USING RECONFIGURABLE MODULAR ROBOTS FOR RAPID DEVELOPMENT OF DYNAMIC LOCOMOTION EXPERIMENTS

Jimmy Sastra

A DISSERTATION

in

Mechanical Engineering and Applied Mechanics

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2012

Mark Yim
Professor Mechanical Engineering and Applied Mechanics
Supervisor of Dissertation

Jennifer Lukes
Associate Professor Mechanical Engineering and Applied Mechanics
Graduate Group Chairperson

Dissertation Committee

Daniel E. Koditschek, Professor of Electrical and Systems Engineering

Vijay Kumar, Professor of Mechanical Engineering and Applied Mechanics

USING RECONFIGURABLE MODULAR ROBOTS FOR RAPID DEVELOPMENT OF DYNAMIC LOCOMOTION EXPERIMENTS

To the crazy ones.

Acknowledgements

That only my name appears below the title of this dissertation is grossly misleading. This work would not have been possible without the hardworking members of the ModLab team. In particular I'd like to thank:

Sachin Chitta	Matthew Piccoli	Babak Shirmohammadi
Daniel Gomez-Ibanez	Michael Park	Kevin Galloway
Chris Thorne	Thomas Mather	Paul White
Jaimeen Kapadia	Ian O'Hara	Ian Stephens
Jay Davey	Jedtsada Laucharoen	Willy Bernal-Heradia
Uriah Baalke	Steven Kum	Mohit Bhoite
Daniel Meana	Stella Latscha	

It has been my pleasure to work with such an excellent group. I've learned so much from all of you. To future students of ModLab: I have made every mistake possible. I hope you will do the same. This is the time and place to do so.

Next, I would like to thank my mentor, Mark Yim, for giving me the confidence to follow my own voice in a rigorously logical world, where people may look at you funny, when you're wielding a hot-glue gun with foamcore, and tape in the name of creativity. Thank you for listening to my wacky ideas and letting me indulge in all my interests. I believe nothing has been left undabbled. The breadth of my education can only be attributed to your teachings. To give a sports analogy: Total Soccer is a strategy in which any player can take on the role of any other player on the field. A midfielder can instantly take over as attacker, fluidly, at any point in

the game. I believe your multi-disciplinary lab has trained me and everyone in it as an army of *Total Engineers*.

I want also to thank my other mentor, Shai Revzen, for being the perfect complement to Mark. Your hard questions have always led to the greatest insights. At your hands, my analytical muscles have gotten a tremendous workout. You have profoundly changed the way I think. The depth of my education is due to you. I have high hopes for your new lab. I expect great things.

I would also like to make a special note of Maryeileen Griffith and Olivia Brubaker who have gone above and beyond in helping me navigate the administrative shoots and ladders.

I want to thank Suresh Swaminathan for being the continuous solid friend I can always depend on. From the beginnings as poor students in a little dormitory room to, well, I'm still a poor student. Thank you for always being there for me, and letting me eat your food. I want to thank Michael Park for teaching me the zen of moped maintenance and always offering a unique perspective on life during the inevitable existential crises one endures as a Ph.D. student. I want to thank all the rest of my friends, who have been foolish enough to follow me on my adventures at XPLR-club and rooftop movie nights. I want to thank my friends Deepak Kollali, Karen Yung, Amelie Boquoi, Tina Chen, Jan Baranski, Andrea Rosen, Gabriel Lopes, Emilie Yane, Olga Shebanova, Falon Shrokman, Roanne Mejilla, Paul Vernaza, and many more than I can fit on this page. You have added colour to my life.

Most importantly, I must thank my Mom. You have raised me to be the person I have become. I owe it all to you. I will always be your son. I want to thank my little brother for being more mature and responsible than I am, and therefore being the one that takes care of me. Last, I want to thank my Dad for helping me continuously refine my Hedgehog concept. Who knows where it may lead, since it is evident I have an affinity for the road less traveled. You have shown me the world.

ABSTRACT

USING RECONFIGURABLE MODULAR ROBOTS FOR RAPID DEVELOPMENT OF DYNAMIC LOCOMOTION EXPERIMENTS

Jimmy Sastra

Mark Yim

In locomotion research, prototypes ranging from purely passive mechanical linkages to full-fledged autonomous mechatronic machines are built to validate locomotion principles and explore different morphologies. Being able to quickly build robotic prototypes has the capability to improve workflow, productivity, and innovation. Modular Robots, for instance, allow us to build robots quickly, and rapidly explore different morphologies.

We present the design and development of a Modular Robot system called CK-Bot. One of the major innovations of this system is a connection mechanism that allows the robot to be instantaneously reconfigured manually, while still maintaining a robust connection. We show the practical utility of rapidly building machines with modules in product design, and emergency response, but choose to focus on dynamic locomotion research. To show that this system can indeed be a useful tool for dynamic locomotion research, we use two of the prototypes and analyze their dynamic locomotion principles.

