Gait Generation and Optimization for Legged Robots

Joel D. Weingarten† Martin Buehler§ Richard E. Groff† Daniel E. Koditschek†
jweingar@umich.edu buehler@cim.mcgill.ca regroff@umich.edu kod@umich.edu
†Department of Electrical Engineering and Computer Science, The University of Michigan
§Center for Intelligent Machines, Ambulatory Robotics Laboratory, McGill University

***Abstract*— This paper presents a general framework for representing and generating gaits for legged robots. We introduce a convenient parametrization of gait generators as dynamical systems possessing specified stable limit cycles over an appropriate torus. Inspired by biology, this parametrization affords a continuous selection of operation within a coordination design plane spanned by axes that determine the mix of "feedforward/feedback" and centralized/decentralized" control. Applying optimization to the parameterized gait generation system allowed RHex, our robotic hexapod, to learn new gaits demonstrating significant performance increases. For example, RHex can now run at 2.4m/s (up from 0.8m/s), run with a specific resistance of 0.6 (down from 2.0), climb 45° inclines (up from 25°), and traverse 35° inclines (up from 15°).**

I. Introduction

Legged systems locomote by producing periodic leg motions, whose intermittent contact with the ground imposes recurring reaction forces that propel the body mass center. The richly variable nature of possible leg cycles maps onto a huge and usefully diverse range of possible motions. In biological legged systems, this rich behavioral suite is well expressed by the profusion of qualitatively distinct and independently named modes of movement or "gaits:" e.g. walk, trot, canter, gallop, pronk, and so on. This paper presents a framework for developing a broad palette of "gaits" for our hexapod robot RHex.

RHex (see Figure 1) is a power- and computation-autonomous hexapod robot [1]. Inspired by cockroach locomotion, RHex features compliant legs and a simple mechanical design. Each leg has a single actuated mechanical degree of freedom and can rotate fully about the hip joint. A growing body of evidence suggests that high speed cockroach runners, whose limbs must move at rates faster than the speed at which neural signals can travel in the body, employ an open loop feedforward style of gait control [2]. Further inspired by this principle of cockroach locomotion, the original control design for RHex employed an essentially open loop control strategy incorporating hand-tuned reference trajectories for the leg joint angles, the "clock" signal depicted in Figure 2. In this "alternating tripod" scheme, the same reference trajectory is applied to each leg, but the reference trajectory of the left tripod is 180 degrees out of phase from the reference trajectory of the right tripod. See Figure 4 *left* for the tripods of the walking gait. Intuitively, the reference trajectory makes the legs move slower while they are are putatively on ground, and faster while recirculating through the air. At each hip, a single actuator applies torque to a leg shaft through a local PD controller that regulates the difference between the reference signal and the motor shaft angle and velocity. Since this local hip feedback provides no information about the true state of the leg (e.g., is it in stance or in swing?), or the body, the robot operates in a task open loop manner.

Even with such a simple control strategy, RHex displays impressive mobility, traversing obstacles higher than its leg length [1] at speeds well above 0.5m/s over a wide variety of different terrains. Moreover, hand-built variations on the original clock trajectory of Figure 2 have led to a variety of other gaits such as pronking, stair climbing, and hill climbing. Empirically, a given set of parameters performs quite differently on different surfaces – linoleum, concrete, pavement, gravel, grass, etc. Conversely, on a given surface we have found that slight variations in parameter settings can have significant impact on performance. Our present insight concerning the relationship between clock excitation signal and the physical limit cycle it elicits over any specific terrain remains quite limited. Moreover, the parameter space that shapes the performance relevant details of any given clock signal is quite large. These observations motivate the central focus of this paper — the development of a more systematic method for generating gaits and a more automated method for tuning the accompanying gait parameters.

The feedforward paradigm has proven extremely successful in the control of RHex. Nevertheless, it seems intuitively clear that the introduction of information about the legs' states and the robots' body should yield still greater performance benefits. But what information should be collected, and how should it be used to adjust the reference clock? The designer is presented with at least two distinct architectural "axes" along which to locate a gait coordination controller. There is a choice to be made concerning the relative strength of the feedback and feedforward components. And there is an independent choice of how centrally to bring to bear the information originating at more distal body locations. From the biological literature [3], [4], [5], [6], [7], [8], [9], it seems clear that different animals operate in different regions of this design space for different operating regimes. Biology inspires our search for a parameterized gait generation system that permits exploration of the feedforward-feedback and centralized-decentralized axes of the controls space.

