

Gait Regulation and Feedback on a Robotic Climbing Hexapod

G. Clark Haynes and Alfred A. Rizzi
The Robotics Institute
Carnegie Mellon University
Email: {gch,arizzi}@cs.cmu.edu

Abstract—This paper proposes a novel method of applying feedback control for legged robots, by directly modifying parameters of a robot’s gait pattern. Gaits are a popular means of producing stable locomotion for legged robots, through the use of cyclic feedforward motion patterns, while requiring little to no sensory information. We are interested in incorporating feedback with these systems, and make use of salient parameters, found in gait patterns, to produce behaviors that span the space of possible gaits. These concepts are applied to a robotic hexapod, which, through the use of compliant microspines on its feet, is capable of climbing hard vertical textured surfaces, such as stucco. Experimental results are obtained comparing the use of a purely feedforward gait pattern to a behavior that actively modifies gait parameters while climbing, based upon sensory data.

I. INTRODUCTION

This paper develops a method for incorporating feedback into gait-based controllers, producing complex behaviors from relatively simple control systems. By applying control to a suitable parameterization of possible gaits, we produce robot behaviors that exhibit increased performance at a given task. In our realization of these concepts, a feedback controller modifies the gait of a robot to control task performance, while a second control system regulates the gait, to keep the overall robot behavior within those gaits suitable for the task.

Gaits are an abstracted view of legged locomotion, simple specifications of the cyclic feedforward motions each leg should take. As gaits are generally tuned for specific sets of operating conditions, they work well as long as a robot remains within those conditions.

Other methods have been proposed for producing complex behaviors on legged robots. One approach is to design a reactive system whose emergent behavior resembles a desired style of locomotion. While it is possible to encode some general properties of the desired system, the design process is non-intuitive and the emergent behavior can sometimes be a surprise to the designer. A second approach is to fully plan the individual motions for the legs and feet of a robot, letting the planning system determine the end-result behavior. Legged locomotion, however, necessarily involves complex and often nonholonomic constraints relating to surface contact making a planning approach very difficult to implement.

The use of gaits can overcome the limitations of these other methods by encoding desired behaviors directly into the feedforward patterns. Incorporating simple feedback control, using the gait-based strategies developed in this paper, allows a layering of feedback atop these feedforward gait patterns.

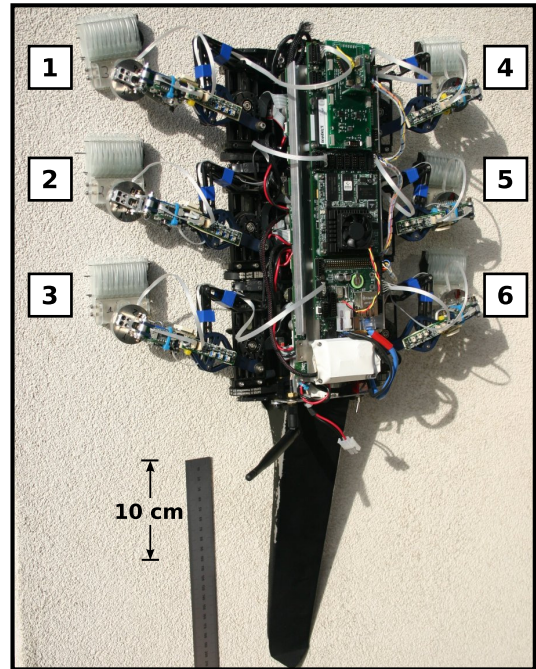


Fig. 1. The RiSE robotic platform. RiSE is a hexapodal robot able to climb on a variety of vertical surfaces as well as demonstrate horizontal mobility. RiSE uses compliant microspines on its feet to reliably attach to textured vertical surfaces, such as stucco, as seen here. Leg numbering conventions are noted

Furthermore, the relatively simple gait parameters we have developed are common to all robot gaits, and allow this work to be applied to a variety of robotic systems.

These concepts are applied to the RiSE robot (see Fig. 1) [1], where sensory feedback and gait regulation are performed on gaits while climbing textured walls. RiSE makes use of distributed feet, containing dozens of microspines each, to carry the load of the robot while it climbs [2]. Early intuition suggested that the robot performs better when forces are loaded evenly among the various feet. A foot carrying too much load is in danger of slipping, whereas a foot with too little load will not properly engage the surface. Using gait-based feedback strategies, we develop a controller to evenly balance force among the robot’s feet. Furthermore, we perform gait regulation to keep the robot in a pentapedal gait, minimizing the risk of falling by keeping five feet in contact with the surface at any given time.

A. Related Research

Ever since the first walking mechanisms were developed, roboticists have been trying to create robots that exhibit complex, animal-like motion. Biologists have attempted to reverse-engineer the neuronal bases of locomotion [3], [4] while their applied counterparts have created robots that use networks of simple reflexes and coordination schemes to locomote [4]–[7]. While complex behaviors can emerge from these systems, sometimes by chance, they are often tied to specific robotic implementations, and are not generalizable to other platforms.

The alternate approach has been deliberate and careful planning of every footfall a robot makes [8], [9]. These methods require accurate sensor information, accurate modeling of the constraints related to locomotion, and significant computational power to perform the planning, all of which are difficult to achieve on a mobile platform. While high-level behaviors have been exhibited, these methods have generally been largely unsuccessful on small, fast, and possibly dynamic mobile robots.

The idea of using gait patterns to encode locomotion, however, has become a popular alternative, due to its simplicity and the ability to involve robot dynamics. Robots store prespecified motion patterns, replaying each when necessary [10]–[12]. Much work has focused on understanding parameters for gaits, largely depending on artificial learning of effective gaits through automated tuning [7], [13]–[15]. These parameterizations have been robot specific, and little attention has been given to finding salient parameters common to multiple legged robots.