The first locomotion principle is a loop configuration that uses a sensor-based feedback controller to achieve dynamic rolling. The robot senses its position relative to the ground and changes its shape as it rolls. Using simulation and experimental results, we show ways in which the desired shape can be varied to achieve higher terminal velocities. One of our major findings is that more elongated shapes achieve higher terminal velocities than rounder shapes. We also show that rounder shapes have lower specific resistance and are thus more energy efficient. The highest velocity achieved in this work is 26 module lengths per second (1.6m/s), which is believed to be the fastest gait yet implemented for an untethered modular robot.

The second locomotion principle is a novel biologically-inspired legged style of locomotion. Passively compliant leg attachments are utilized to achieve a dynamic running gait using body articulation. We used gradient search to optimize the running gait parameters on two sets of legs with different stiffness. With experimental data and analysis we show that the softer legs run like a Lateral Leg Spring (LLS) model.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Prototyping Matters

In a study by Tom Wujec called the Marshmallow Challenge [99], teams were given 20 sticks of spaghetti, one yard of tape, one yard of string, and one marshmallow. They were asked to build a tower in 20 minutes. The marshmallow had to be at the top of the tower and the team with the highest tower would win. Wujec found that among the teams that consistently did the worst were recent business school graduates. Among the best were kindergarteners. How did the kindergarteners beat the adults? Wujec realized that business students spent much of their time creating a single plan, then executing and building only a single version of the tower. Conversely, the kindergarteners immediately started playing with the supplies, builing numerous prototypes while iterating their designs and learning from each step.

This wasn't the first time that experimentation trumped extensive planning and expertise. On December 17, 1903, for instance, two bicycle mechanics were the first to pilot a powered airplane [11]. Their discovery was to combine two fundamental principles of flight: using lift of the wings and controlling lateral axis by twisting them, they could successfully pilot their machine through the air [45]. How did the Wright brothers, who were bicycle mechanics, beat out researchers and scientists like

Samuel Langley, an educated scientist, director of the Smithsonian Institution, and backed by a large grant and team of engineers? – The Wrights started experimenting with numerous prototypes using kites and gliders. Through this experimentation, they came to understand the individual components and sub-principles needed for flight, and gradually they improved their designs. Langley, on the other hand, made only two planes. Both of which he launched off of a houseboat. Both flights ended in a crash.

Today, one of the frontiers that scientists, inventors and engineers are still working to conquer is legged locomotion over rough terrain. No researcher yet, however, has succesfully applied the fundamental principles of legged locomotion to the design of a practical machine. As a result, such legged locomotion remains largely hypothetical and is still regarded as unsolved [72]. It has been shown that mobility, speed and efficiency can be dramatically improved by moving *dynamically.* Machines have been constructed that can take advantage of the velocities and kinetic energies of their masses as they walk [24], run [14], hop [75], or undulate [44]. The actual building of these physical machines has been an important part of dynamic locomotion research. The consideration given to designs of these mechanical structures and the algorithms required for controlling the robot have offered insights that might not have been gained through pure simulation or studies of animal locomotion. By using modular hardware though, one can quickly prototype the physical proof of a concept, like a gait, allowing researchers to move quickly from hypothesis to experiment, while also quickly repeating experiments with different morphologies by reconfiguring the modular robot rather than having to build a robot from scratch for each different hypothesis.

1.2 Dynamic Locomotion

We first distinguish between quasi-static and dynamic motion. In quasi-static motion the robot maintains static equilibrium throughout its motion. At any point the robot can freeze its joints and the robot does not fall over. This imposes a speed restriction since velocities must be limited in order to minimize inertial effects. As a result, dynamic motion could thus provide faster robots. Another necessary, but not sufficient requirement is that the center of mass of the robot is always above its support polygon. For legged robots this means that feet have a large support area like in most bipedal robots [41], or at least four legs, and more commonly six legs are needed. Robots using dynamic motion, however, can have fewer legs [59].

To control a static machine, one can simply solve the transformations at the joints. Taking the ground contacts into account and solving the resulting closed-loop constraints one can directly solve and manipulate the position and orientation of the robot using the joint values of the actuators. Controlling robots to move dynamically, however, is a much harder problem. The orientation and position of the robot can only be influenced indirectly, by making the system bounce, tip, or fall over [74].

The study of dynamic locomotion started many decades ago. In Table 1.1 are listed some of the major milestones in dynamic locomotion research.