The present paper pursues a general framework for representing and parameterizing this plane of coordination design

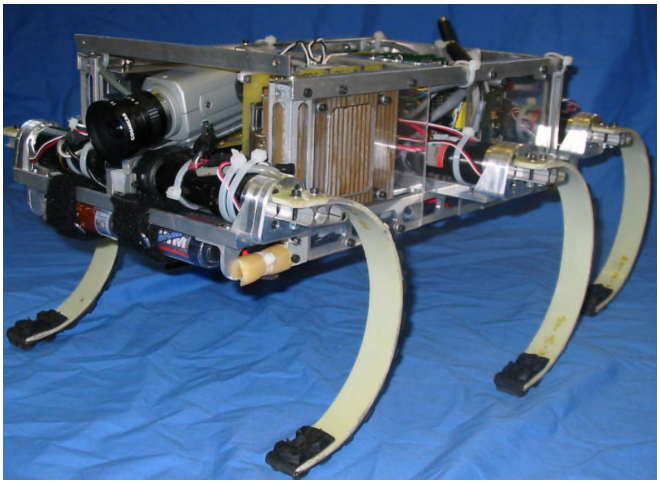

Fig. 1. RHex is a power- and computation-autonomous robotic hexapod, featuring compliant legs and a simple mechanical design. The chassis measures 48cm×22cm×12.5cm, and the distance from hip to ground in normal standing posture is 15.5cm.

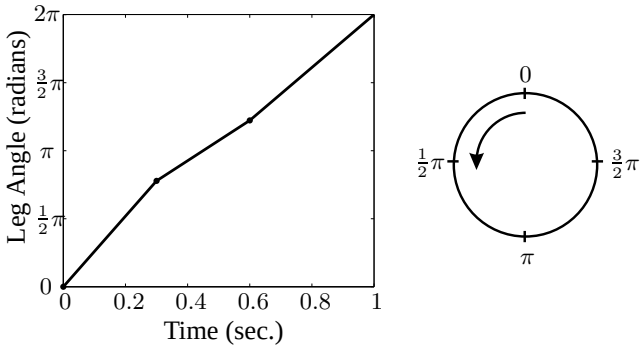

Fig. 2. *Left* - The "clock" signal that drives the legs for centralized-feedforward walking. *Right* - angle naming convention

architectures first introduced in [10], [11]. In this work, we are concerned to expand systematically the repertoire of gaits available to RHex and introduce an automated procedure for their tuning in any specific environment. Section II defines gait more carefully and presents the gait generator, a parameterized system that allows a gait to be shaped and permits exploration of the coordination plane. Section III presents our approach to gait optimization. Section IV presents the results of a series of gait optimization experiments performed on our robotic platform RHex.

II. Gait Generation System

Gaits and gait generation have been addressed variously in the literature, without consensus on a single definition for either. "Gait" sometimes encompasses just the footfall pattern of the behavior [12], and other times includes the the full kinematic trajectories of the joints. We will use the notion informally to denote the particular limit cycle of the gait generator. Our focus on the phase space representation of gait (i.e., the k-torus to be introduced below) seems most natural given the morphology of our robotic platform, RHex, in which the legs may be rotated arbitrarily often about the hip, as compared to many legged systems where the joints move within some constrained set of angles. There are many alternative frameworks, perhaps the most interesting and most comprehensively studied focusing on the role of symmetry rather than morphology [13].

Define the *gait state space* $\mathcal{G}$ to be the set of possible leg angles, in the present case $\mathcal{G} = T^k$, the k-fold cross product of the circle (where k is the number of legs) known as the k-torus[1]. A *gait* is a closed path, or cycle, in $\mathcal{G}$. Under this definition qualitatively similar behaviors, e.g. having identical foot fall patterns, but generated by different trajectories through gait state space are considered to be different gaits.

The need for leg coordination inclines us toward dynamical systems for gait generation. In our architecture, we define a dynamical gait generator as a dynamical system on $\mathcal{G}$ that, when decoupled from the environment, exhibits a stable limit cycle that we will call the generated gait. The state of the gait generator, $\phi \in \mathcal{G}$, serves as an online reference trajectory for the actual leg angle, $\theta \in \mathcal{G}$, which is tracked using standard PD control,

$$\ddot{\theta} = -k_p(\theta - \phi) - k_d(\dot{\theta} - \dot{\phi}) \tag{1}$$

where k_p and k_d are the PD gains.

Thus it remains to find a parameterized family of dynamical systems for ϕ with stable limit cycles (when decoupled from the environment) that offers parametric freedom over the shape of the limit cycle, as well as over the centralized-decentralized and feedforward-feedback nature of the coordination.

As a tool to this end, we introduce the clock space, $\mathcal{C} = T^k$. A stereotypical "model" gait, generated by a dynamical system, $\dot{s} = g(s, \theta)$, $s \in \mathcal{C}$, serves as a representative for the homotopy class — the underlying "rhythm" or footfall pattern — of the desired gait. . A deformation of $\mathcal{C}$ into $\mathcal{G}$ is used to generate topologically conjugate dynamics in the gait space, $\mathcal{G}$, thereby imposing upon the model rhythm the details of relative leg phase timing as may be required by the particular terrain through which the gait is being driven. The conjugate dynamical system will have also have a stable limit cycle that is the deformation of the template limit cycle. If the deformation is parameterized, then the shape of the limit cycle in $\mathcal{G}$, i.e. the gait, can be tuned by modifying the parameters of the deformation.