For feedback control with gaits, systems have been engineered that use leg coordination mechanisms [16] to locally make changes to a single feedforward gait [17]. This is differentiated from our work, where the feedback applies changes to the gait itself, resulting in a large set of possible gait patterns the robot may use.

II. GAITS AND BEHAVIORS

A. Defining and Parameterizing Gaits

A gait is a cyclic motion pattern that produces locomotion through a sequence of foot contacts with the ground. The legs provide support for the robot while the forces resulting from ground contact propel the body forward.

Gaits are motion patterns, simple functions that map from phase space, \mathbb{P} , to the configuration space of a robot, \mathbb{Q} . Phase is a concept similar to time, the major distinction being that time has constant velocity, whereas phase velocity is allowed to change. Gaits are *cyclic*, so the domain, \mathbb{P} , is topologically equivalent to the unit circle, \mathbb{S}^1 . The variable $\phi \in \mathbb{P}$ is used to denote the phase angle of the *master clock*, the clock that drives a gait.

If \mathbb{G} is the space of all possible gaits, then a gait $g \in \mathbb{G}$ is a periodic, continuous, and injective function from phase angle to desired robot configuration,

$$g : \mathbb{P} \rightarrow \mathbb{Q}. \quad (1)$$

On a robotic hexapod with actuators located only on the legs, the configuration of the robot is naturally thought of as

the Cartesian product of individual configuration spaces for each leg. If \mathbb{Q}_i is the configuration space of leg i then

$$\mathbb{Q} = \mathbb{Q}_1 \times \mathbb{Q}_2 \times \mathbb{Q}_3 \times \mathbb{Q}_4 \times \mathbb{Q}_5 \times \mathbb{Q}_6. \quad (2)$$

Likewise, it is useful to think of a gait as separate individual functions, $g_{1-6}(\phi)$, where $g_i(\phi)$ is the function dictating movement on leg i .

Using this decomposition, useful parameters emerge from g . An important distinction is made between a leg that is in contact with the ground, termed *in stance*, and a leg that is recirculating in the air, *in flight*. The phase angle at which a given leg begins stance is called the *stance phase offset*, $\rho_i \in \mathbb{P}$.

For a given ρ_i , $g_i(\phi)$ is *normalized*, producing a related function for which $\phi = 0$ corresponds to a leg beginning stance,

$$g_i(\phi) = g_{n,i}(\phi - \rho_i). \quad (3)$$

Note that $g_i(\rho_i) = g_{n,i}(0)$. This notation is useful because, in most cases, the $g_{n,i}(\phi)$ functions are identical for multiple legs, and the stance phase offsets are readily accessible parameters, decoupled from the rest of the gait.

Stance phase offsets are temporal parameters, and suggest a structure to the timing of a gait. Multiple legs with the same stance phase offset make contact with the ground at the same time, whereas differing values dictate an order in which legs make contact.

It should be noted that gaits are actually defined by the *phase differences* between legs, so representing a gait with six phase offsets is an over-parameterization. Two gaits with phase offsets a fixed value apart are equivalent.

B. Gaits as a Basis for Behaviors

Unlike gaits, which are constrained to producing only periodic motion, behaviors can include non-cyclic motion and can be arbitrarily complex. While there have been a variety of approaches to encoding complex behaviors on legged robots, the use of gaits as a basis allows the layering of relatively simple feedback systems on top of already-existing feedforward gaits.

As a feedforward gait encodes a desired style of locomotion, a behavior that uses that gait as its basis builds upon that same style of locomotion, provided the behavior stays relatively “close” in gait space, \mathbb{G} .

To write a gait-based behavior, we design controllers that output a desired change of gait. Mathematically, this is a first-order continuous system that maps from some set of control inputs, \mathbb{U} , to the tangent space of gaits,

$$b : \mathbb{U} \rightarrow T\mathbb{G}. \quad (4)$$

Performing control on a gait, by following what are just *gait velocities*, a behavior is just a curve, in gait space, directed by the various time-dependent control inputs.

In this paper, we use stance phase offsets, $\rho_i \in \mathbb{P}$, as a relatively simple method of parameterizing gaits. A set of six

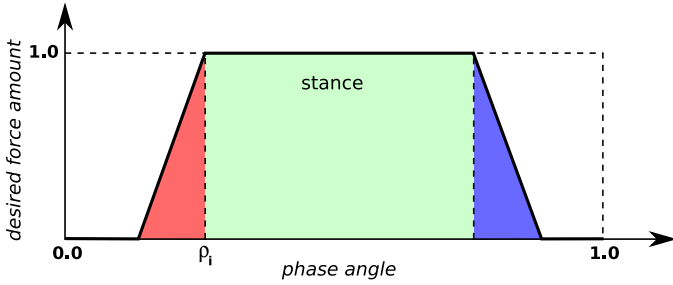


Fig. 2. The desired force profile, $d_i(\phi)$. During stance, the area lightly shaded, a foot should carry 100% of its desired force, whereas during flight, it shouldn't carry anything. Attachment and detachment involve adding and removing load, over some finite period.

phases offsets, therefore, is a subset of the full gait space, $\mathbb{P}^6 \subset \mathbb{G}$.

The gait-based behavioral controller developed in this paper is given by

$$\dot{\rho}_i(t) = b_{f,i}(t) + b_{c,i}(t), \quad (5)$$

and produces a set of phase offset velocities. One component of the controller, $b_{f,i}$, is a force-balancing controller, developed in III, while the other portion, $b_{c,i}$, performs leg coordination and gait regulation, IV. The robot simply plays the motion patterns of the current gait, while this controller continuously modifies the parameters of the gait, as necessary.

