatter at the first contact with the floor. Mounting on the inside of the foot would place the sensor on the exact bottom of the foot and would also require trimming 0.125" off of the diameter of the sensor. In many cases, the bottom of the robot foot does not really impact the floor; the side of the foot will hit the floor first. Thus the sensors would be of limited usefulness if mounted on the bottom of the foot. Due to the extensive amount of time required to seat the sensor perfectly flat on the bottom of the robot foot and the limited sensing abilities, these particular piezoelectric crystals were deemed impractical for sensing surface conditions.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

A catalog of gaits has been developed for use in a dynamic gait optimization system for the Arturo robot. Some of these gaits, which were developed with a genetic algorithm search and a motion simulator, showed significantly worse performance when applied to the physical robot rather than the motion simulator. Other controllers, such as wl1beqxw and wlwtyohr, performed fundamentally the same on both platforms, showing very good performance in both cases.

The two methods of surface determination that were examined were both deemed infeasible. The joint position feedback does allow the surface conditions to be determined; there are no significant differences in the joint position curves on different surfaces. The use of piezoelectric sensors (particularly Leybold Inficon 6 Mhz oscillators) on the robot feet was deemed infeasible due to their fragility and mounting difficulties.

5.2 Future Work

The actual implementation of a dynamic gait optimization system for the Arturo robot is left as a future project. In order to make such a system truly useful, gaits for use on rougher surfaces must also be developed and verified; this thesis concentrated solely on the development of gaits for use on flat surfaces. A method for reliably determining surface conditions autonomously is also necessary. Care must also be taken to ensure that the Arturo robot does not become unstable during the transition from one gait to another.

5.3 Suggested Improvements to Arturo Robot

There are several improvements that should be made to the Arturo robot before any significant further work is done. First, the body should be widened slightly to allow the servomotors to be mounted horizontally rather than vertically. The motors could then be mounted horizontally, eliminating the gearing currently necessary to create the 90° angle in the drive shaft for each joint. The servomotors could then be used to drive the joints directly.

Second, all shaft connections should be more permanently fixed in place. The current system of setscrews has a tendency allow the connections to loosen with use. The mounting of gears and sprockets on the motor shaft should be carried out with either pins or keys, which really prevent any rotational looseness in the connections.

Third, the encoders should be mounted directly to the servomotors, not to the gearbox output shaft. This method of mounting the encoders would significantly increase the encoder resolution and therefore the accuracy of the joint position data. The current system only allows about 600 encoder counts for the entire range of motion of each joint; mounting the encoder on the motor directly would increase this to about 30,000.

REFERENCES

1. S. Azarm, R.J. Chen, and L.W. Tsai, "Walking Robot (Terrapin 1)," http://www.isr.umd.edu:80/TechReports/ISR/1987/TR_87-129/TR_87-129.pdf, April 1987.

2. L. Eshelman and J. Schaffer, "Crossover's Niche," *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 9-14, July 1993.

3. G. Grasso, "Genetically Evolved Dynamic Control for Quadruped Robot Walking," Doctoral Dissertation, Department of Computer and Information Science, New Jersey Institute of Technology, Newark, NJ, January 2000.

4. S. Hirose, M. Nose, H. Kikuchi, and Y. Umetani, "Adaptive Gait Control of a Quadruped Walking Vehicle," *Robotics Research: The First International Symposium*, pp. 253-277, August 1983.

5. S. Hirose, T. Masui, H. Kikuchi, Y. Fukada, and Y. Umetani, "TITAN III: A Quadruped Walking Vehicle," *Robotics Research: The Second International Symposium*, pp. 325-331, August 1984.

6. H. Kimura, K. Sakurama, and S. Akiyama, "Dynamic Walking and Running of the Quadruped Using Neural Oscillator," http://www.kimura.is.ued.ac.jp/research/Patrush/PAPER/iros98.pdf, June 1998.

7. H. Miura, I. Shimoyama, M. Mitsuishi, and H. Kimura, "Dynamical Walk of Quadruped Robot (COLLIE-1)," *Robotics Research: The Second International Symposium*, pp. 317-324, August 1984.

8. A. Nagakubo and S. Hirose, "Walking and Running of the Quadruped Wall-Climbing Robot," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 2, pp.1005-1012, May 1994.

9. M. Raibert, *Legged Robots That Balance*, The MIT Press, Cambridge, Massachusetts, 1986.

10. G. Rawlins, *Foundations of Genetic Algorithms*, Morgan Kaufmann Publishers, Inc., San Mateo, California, 1991.

11. G. Sukhatme, "The Design and Control of a Prototype Quadruped Microrover," *Autonomous Robots*, vol. 4, no. 2, pp. 211-220, April 1997.

REFERENCES
(Continued)

12. G. Sukhatme, S. Brizius, S. Cozy, and G. Bekey, "A Strategy for Quadruped Walking on Uneven Terrain," *Proceedings of the 1997 8th International Conference on Advanced Robotics*, pp. 291-296, July 1997.

13. D. Tate and A. Smith, "Expected Allele Coverage and the Role of Mutation in Genetic Algorithms," *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 31-37, July 1993.

14. P. Trachtman, "Redefining Robots," *Smithsonian*, vol. 30, no. 11, pp. 96-112, February 2000.

15. D. Wettergreen and C. Thorpe, "Gait Generation for Legged Robots," http://www.ri.cmu.edu/pub_files/pub2/wettergreen_david_1992_1.pdf, July 1992.

16. D. Wettergreen, "Robotic Walking in Natural Terrain: Gait Planning and Behavior-Based Control for Statically-Stable Walking Robots," Doctoral Dissertation, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1995.

17. K. Yoneda, H. Hyama, S. Hirose, "Sky-Hook Suspension Control of a Quadruped Walking Vehicle," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 999-1004, May 1994.