A. A Model Gait Generator

The dynamical system that generates the template in the clock space is adapted from the work of Klavins [11], [14]. Given the rotation vector [2], $A \in \mathbb{Z}^k$, of the desired limit cycle, Klavins uses a combination of a potential field with a global drift term to create a dynamical system with a stable limit cycle that is a straight line (mod 1) parallel to A. (Thus the limit cycle has rotation vector A.) The clock dynamics in the present

[1] We represent T^k as $\mathbb{R}^k/\mathbb{Z}^k$, equivalent to the unit box $[0, 1]^k$ with opposing faces identified.

[2] The rotation vector of a closed path on the torus gives the number of times A_i that the path wraps around the i^{th} dimension. In 2 dimensions, the rotation vector is sometimes replaced with the winding number, which is the ratio A_2/A_1

framework are given by

$$\begin{aligned} \dot{s} &= W + k_{cd}\nabla\Psi + k_{ff}\kappa(s,\theta) \qquad (2)\\ \text{where} \quad \Psi(s) &= \sum_{i<j} \cos\left(2\pi\left(A_i s_j - A_j s_i\right)\right). \end{aligned}$$

where $s = [s_1 \cdots s_k]^{\mathrm{T}} \in \mathcal{C}$ is the state of clock. $W = \omega A$ is the global drift term, where ω is the speed of the clock and A is the rotation vector of the limit cycle. The period of the limit cycle can be adjusted by changing ω. The potential term, $k_{cd}\nabla\Psi$, $k_{cd} \geq 0$, accounts for coordination between the clock variables. The rate at which trajectories in $\mathcal{C}$ converge to the limit cycle can be increased by increasing k_{cd}. The remaining term, $k_{ff}\kappa(s,\theta)$, $k_{ff} \geq 0$, accounts for coordination between a clock variable and its associated leg. Notice that $s \in \mathcal{C}$ and $\theta \in \mathcal{G}$, but to appropriately compare these states, they should be transformed to be in the same space. For this reason, the form of κ will be presented after introducing the deformation of $\mathcal{C}$ into $\mathcal{G}$.

1) A Conjugate System for Introducing Feedback: The deformation between $\mathcal{C}$ and $\mathcal{G}$ is formally a homeomorphism, a function that is continuous, invertible, and has a continuous inverse. Let $h : \mathcal{C} \longrightarrow \mathcal{G}$ be a homeomorphism that transforms the template limit cycle in $\mathcal{C}$ into the desired gait in $\mathcal{G}$. Then h can be used to form a dynamical system conjugate [3] to $\dot{s} = g(s,\theta)$. For technical reasons, rather than working with the conjugate system directly, we will continue to write the dynamics on $\mathcal{C}$ and apply h to follow the evolution in $\mathcal{G}$. The gait generator state is given by

$$\phi = h(s) \qquad (3)$$

The homeomorphism h suggests the appropriate form for κ, namely

$$\kappa(s,\theta) = -\sin\left(2\pi(s - h^{-1}(\theta))\right), \qquad (4)$$

where $\sin$ of a vector is taken component-wise and is introduced since every continuous function on T^k must be k-periodic over the covering space, $\mathbb{R}^k$. Notice that $\theta \in \mathcal{G}$ has been transformed through h^{-1} into the corresponding point in $\mathcal{C}$, so that the difference $s - h^{-1}(\theta)$ serves as a "clock phase" error in the model space. The gain k_{ff} offers a tradeoff between the physical leg tracking the gait generator (k_{ff} low) and the gait generator tracking the physical leg (k_{ff} high).

It is instructive to examine the behavior of this system at a few points in (k_{ff}, k_{cd}) space. When $k_{ff} = 0$, the gait generator is decoupled from the input of the actual leg angles θ. In this case s will converge to the template gait, with the speed of convergence dependent on the magnitude of k_{cd}. Moreover, if s is on the limit cycle, then it will remain on the limit cycle indefinitely, since no perturbations can enter to move it away. The gait generator state ϕ will produce the time trajectory specified by $h(s)$. This replicates the pure feedforward paradigm initially employed on RHex.

When $k_{ff} > 0$, the gait generator becomes coupled to the environment. Though the coupled system can, and empirically does, fall into a limit cycle, the projection of that limit cycle onto s will not in general match the template limit cycle. This is due to periodic perturbations caused by interaction with the environment.