III. FORCE-BALANCING CONTROLLER

Legged robots locomote via the generation of ground reaction forces. By comparing a foot's actual measured force against a nominal force, we can understand how effectively a foot is being used in a gait. The control system developed in this section builds upon this idea and tries to modify a gait to better balance forces among the various feet in stance, to prevent any one leg from carrying too much or too little load.

A. Desired Forces

While locomoting, each foot should produce some nominal force during stance. To balance the forces among our legs, we must first understand the characteristics of this desired force.

In steady state, each gait has a characteristic set of ground reaction forces. In the case of a climbing robot, we are most interested in foot traction, the force felt in the fore-aft direction, the robot's direction of travel, as this gives us an idea of the weight that each foot carries. Furthermore, this is the same direction that legs in stance move as they propel a robot forward.

While a foot has a desired traction force during stance, we must also take into account a period of loading before stance, *attachment*, and unloading after, *detachment*. There is, therefore, some profile that determines the desired percentage of force a leg should have when in contact with the ground. Figure 2 shows the force profile used in this paper, $d_i(\phi)$. Each foot carries no load during flight, and begins to carry full load starting at its stance phase offset angle, ρ_i .

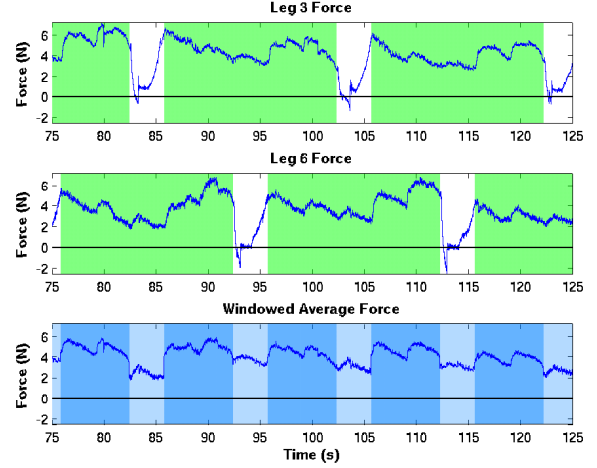


Fig. 3. An example of computing an average traction force for two feet. The top two plots show measured force from two separate legs, shaded regions indicating stance. The bottom plot is their average, during stance. The amount of shading indicates whether 1 or 2 measurements were used to compute the average.

B. Computing an Average Force

In order to determine if a foot is carrying too much or too little force, however, we must understand how to calculate an average force, taking into account the fact that feet load and unload at different times in a gait pattern.

When calculating an average force, we are only interested in forces measured while a foot is in contact with the ground, ignoring measurements during recirculation. If $f_{m,i}$ is the measured traction force for a given foot, (6) simply removes spurious data during flight.

$$f_i = \begin{cases} 0 & \text{if leg } i \text{ is in flight} \\ f_{m,i} & \text{if leg } i \text{ is not in flight} \end{cases} \quad (6)$$

Using this equation to sum the forces of all the feet gives us the total force, carried by the robot, of feet in contact with the ground. Figuring out how many feet to divide by, to compute an average, is slightly more difficult, however.

A foot in stance should count as a full unit – if a robot has three feet in stance, the total force should be divided by three. A foot that is in attachment or detachment, however, is only meant to be carrying a partial amount of force, therefore it counts as only a partial unit. A robot with three feet in stance, and one foot that is halfway done attaching should be averaged amongst 3.5 feet, not four. The function $d_i(\phi)$ provides us with the perfect method of counting how many feet to average over. Therefore the equation to compute the average force a is given by

$$a = \frac{\sum_{i=1}^6 f_i}{\sum_{i=1}^6 d_i(\phi)}. \quad (7)$$

Figure 3 shows a simplified version of (7), for only two feet, visually showing an average that ignores a foot in flight.

We now have a method to compute the average force of a foot in full contact with the ground. To compute the desired force, in Newtons, we simply multiply (7) by $d_i(\phi)$ to yield

$$f_{d,i} = d_i(\phi)a. \quad (8)$$

C. Gait-Level Force Control

With both a measurement of a foot's actual traction force, and its desired force, as determined by the average force calculation, we construct a gait-based behavioral controller that computes changes to the current gait through phase offset velocities.

The phase offsets of a feedforward gait are constant, thus phase offset velocities are always zero. For a feedback behavior that actively modifies phase offsets, however, the phase offset velocities can and will be non-zero. A positive phase offset velocity corresponds to delaying a given leg, relative to the other legs in a gait, whereas a negative phase offset velocity speeds up the leg.

Since traction forces, when climbing, are generally aligned with the direction in which a foot moves, speeding up and slowing down an individual foot will generally increase or decrease the traction force of that foot, respectively. Thus, modifying the phase offsets of a gait should be sufficient to actively control traction forces while climbing.

A simple control system would compare the actual measured traction to the desired traction force, and speed up or slow down a leg as necessary. A simple proportional controller, with gain k_p , can be constructed,

$$\tilde{f}_i = f_i - f_{d,i} \quad (9)$$

$$b_{f,i}(i) = k_p \tilde{f}_i. \quad (10)$$

Due to the relative constraints and limits of our current sensing and actuation mechanisms, we chose to implement a discretized version of this controller, using simple thresholds to determine when to speed or slow individual legs. This is done by introducing an allowable error deadband, within which a leg's force is considered satisfactory,

$$b_{f,i}(i) = \begin{cases} 0 & \text{if } f_i \leq t_{upper} f_{d,i} \text{ and } f_i \geq t_{lower} f_{d,i} \\ r & \text{if } f_i > t_{upper} f_{d,i} \\ -r & \text{if } f_i < t_{lower} f_{d,i}. \end{cases} \quad (11)$$

$t_{upper} > 1.0$ and $t_{lower} < 1.0$ are chosen values representing the deadband region. The rate used to speed or slow legs is specified by r .