In 1983 Marc Raibert hypothesized that running is like jumping on a pogo stick. He built the first one-legged robot that could run dynamically. Its leg consisted of a pneumatic cylinder with electronic solenoid valves that could trap the air in the chamber to make the leg springy or to impart a thrust. Two other pneumatic actuators acted as an antagonistic pair to control the angle the leg makes with the body. The robot was propelled by controlling three parameters: 1) hopping height: by applying a thrust during the stance phase of the leg for a specified period; 2) forward speed: by choosing a touch-down angle for the leg during flight; 3) body attitude: by applying an appropriate torque between the body and the leg during

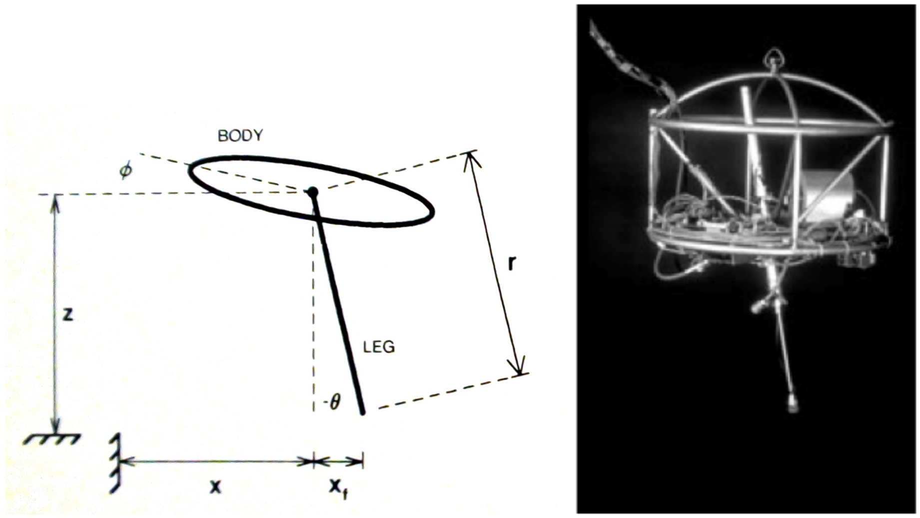
When: Who	Hypothesis	Robot Experiment
1984: Marc Raibert [71]	Running dynamics is like jumping on a pogo stick	Raibert's Hoppers use pneumatic legs to achieve dynamic running
1990: Tad McGeer [59]	Walking dynamics is similar to a rimless wheel with spokes	Passive Dynamic Walkers uses mechanical linkages only (no actuators) to walk down a slight incline
1993: Shigeo Hirose [42]	Snakes locomote by undulating their skin and making use of the differential friction of their skin	ACM (Active Chord Mechanism) snake-like robot achieves differential friction using passive wheels
2001: Martin Buehler [94]	Cockroaches can run fast over rough terrain due to an open loop tripod gait with compliant legs	RHex (Robotic Hexapod) is has six rotating compliant legs and is the fastest and most efficient running robot over rough terrain to date

Table 1.1: Dynamic Locomotion Milestones

stance phase [71].

Tad McGeer showed that the dynamics of walking are similar to those of a rimless wheel with spokes going down a shallow incline. Using only mechanical linkages, his passive dynamic walkers could start on a shallow slope and settle into a steady gait comparable to human walking. The robots used only gravity as a means of energy input, showing that walking machines could in fact be incredibly efficient on flat terrain, almost as efficient as a wheel [59].

Today, the fastest and most efficient running robot is the RHex robot developed by Martin Buehler et al. Realizing that modern-day actuators are designed to spin continuously rather than contract and expand like a muscle, Buehler's team designed a shoe-box-sized robot whose legs can rotate continuously. Believing that cockroaches can run fast over rough terrain due to an open-loop tripod gait with springy legs, the RHex robot utilizes six rotating compliant legs with an open loop tripod gait allowing it to run at speeds of up to one body length per second and can traverse height variations well exceeding it's body clearance. [94]



(a) Hopping Hypothesis (b) Hopping Robot

Figure 1.1: Hopping robot developed by Dr. Marc Raibert at MIT Leg Lab who hypothesized that running is like jumping on a pogo stick. This hypothesis was realized using a one legged robot with a pneumatic cylinder functioning as the spring as well as the actuator.

Inspired by snakes and hypothesizing that locomotion occurs by undulating their body and utilizing the differential friction of their skin, Dr. Shigeo Hirose made robots with snake-like bodies using passive wheels to simulate minimal friction along the body in the fore aft direction and maximum friction from side to side in the lateral direction [42].

1.3 Modular Robots

In this section we look at some of the major milestones achieved in the field of Modular Robots. Many research groups around the world have constructed a variety of designs, making Modular Robotics a well-established field. Table 1.2 lists some of the major milestones in modular robotics hardware.

CEBOT [34], the first modular robot, was built in 1988 by Toshio Fukuda.