[3] One typically employs a diffeomorphism, a homeomorphism that is smooth and has a smooth inverse, when defining a conjugate dynamical system. Without smoothness, the conjugate vector field $Dh \circ g \circ h^{-1}$ is not well defined. Even so, the flow can still be defined and similar results obtained.

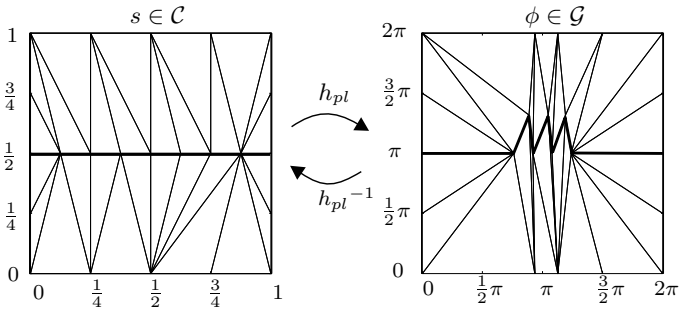

Fig. 3. An example of $h_{pl} : \mathcal{C} \longrightarrow \mathcal{G}$ for the hill traversal gait. The model gait and generated gait are indicated by thicker lines. In this example, the vertices of h_{pl} were chosen to lie on the gaits, as discussed in Section III.

2) Construction of Deformations: We now consider the problem of parameterizing the homeomorphism, h, required to implement this scheme. In principle, any finitely parameterized family of homeomorphisms could be used, but unfortunately few computationally effective (i.e. finitely parametrized) instances are available. With most parameterized function families, such as polynomials, radial basis functions, neural networks, etc., it is difficult just to determine whether a given parameterization yields a function that is invertible over its range, let alone whether it is a homeomorphism between the domain and codomain. We choose to use piecewise linear homeomorphisms (PLH) to represent h, and indicate this by writing h_{pl}. The PL functions are, to the best of the authors' knowledge, the most flexible finitely parameterized class of functions that can be verified to be one-to-one and onto. Moreover, a PL function can be inverted in closed form and the domain and range may be determined directly from the parameterization.

A PLH can be parameterized by a triplet (P, Q, T), where P is the set of domain vertices, Q is the set of corresponding codomain vertices, and T is a triangulation of P, represented as a list of sets, each set representing a k-simplex of the triangulation and containing $k+1$ indices of vertices of P. An example of a PLH is provided in Fig. 3. For more on PLH representations and computation, see [15], [16].

The complete system, gait generator coupled with physical legs, is given by Eqns 1, 2, 3 and 4. The system is parameterized by the coordination gains k_{cd} and k_{ff}, the PD gains k_p and k_d, and the clock speed ω have been discussed, as well the parameterization of h_{pl}. The rotation vector A enters as discrete parameter. This parameter set is larger, much larger if h_{pl} has many vertices, than originally employed in centralized-feedforward control of RHex. As the number of parameters increases, finding good operating regions in the parameter space can potentially become more difficult.

III. Gait Optimization

The gait generator from the previous section has a variety of parameters that can be adjusted. Even in the initial control strategy for RHex, pure feedforward ($k_{ff} = 0$) and centralized control ($k_{cd} \to \infty$), hand-tuning parameters showed demonstrable

improvement. Using intuition, an experienced designer can often find a gait for a given task which works, but improving the performance of the gait is a tedious and time-consuming task, motivating an automated method for optimizing, or "learning," a gait with respect to some performance criterion. Evaluating the performance criterion, discussed in more detail below, at a given gait involves running a series of robot experiments with the gait while measuring, for example, speed, power consumption, accuracy with respect to goal location, etc.

RHexLib, the software architecture for controlling RHex, has been designed such that the same code can be interfaced with the RHex robot or with a dynamical simulation of RHex called SimSect [17]. This simulation environment is used for verifying controllers and testing ideas for new behaviors. Unfortunately performing gait optimization in simulation and then applying the parameter set in the physical robot does not generally produce similar behavior. In a legged robotic system, especially one featuring compliant legs such as RHex, it is fundamentally difficult to obtain accurate models of, for example, actuators, nonlinear springs and damping in the legs, varying friction coefficients, and complex ground-body interactions. While the simulation environment is invaluable for prototyping code and testing proof-of-concept behaviors, it is nearly useless for optimization purposes.

Thus gait optimization is performed using the physical robot, from which arise a number of difficulties. A given gait is evaluated through an experiment, which consists of several of trials (we typically run two duplicates). typically 2. The performance criterion is computed for each trial and averaged to give the performance of the gait. Running multiple trials on the same gait is necessary to reduce the noise in the measurement of the performance criterion, which adversely affects the optimization procedure. To illustrate, consider a trial for the walking or running gaits. A trial involves running the robot between two lines placed 8m apart on the terrain. The operator measures by hand the time it takes to traverse the distance, while the robot tracks other factors, such as power consumption. Inaccuracies in operator timing and unmodeled operator intervention such as steering (necessary when the robot is perturbed by the environment or when a gait is not very stable), introduce variability in addition to the natural variability of the physical system such as changes in leg stiffness as the materials degrade, varying estimates of power usage as battery levels change.