Figure 4 visually shows this controller in action, showing the response for a single leg in stance. When the traction force exceeds its desired value, the controller slows the leg down until the force is reduced. Later in stance, when the traction drops below the deadband region, the controller speeds the leg up, and so on.

IV. GAIT REGULATION

The control system developed in III-C makes changes to a gait, by modifying the stance phase offsets, to better match

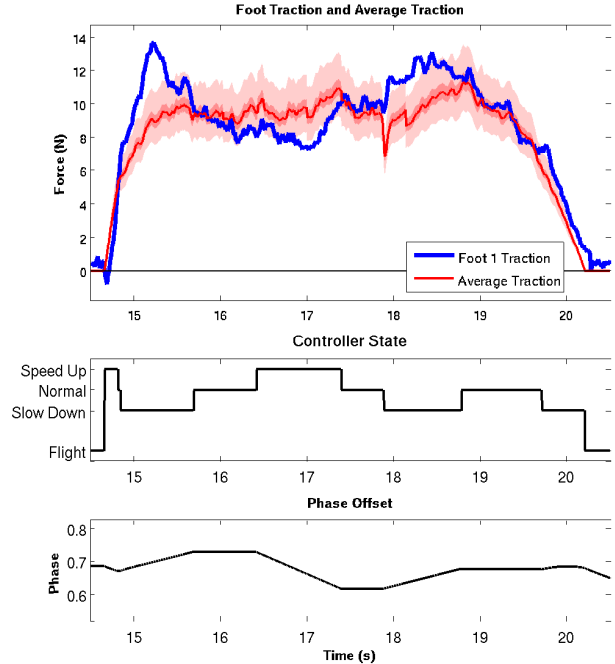


Fig. 4. Results with the gait-based force controller from (11). The top plot shows the actual traction force for a given foot, as well as the desired force, computed from the average traction force, with a shaded deadband region. Note how the force follows a profile similar to that in Figure 2. The middle plot shows the action taken by the controller, and the bottom plot shows the resulting change to the leg's stance phase offset, ρ_i .

forces among legs. Such a change, however, can result in unstable robot configurations, for instance one with too many legs recirculating at the same time.

In this section, we introduce the concept of *gait regulation*, building coordinating mechanisms between legs to keep a robot within a set of gaits that are suitable for a given task.

A. Understanding Pentapedal Gaits

Pentapedal gaits, as are used on a robot to climb many vertical surfaces, are interesting for many reasons. Foremost, they are gait patterns that are designed to keep all six legs of a hexapod out-of-phase with one another, always having five legs on the ground and recirculating only a single leg at a time. This translates to the stance phase offsets being perfectly spaced out, by sixths, on the circle, S^1 .

Combinatorics are involved in the choice of a pentapod's leg order. For a given starting leg, there are $(6 - 1)! = 120$ different ways to order the remaining legs, resulting in 120 distinct pentapedal gaits, with no appreciable difference in performance between them.

When a given leg's phase offset is modified, such as by the control system in (11), it is possible for the leg to leapfrog another, switching their order in the current gait. The nearest pentapod, in gait space, changes to another one of the 120 possible choices.

B. Leg Coordination

Using simple control laws, we construct a coordinating mechanism whose stable regions correspond to each of the 120 possible pentapedal gaits.

While previous work in leg coordination has dealt primarily with sets of decentralized clocks [16], [17], the use of a single master clock with six phase offsets is equivalent. Rather than coordinating multiple clocks, we coordinate the phase offsets that are applied to a single clock.

Keeping legs perfectly out-of-phase with one another suggests the use of a fully-connected coordination network. Applying a virtual force using a network of cosines is suitable to handle this. When comparing phase offsets, as in (12), we use modulo arithmetic to map the difference into a phase difference of range $[0, 1)$. We also take care to then map the range $[0, 1)$ to radians, $[0, 2\pi)$;

$$e_{i,j} = \rho_i(t) - \rho_j(t) \pmod{1.0} \quad (12)$$

$$c_i(j) = \begin{cases} 0 & \text{if } i = j \\ \cos(\frac{1}{2}(2\pi e_{i,j})) & \text{if } i \neq j \end{cases} \quad (13)$$

$$b_{c,i}(t) = \sum_{j=1}^N c_i(j). \quad (14)$$

$b_{c,i}$ can then be used to coordinate the phase offsets of a gait, as in (5). As before, a positive phase offset velocity will delay a given leg, while a negative value speeds it up. The coordination network encoded by (14) is capable of driving an arbitrary number of items out-of-phase with one another, and is zero-valued at stable configurations, including all of the 120 possible orderings. Fig. 5 shows an example of coordination among six simulated legs.

C. Pairing Gait Regulation with Feedback

We must take care to prevent the leg coordination controller from interfering with the force-balancing control system from III, thus we develop a strategy to coordinate legs only during flight, regulating gaits on a stride to stride basis.

It is undesirable to perform leg coordination during stance, as changing a leg's phase offset has the possible effect of canceling out the force-balancing controller. Using a simple weight to reflect whether or not a leg is in flight, we construct a modified gait regulation controller.

$$b_{c,i}(t) = (1 - d_i(\phi)) \sum_{j=1}^N c_i(j) \quad (15)$$

We make use of the function previously shown in Fig. 2, which, when subtracted from 1 is maximal during flight and zero during stance. Fig. 6 shows the result of this in simulation.

V. EXPERIMENTAL RESULTS

We apply gait-based behavioral control to a climbing hexapod, the RiSE robotic platform. Comparing the overall foot traction forces of a simple feedforward gait to those of the feedback behavior described in this paper, we see an improvement in the distribution of load among the feet as well

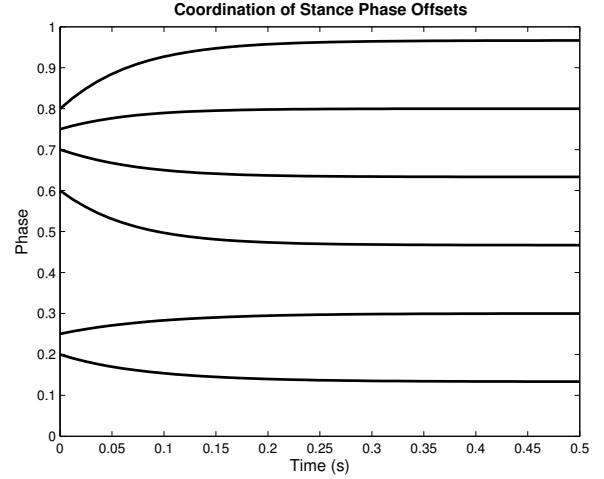


Fig. 5. A network of cosines repel all phase offsets away from each other, quickly stabilizing in a pentapedal gait, no matter the starting order of the phase offsets.

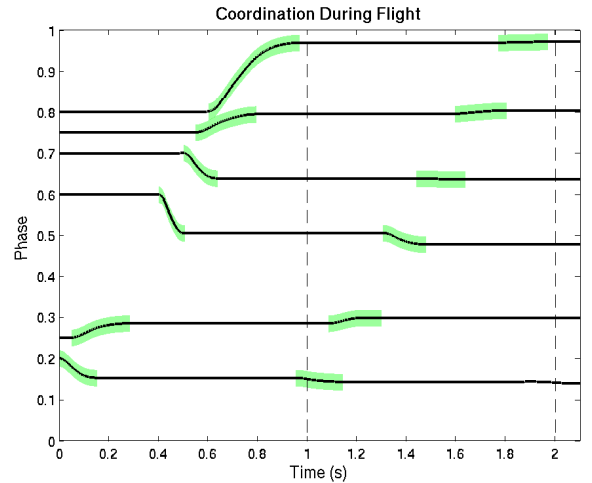


Fig. 6. When allowing coordination to occur only during flight, the shaded regions, it takes multiple strides to have the robot return to a pentapedal gait. Note how, when regulated, legs recirculate one at a time.

as increased robustness to failure when using the feedback behavior.

A. RiSE Robot

The RiSE robot, Fig. 1, is a bioinspired hexapod designed to climb on a variety of vertical surfaces as well as to demonstrate horizontal mobility.

The robot has a total of twelve degrees of freedom, two joints on each of the six legs. The *wing* joint lifts and lowers a foot onto the climbing surface, while the *crank* joint moves a foot up and down along the surface and is primarily responsible for lifting the robot as it climbs.

RiSE has a body length, not including the tail, of 41 cm and a total weight of 2.8 kg. RiSE is computationally and power autonomous, runs on a 266 MHz processor, and is capable of operating without a tether for extended periods of time. Motor

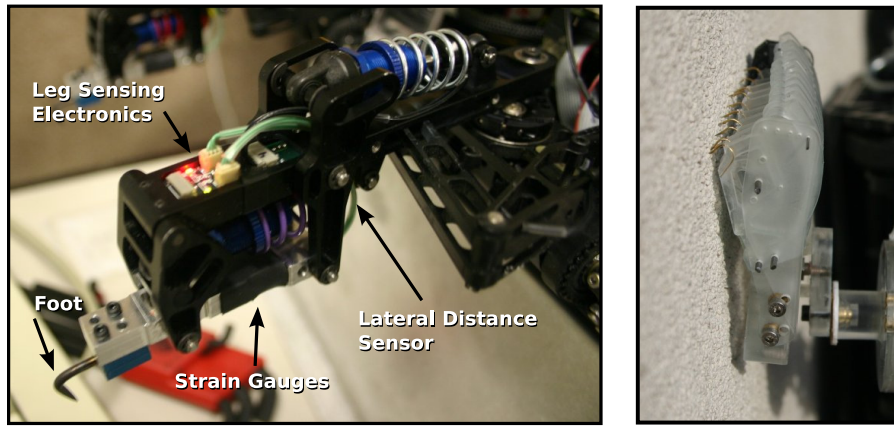


Fig. 7. *Left*: Annotated picture of a RiSE lower leg. Various types of feet can be installed at the distal end of the leg. Two strain gauges measure the normal and fore-aft forces on a foot. With a compliant passive degree-of-freedom in the lateral direction, a distance sensor, paired with a spring constant, gives us lateral force. *Right*: A RiSE foot with individually-compliant microspines on a stucco surface.

level control is performed at a frequency of 300 Hz, while a human operator issues task-level commands to the robot via a laptop computer, connected over an 802.11b wireless connection.

To climb on flat textured surfaces, the robot makes use of feet containing arrays of individually-compliant microspine toes, Fig. 7, each toe is capable of carrying a small fraction of the total weight [2]. When many toes make stable contact with asperities on a surface, force in the fore-aft direction (traction force) is distributed throughout a foot. Furthermore, the robot’s total weight requires that this force be distributed amongst a majority of the six legs while climbing.

Each of RiSE’s legs is instrumented with a set of sensors to accurately measure forces between the robot and the ground, shown in Fig. 7. For the purpose of our experiments, we studied the traction force of each foot, measured by the strain gauge aligned in the fore-aft direction.