We have chosen the Nelder-Mead algorithm [18], a derivative free simplex method for scalar function optimization, to perform gait optimization. While little is known analytically about the performance of Nelder-Mead, it has been empirically observed to perform well on a wide range of optimization problems and is straightforward to implement. A derivative free approach to hill climibing is desired in this setting because experimental variability makes the approximation of gradients difficult[4]. Nelder-Mead is designed for use on Euclidean space. The parameters k_p, k_d, k_{ff}, and k_{cd}, are in $\mathbb{R}_+$, and so are Euclidean with a simple constraint, but the space of parameters for h_{pl} is much more complicated. If vertices in the domain or codomain move such that simplices in the respective triangulations tangle, then h_{pl} fails to be a homeomorphism, and can even fail to be a function. Though the parameter space for valid homeomorphisms is geometrically well defined, it is difficult to write down explicitly, and even more difficult to incorporate into an optimization procedure. At present, this difficulty is avoided by starting Nelder-Mead with a relatively small initial simplex, having observed in this application that the search tends to be local, avoiding tangles. In the future, we will seek a more powerful way of dealing with the complexities of the parameterization of piecewise linear homeomorphisms.

Gait optimization on $\mathcal{G} = T^6$ remains a difficult task, because a large number of vertices, hence parameters, are required to represent a piecewise linear homeomorphism on T^6. More parameters to be optimized means more data is needed to optimize them, and since experiments are expensive with respect to robot and operator time, several steps have been taken to reduce the number of parameters significantly.

In order to reduce parameters, RHex is modeled as a virtual biped, simplifying $\mathcal{G} = T^6$ to $\mathcal{G} = T^2$ by grouping legs together as *virtual legs*. The position of a virtual leg is the "average[5]" of the positions of its individual legs. Figure 4 illustrates how legs are grouped for several typical gaits. In the future we would like to investigate using three or more virtual legs.

Rather than using a general h_{pl}, the domain vertices can be placed on the model gait and the space triangulated such that edges of the triangulation lie entirely along the model gait. In this case, the generated gait will be given explicitly by the range vertices corresponding to the domain vertices on the model gait. Figure 3 shows an example of this. Placing the vertices in this way proffers two benefits. First, the number of parameters in h_{pl} is held to a bare minimum to represent a particular gait. Second, since the gait is made apparent, this representation offers a handle for applying operator intuition in the selection of initial parameter sets. In fact in the virtual biped case, if the gait and the model gait are specified, the algorithm of [19] can be applied appropriately to generate a piecewise linear homeomorphism that maps the model gait into the gait.

IV. Experimental Results

RHex is a continually evolving platform. The first sets of experiments presented here were performed using RHex version 1.0, while the second set of experiments used RHex version 1.5. The fundamental difference is the addition in version 1.5 of a camera and gyroscope into the chassis and an increased mass of 1.2KG. None of the sensor data from the additional components was used during the second set experiments.

A. Optimizing h_{pl}

The experiments in this section optimize only over the parameters in h_{pl} and PD gains, k_p and k_d, while keeping the gait generation system fixed in the feedforward-centralized ($k_{ff} = 0, k_{cd} \rightarrow \infty$) paradigm, the control strategy initially implemented on RHex.

[4]From a more formal perspective, it is generally not known whether a given performance measure defines a smooth cost function, further limiting the appeal of strict gradient based descent.

[5]the notion of average is not well defined on S^1, for example consider 2 legs π radians apart. Nevertheless, since the legs in a virtual leg are supposed to be synchronized, they are never that far apart, and the average is take in the natural way.

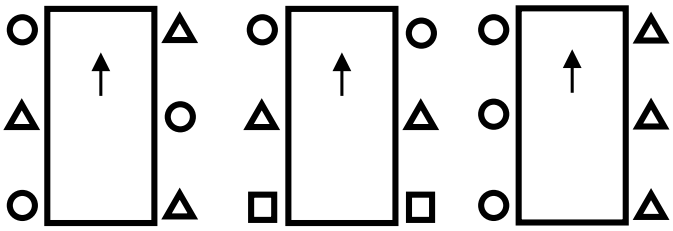

Fig. 4. A virtual leg consists of several physical legs which are driven synchronously. The graphic shows how legs are grouped into virtual legs for (*left*) tripod walking , (*middle*) hill- and stair-climbing, and (*right*) hill traversal.