B. Experimental Methodology

A climbing wall was built for simple robot tests on flat surfaces, allowing easy climbing angle modifications, ranging from horizontal to vertical. Using the microspine feet, RiSE is capable of climbing on a variety of stucco-like surfaces. For in-lab tests we found that asphalt roof shingles offered a workable substitute without having to recreate a full stucco wall.

To minimize the variation between experiments, feedforward and feedback controllers were run alternately for a total of six runs, three runs apiece. Each experiment lasted approximately 1.5 minutes, and involved a climb of 2 body lengths on the test wall surface. The lack of robustness in the feedforward gait prevented the robot from climbing a vertical surface, thus the test wall was set at a climbing angle of 75 degrees. It should be noted that the feedback behavior is capable of climbing vertical surfaces, owing to its increased robustness, but was also tested at 75 degrees, for uniformity.

The spatial trajectory of the feedforward gait was tuned empirically prior to our work with feedback behaviors. During attachment, each foot is presented flat to the surface, and pulled downward to allow toes to catch on surface asperities.

The foot continues through stance, slowly pulling the robot upwards on the wall, until detachment, when the foot reverses direction to unload force before recirculating in flight.

It should be noted that the feedforward gait does perform local proportional-derivative (PD) control at each of the robot’s joints, thus it is really only task open-loop, performing some low-level feedback control. The feedback behavior, however, can be considered task closed-loop, since it uses force sensors to perform control depending upon success at a task, in this case balancing forces while climbing.

The feedforward gait used was just one of the 120 possible pentapedal gaits, with each leg having a constant phase offset and following the same periodic motions as it climbed. The feedback behavior used the same spatial trajectory as the feedforward gait, but incorporated modification of stance phase offsets to vary the timing between legs, by use of the controller in (5). This included force-balancing during stance and gait regulation during flight.

C. Quantitative Analysis of Traction Forces

To directly compare the results of the feedforward gait with the feedback behavior, we analyzed data collected from the robot during experiments. This data included the traction force of each foot, as used for input to the feedback behavior, as well as specific gait parameters, such as leg phase offsets (constant in the feedforward case) and the phase angle of the master clock.

Since the robot inherently performs a repetitive task, moving each foot through the same motions time and time again, a regular pattern emerges in our observed traction force. We call this pattern a *force profile*.

Just as a gait is a function from phase angle to robot configuration, a force profile is a similar function from phase angle to ground reaction forces, a foot’s traction force in our case. The phase angle of a leg, at any given time, is computed by subtracting a leg’s phase offset from the phase angle of the master clock, $\psi_i = \phi - \rho_i$.

As each foot repeats its motions over and over, we obtain additional measurements of force for each leg phase angle. By aggregating these forces by stride, we compute meaningful

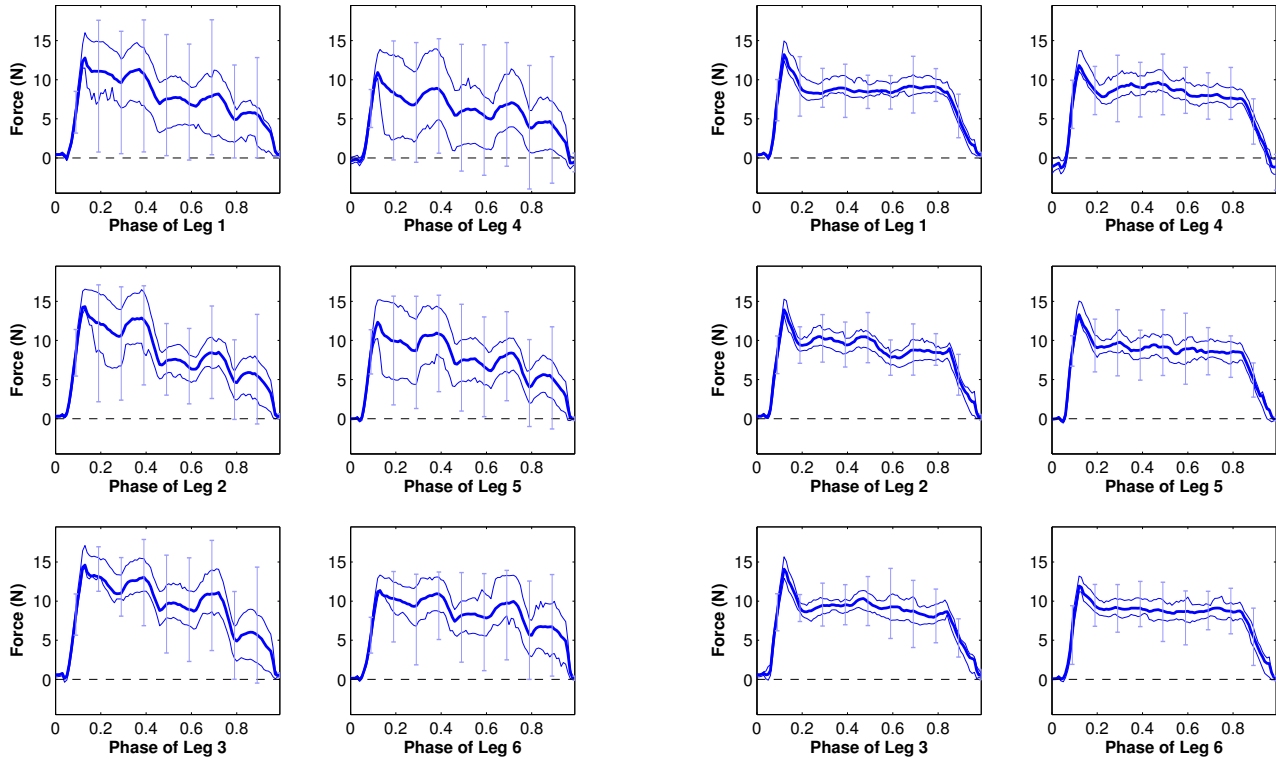


Fig. 8. Traction force profiles of the feedforward gait, for all six feet of the robot. The thick center line corresponds to the computed average, while the surrounding lines indicate quartile values. Upper and lower bounds, excluding statistical outliers, are marked every 0.1 phase. Stance occurs during the portion of phase when the traction force is non-zero. Flight corresponds to the left and righthand sides of each plot, where force is approximately zero.

statistics based upon phase angle. This includes average values, quartiles, upper and lower bounds, as well as statistical outliers. Visual representations of force profiles, comparing the feedforward gait to the feedback behavior, are shown in Figs. 8 and 9.

Data from all three feedforward gait experiments was used to produce the force profiles in Fig. 8. The average traction force seen in these profiles seems to jump up and down throughout stance. This is attributed to other feet attaching and detaching during a stride. Furthermore, a foot's average traction seems to drop over the course of a stride, meaning that a foot that has recently attached is more likely to carry load than a foot near the end of stance. Lastly, the inner-quartile distance of these force profiles, particularly for the front feet (legs 1 and 4), is quite large, indicating a high variance in how much force a foot may end up carrying on each stride.

Similarly, data from all three feedback experiments was analyzed to produce the force profiles in Fig. 9. We see several notable differences from the feedforward force profiles in Fig. 8. Foremost, the traction force during stance stays relatively constant and flat, compared to the irregularities and tapering off of the feedforward gait. Also note how similar in

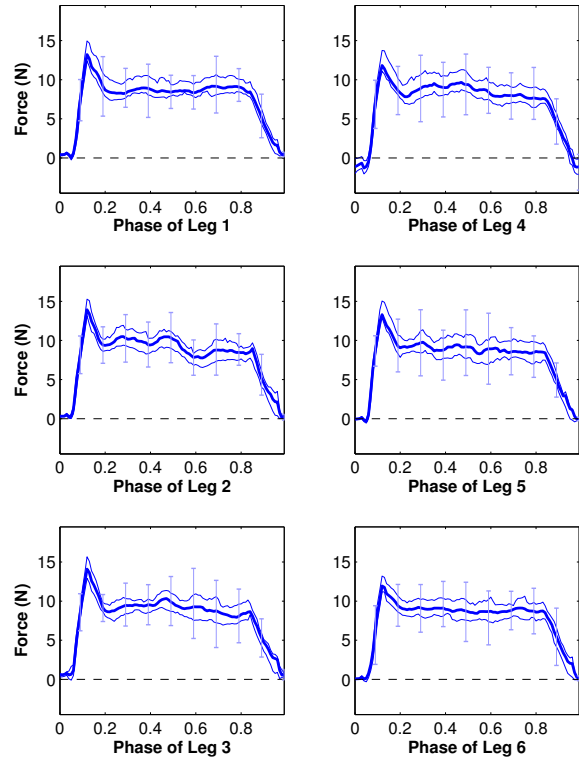


Fig. 9. Traction force profiles of the feedback behavior. As in Fig. 8, the average value is surrounded by quartile values, with upper and lower bounds marked occasionally throughout.

shape each force profile is compared to the desired percent of force in Fig. 2. Lastly, we can see that the quartile values are closer to the average value, and that the upper and lower bounds are much tighter.

This marked increase in repeatability of the force profiles enables the feedback behavior to climb more robustly than the feedforward behavior that utilizes the same gait.

D. Behavioral Improvements

In addition to quantitative improvements in measured foot forces, our feedback behavior resulted in increased behavioral capabilities, allowing us to exhibit the RiSE robot climbing robustly on a variety of challenging surfaces.

Subjectively, the robot seems to grasp a surface better when applying feedback, thanks to the active monitoring of traction forces by the force-balancing controller. Feet lose grip on a climbing surface less often, and this increased robustness allowed the robot to complete an untethered climb of a three story stucco building, while undergoing testing at the Southwest Research Institute in April 2006 (Fig. 10). This is an example of a task where a feedforward gait's lack of robustness makes the traditional open-loop strategy unsuitable.

Furthermore, this same behavior, without modification, was successfully tested on a variety of stucco surfaces, suggesting a robustness to variation in surface properties, and was also quite

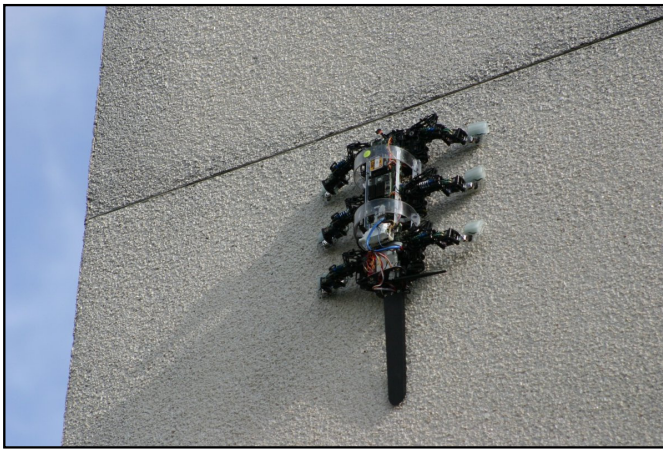


Fig. 10. The RiSE robotic platform, while completing an untethered climb of a three story building, making use of the feedback behavior developed in this paper.

successful at climbing brick walls. On the brick, a foot would only occasionally gain traction when microspines contacted the face of a brick, and would more often prefer to attach at the slightly ungrouted edge of a brick. Our controller naturally encoded this behavior, speeding up a foot until it carried enough force, such as when it found purchase on the edge of a brick.