1) Walking: The standard walking/running gait of RHex uses an alternating tripod gait, in which legs are grouped corresponding to the tripods into two virtual legs (see Fig 4 *left*), which are π radians out of phase. The gait generator was successively optimized with respect to two different performance criteria, endurance and speed. The performance criterion for endurance, f_e, is specific resistance

$$f_e = P_{av}/mgv_{av} \tag{5}$$

where P_{av} is the average power[6], m is the mass of the robot, g is acceleration of gravity, and v_{av} is average velocity. The performance criterion for speed, f_s is

$$f_s = P_{av}/mg{v_{av}}^3, \tag{6}$$

which is the specific resistance with an extra velocity term in the denominator. The inverse of velocity was tested and rejected as a performance criterion, because it led to gaits that were fast but extremely sensitive to perturbations from the environment. Trials were an eight meter run across linoleum, and experiments consisted of two trials. The experimenter hand-timed each trial, while current and voltage sensors on board the robot measured the power consumption.

We found that the performance of the optimization was highly dependent on the quality of the initial condition given to Nelder-Mead, and relied on the intuition of the experimenter to create good initial conditions. Though a number of anomalous descents "failed," in that poor initial conditions or anomalous measurements led Nelder-Mead to select a parameter set that did not perform capably and often led to spurious minima at "dead ends" of performance, the successful descents yielded significant performance increases. Figure 5 shows a typical descent. Table I shows that maximum speed of the speed gait increased threefold, up to 2.4m/s, and specific resistance was lowered more than threefold to 0.6. With both the speed and endurance gaits RHex achieves a true aerial phase and is thus running rather than walking. Using the optimized endurance gait, RHex can travel over 3.3km on a single set of batteries, up from 750m.

2) Hill Climbing: The standard walking gait has trouble climbing hills of significant slope, motivating the hand design of the hill-climbing gait in which RHex lies on its chassis, thrusts forwards with its legs, and then falls onto its chassis again. The virtual leg assignment is shown in Fig 4 *middle*. The largest impediment to climbing steep slopes with this gait was overheating the motors, especially the rear motors that support much of the weight while in motion. Optimization was applied to overcome this weakness, using a performance criterion measuring the power consumption over a safe threshold,

$$f_h = \int_0^T \max\{P - P_{th}, 0\}dt, \tag{7}$$

where T is the time of the trial, P is the instantaneous power, and P_{th} is the threshold power. A trial was a 2m climb up a slope of $40°$, and an experiment was two trials. Success of the optimization procedure was limited for the practical reason that a poor choice of parameters leads to overheated, or even burned out, motors, which must then be allowed to cool. Though hindered, the optimization produced substantially bet-

[6]Sometimes only mechanical power is included in the calculation, other times the total power (which includes power for the on-board computation and inefficiencies in the electronics) is used. In this work we shall use total power.

TABLE I

OPTIMIZATION OF WALKING GAIT WITH RESPECT TO f_e AND f_s

	f_e	f_s
Spec. Res. - Before	2.0	4.0
- After	0.6	0.7
Velocity - Before	0.5 m/s	0.8 m/s
- After	1.2 m/s	2.4 m/s
Descents: Success/Total	4 / 10	2 / 5

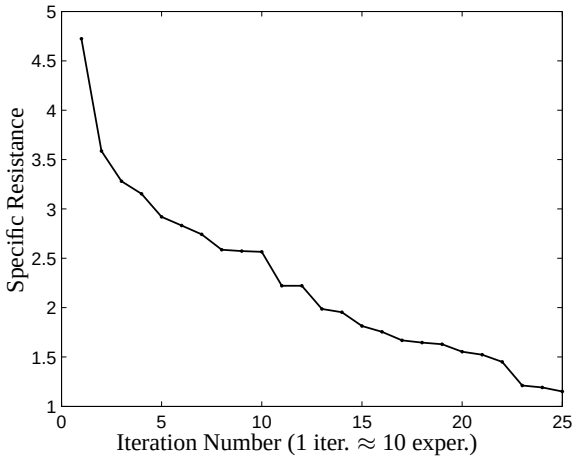

Fig. 5. Progress of Nelder-Mead optimizing walking gait for f_e

TABLE II

HILL CLIMBING GAIT OPTIMIZATION. MAXIMUM SLOPE CLIMBABLE AND MAXIMUM DISTANCE TILL MOTOR OVERHEATING ARE SHOWN FOR BEFORE AND AFTER OPTIMIZATION FOR VARIOUS SURFACES.