Overall, these various experiments have demonstrated the utility of performing gait-based control, showing significant improvements in performance and capabilities over feedforward gaits.

VI. CONCLUSION

Through an understanding of the implicit properties of legged robot gaits, we have presented a novel method for applying feedback control directly to the space of gaits, while building upon existing feedforward gaits. Task-level feedback is performed, by comparing a foot's ground reaction forces against a desired force profile, and applying control through gait parameters. Gait regulation is then used to coordinate a robot's legs, to keep the robot within a suitable range of possible gaits.

We have demonstrated the utility of these two separate control systems in one unified behavior on a robotic hexapod. Preliminary experiments suggest that the feedback behavior exhibits large improvements over a simple feedforward gait when performing identical tasks, such as climbing a challenging surface.

Our future work, with regard to gait-based control for legged robots, will head in several related directions. By more closely aligning this work with that in [16], we hope to extend our methods of gait regulation to incorporate other types of gaits, such as tetrapod and tripod gaits, in addition to the current capability of coordinating pentapedal gaits. Furthermore, we are exploring further methods of parameterizing gaits to perform

feedback on additional axes of ground reaction force, the normal and lateral forces. Lastly, owing to the general concepts of gaits on legged robots, we hope to apply these methods of control to additional types of gaits, not just the stucco-climbing gait presented here, as well as to robots of different morphologies, including quadrupeds.

ACKNOWLEDGMENTS

This work is supported in part by the DSO Biodynamics Program, within the Defense Advanced Research Projects Agency, under contract DARPA/SPAWAR N66001-03-C-8045.

REFERENCES

- [1] K. Autumn, M. Buehler, M. Cutkosky, R. Fearing, R. J. Full, D. Goldman, R. Groff, W. Provancher, A. A. Rizzi, U. Saranli, A. Saunders, and D. E. Koditschek, "Robotics in scansorial environments," G. R. Gerhart, C. M. Shoemaker, and D. W. Gage, Eds., vol. 5804, no. 1. SPIE, 2005, pp. 291–302. [Online]. Available: <http://link.aip.org/link/?PSI/5804/291/1>
- [2] A. T. Asbeck, S. Kim, M. R. Cutkosky, W. R. Provancher, and M. Lanzetta, "Scaling hard vertical surfaces with compliant microspine arrays," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.
- [3] D. M. Wilson, "Insect walking," *Annual Review of Entomology*, vol. 11, pp. 103 – 122, 1966.
- [4] H. Cruse, "What mechanisms coordinate leg movement in walking arthropods?" *Trends in Neurosciences*, vol. 13, pp. 15 – 21, 1990.
- [5] R. A. Brooks, "A Robot That Walks; Emergent Behaviors from a Carefully Evolved Network," MIT AI Lab, Memo 1091, February 1989.
- [6] K. S. Espenschied, R. D. Quinn, H. J. Chiel, and R. D. Beer, "Leg coordination mechanisms in stick insect applied to hexapod robot locomotion," *Adaptive Behavior*, vol. 1, no. 4, pp. 455 – 468, 1993.
- [7] M. Lewis, A. Fagg, and G. Bekey, "Genetic algorithms for gait synthesis in a hexapod robot," 1993. [Online]. Available: citeseer.ist.psu.edu/lewis94genetic.html
- [8] D. Wettergreen, H. Thomas, and C. Thorpe, "Planning strategies for the ambler walking robot," in *IEEE Int. Conf. on Systems Engineering*, August 1990, pp. 198 – 203.
- [9] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. K. Hodgins, and T. Kanade, "Footstep planning for the honda asimo humanoid," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, April 2005.
- [10] R. Altendorfer, N. Moore, H. Komsuoglu, M. Buehler, H. B. Brown, D. McMordie, U. Saranli, R. Full, and D. Koditschek, "Rhex: A biologically inspired hexapod runner," *Autonomous Robots*, vol. 11, p. 207, 2001.
- [11] S. Schaal and C. G. Atkeson, "Open Loop Stable Strategies for Robot Juggling," in *International Conference on Robotics and Automation*, vol. 3, GA, Atlanta, 1993, pp. 913–918.
- [12] J. G. Cham, S. A. Bailey, J. E. Clark, R. J. Full, and M. R. Cutkosky, "Fast and robust: Hexapedal robots via shape deposition manufacturing," *International Journal of Robotics Research*, vol. 21, no. 10, 2002.
- [13] J. D. Weingarten, G. A. D. Lopes, M. Buehler, R. E. Groff, and D. E. Koditschek, "Automated gait adaptation for legged robots," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 3, 2004, pp. 2153 – 2158.
- [14] S. Chernova and M. Veloso, "An evolutionary approach to gait learning for four-legged robots," in *In Proceedings of IROS'04*, September 2004.
- [15] V. Zykov, J. Bongard, and H. Lipson, "Evolving dynamic gaits on a physical robot," in *Late Breaking Papers at the 2004 Genetic and Evolutionary Computation Conference*, M. Keijzer, Ed., Seattle, Washington, USA, 26 July 2004. [Online]. Available: <http://www.cs.bham.ac.uk/wbl/biblio/gecco2004/LBP065.pdf>
- [16] E. Klavins and D. Koditschek, "Phase regulation of decentralized cyclic robotic systems," *The International Journal of Robotics Research*, vol. 21, no. 3, 2002.
- [17] J. D. Weingarten, R. E. Groff, and D. E. Koditschek, "A framework for the coordination of legged robot gaits," in *IEEE Int. Conf. on Robotics, Automation and Mechatronics (RAM)*, vol. 2, 2004, pp. 679 – 686.