	Slope		Distance	
	Before	After	Before	After
carpet	$20°$	$45°$	2m	4m
sand	$15°$	$25°$	NA	3m
small rocks	$15°$	$25°$	NA	3m
large rocks	$15°$	$25°$	NA	2m

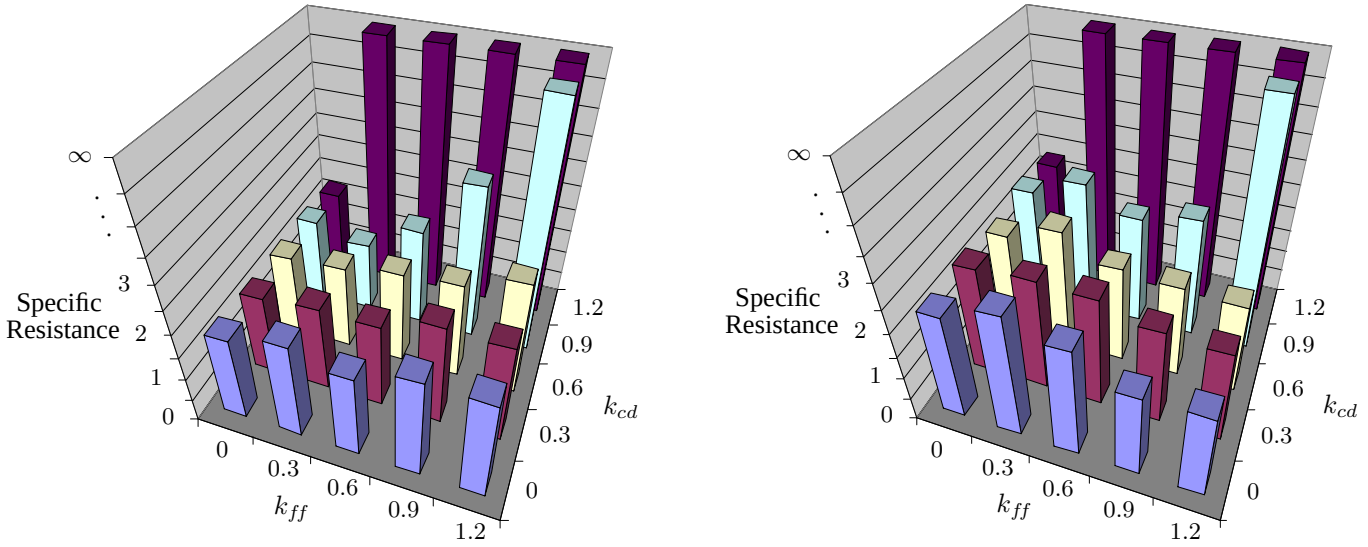

Fig. 6. Plot of the specific resistance as a function of k_{ff} and k_{cd} while running on (*left*) linoleum and (*right*) grass. The present data suggest that there may be different "sweet spots," locations at which specific resistance is lower, on the $k_{ff} - k_{cd}$ coordination plane for different terrains. Gaits which could not complete the trials successfully are marked as ∞.

ter performance. Table II shows that the maximum slope was increased by over 25 degrees on carpet and ten degrees on other surfaces. Also, the maximum distance climbed on carpet without overheating the motors increased by at least a factor of two. Data was not collected for the pre-tuned case on other surfaces. In the future, running the optimization on a hill with lower slope first should allow the optimization to be run subsequently on higher slopes while reducing the risk of overheating the motors.

3) Hill Traversal: Hill traversal is crossing a planar sloped surface along a trajectory of constant elevation. Using a standard walking gait on a slope as low as 10° for this task causes the robot to tumble onto its side down the slope. Hill traversal requires a gait that does not recirculate the legs on the higher side of the slope, as shown in Figure 3, where the second virtual leg (see Figure 4 *right*) does not recirculate. A traversal gait was hand designed, but the most difficult part of the task was maintaining constant elevation without zigzagging up or down the slope, motivating the performance criterion,

$$f_t = s, \tag{8}$$

where s is the distance the robot deviates from the constant elevation path the end of a traversal. Trials were 2m traversals, and experiments consisted of two trials. The optimization was performed on a slopes of 15° to 35° slope. The optimized gait can successfully traverse slopes of 35°, a 250% improvement over the 10° slope traversed by the walking gait.

B. Optimizing over the Coordination Plane

This set of experiments illustrate the effects of k_{ff} and k_{cd} on the gait generation system for a fixed h_{pl}, specifically h_{pl} generated from the endurance walking optimization. The walking optimization was performed on RHex version 1.0, while this experiment uses version 1.5, so h_{pl} is not optimized for the present platform. A grid of values of k_{ff} and k_{cd} were tested on two different types of terrain, linoleum and grass. The trials were 4 meter long, over which the specific resistance was measured. An experiment consisted of 2 trials. Figure 6 plots the results of these experiments. Those these results are preliminary, notice that the plots suggest different minima for the specific resistance. On grass, there seems to be a sweet spot with higher feedback (k_{ff} larger) and moderate centralization, while on linoleum the sweet spot has low feedback and high centralization. This provides some initial evidence forour hypothesis that rougher terrains, ie with more disturbances, will require higher feedback coupled with a decrease in centralization. More extensive experimentation over a variety of terrains is required to validate this hypothesis.

V. Conclusion

In this paper we have presented a gait generation system that provides the freedom to move in leg coordination space as well as providing the ability to shape gaits of a legged robots. We have shown how we use a function optimizer, namely the Nelder-Mead algorithm, to optimize parameters vis-a-vis a chosen cost function.

Using Rhex, we have experimentally validated the gait generation and optimization algorithms presented in this paper. Amongst other results we has seen an impressive factor of 3 improvement in velocity and specific resistance over what was already the record in the class of legged autonomous robots. We have also begun to show the value of moving in leg coordination space by empirically demonstrating that on different terrains the minimal specific resistance is found in different locations of the space.

Further work needs to be done to both better explore the coordination space and draw conclusions as to what the optimum parameter setting should be under different terrain conditions. We also plan to explore how the terrain, via feedback, effects the limit cycle of the system. Finally we wish to automate the tuning process, so that the robot will optimize it's own gait without any intervention from the experimenter.

ACKNOWLEDGMENTS

This work could not have been done without the design and development work on Rhex by Uluc Saranli, the help with coding from Haldun Komsuoglu, and the prior work of Eric Klavins. The work was supported in part by DARPA/ONR Grant N00014-98-0747.

REFERENCES

[1] U. Saranli, M. Buehler, and D.E. Koditschek, "Rhex: A simple and highly mobile hexapod robot," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.

[2] D.L. Jindrich and R.J. Full, "Dynamic stabilization of rapid hexapedal locomotion," *J Exp Biol*, vol. 205, pp. 2803–23, 2002.

[3] H. Wolf, "The locust tegula: significance for flight rhythm generation, wing movement control and aerodynamic force production," *J. Exp. Biol.*, pp. 229–253, 1993.

[4] H Wolf and A Buschges, "Plasticity of synaptic connections in sensory-motor pathways of the adult locust flight system," *J. Exp. Biol.*, pp. 1276–1284, 1997.

[5] A H Cohen, P J Holmes, and RH Rand, "The nature of the coupling between segmental oscillators of the lamprey spinal generator for locomotion: a mathematical model," *J Math Biol*, pp. 345–69, 1982.

[6] S Grillner, "Neurobiological bases of rhythmic motor acts in vertebrates," *Science*, pp. 143–49, 1985.

[7] L Guan, T Kiemel, and A H Cohen, "Impact of movement and movement-related feedback on the lamprey central pattern generator for locomotion," *J Exp Biol*, pp. 2361–70, 2001.

[8] J Schmitz et al., "A biologically inspired controller for hexapod walking: Simple solutions by exploiting physical properties," *Biol Bull*, pp. 195–200, 2001.

[9] H Cruse, "What mechanisms coordinate leg movement in walking arthropods," *Trends in Neuroscience*, pp. 15–21, 1990.

[10] Eric Klavins, Haldun Komsuoglu, Robert J. Full, and Daniel E. Koditschek, *Neurotechnology for Biomimetic Robots*, chapter The Role of Reflexes Versus Central Pattern Generators in Dynamical Legged Locomotion, MIT Press, 2002.

[11] Eric Klavins and Daniel E. Koditschek, "Phase regulation of decentralized cyclic robotic systems," *International Journal of Robotics and Automation*, vol. 21, no. 3, pp. 257–275, 2002.

[12] D. M. Wilson, "Insect walking," *Annu Rev Entomol*, , no. 11, pp. 103–22, 1966.

[13] M. Golubitsky et al., "Symmetry in locomotor central pattern generators and animal gaits," *Nature*, pp. 693–5, 1999.

[14] Eric Klavins, *Decentralized Phase Regulation of Cyclic Robotic Systems*, Ph.D. thesis, University of Michigan, 2002.

[15] Richard E. Groff, Pramod P. Khargonekar, and Daniel E. Koditschek, "A local convergence proof for the minvar algorithm for computing continuous piecewise linear approximations," *SIAM Journal on Numerical Analysis*, *in review*.

[16] Richard E. Groff, *Training Piecewise Linear Homeomorphisms for the Approximation of Maps with Known Invariants*, Ph.D. thesis, University of Michigan, 2002, *in preparation*.

[17] U. Saranli, "Simsect hybrid dynamical simulation environment," Tech. Rep. CSE-TR-436-00, The University of Michigan Department of Computer Science Technical Report, 2000.

[18] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.

[19] Aronov, Seidel, and Souvaine, "On compatible triangulations of simple polygons," *CGTA: Computational Geometry: Theory and Applications*, vol. 3, 1993.