When: Who	Robot	Capability
1988: Toshio Fukuda [34]	CEBOT	fist instance of a modular robot as a manipulator
1994: Mark Yim [103]	Polypod	chain style modular robot with locomotion capabilities
1998: Satoshi Murata [63]	M-Tran	self reconfiguration
2005: Wei min Shen [81]	Superbot	distributed hormonal control
2005: Hod Lipson [112]	Molecubes	self replication

Table 1.2: Modular Robot Milestones

Fukuda proposed a method to determine module type, arrangement, degree of freedom, and link length.

Engineers working on the PolyBot systems [103] (whose predecessor was PolyPod [100]) have pioneered the integration of electrical components, including IR LEDs, Hall Effect sensors on brushless motors, SMA undocking, accelerometers, and a ratchet brake onto a small modular robot. Common modes of locomotion for a modular robot (e.g., rolling, crawling, climbing, etc.) were introduced using this system. It is a dedicated chain-type modular robot unlike CKBot [67], M-TRAN [55], and SuperBot [81] which employ hybrid systems that support both chain and lattice configurations.

The primary strength of the M-TRAN systems is self-reconfiguration. M-TRAN I [63] can connect modules with electro magnets; M-TRAN II [47] uses a shape memory alloy (SMA) actuated latching mechanism. Self-reconfiguration and cluster flow of modules is a central part of the group's research. Similarly, the ATRON robot [65] also features an integrated latching mechanism which is central to its design and is shaped like a sphere.

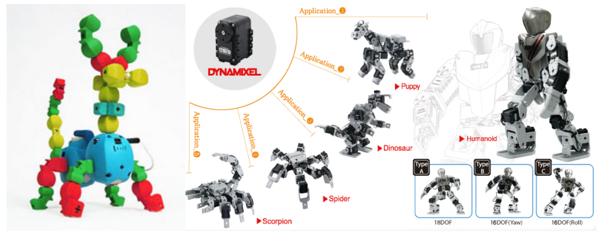
USC's SuperBot [81]features some of the latest work in state-of-the art modular robotic hardware. Borrowing the two-cube design from M-TRAN and adding a third "twisting" degree-of-freedom, SuperBot is kinematically less constrained than similar systems. However, self-reconfiguration has not yet been achieved with SuperBot, as it currently relies on manual assembly. Though SuperBot is a hybrid modular robot, it has not yet demonstrated effective use of its lattice structural abilities.

Cornell's Molecubes [112] feature unique kinematics with their 120° rotational swivel joints on the (111)-plane. This allows for simple "picking-up" motions from "feeder stations" using novel switchable magnets as required for its task of self-replication. Recent developments for an open-source system has encouraged experimentation and development of molecubes for more general applications [111].

By restricting themselves only to snake configurations, a number of research groups have explored gaits using snake-like robots that have a modular design [44]. Most notably, Howie Choset's group has built a 16-module chain of aluminum modules with hobby servos [98], having removed and replaced their electronics. The researchers have focused on using sinusoidal control to create useful locomotion modes, such as linear progression, sidewinding, concertina, and lateral undulation, climbing up trees, while offereing kinematic analysis of these gaits.

In addition to those developed in the academic circle, over the past couple of years modular robots have shown up in the commercial sphere as educational toys. In particular, Topobo [70] as shown in Figure 1.2a was designed specifically to teach children at the age of 5 and above about motion. Active modules contain one axis of rotation using a servo motor and passive pieces come in a straight, tetrahedral, T, 90 degree and elbow geometry. Communication and power are carried on a wire that must be manually routed separately. Programming of motions is done using live record and playback of the motions. One can press record, manipulate the modules, then hit play to play back the same motion. Alternatively, a Queen module is used to achieve a joystick-like control by making other active modules mimic the joint angle of the Queen module.

Just as Lego teaches children in a playful way about static structures, Topobo can instruct children about kinetic mechanisms, helping them to develop intuitions, while teaching basic concepts in physics, such as balance, center of mass, coordination,



(a) Topobo Modules (b) Bioloid Modules

Figure 1.2: Modular Robot toys

relative motion, movement with multiple degrees of freedom, and their relationship between local and global interactions. Topobo motors are limited in actuation torque, low in precision and lack a programmatic interface. So while these are useful as instructional toys for children at the age of 5 and above, they are, however, not sufficient as research tools at the graduate level in academia for children at the age of 20 and above.

A more serious commercial modular system which is better suited for researchers is developed by Robotis [80] and called Bioloid (as shown in Figure 1.2b), whose modules are called Dynamixel. Dynamixel modules have much higher torque and allow position control as well as sensor feedback on position, speed, temperature, current and voltage draw. The modules are centrally controlled using a CM-5 module that stores and executes programs and communicates to the modules using serial communication. Servo modules need to be screwed together using passive pieces [70].

These examples show that research into modular robots has thus far focused on reconfiguration and distributed control, meaning its modules have been heavily geared down and are often slow. Such modular robots have not been designed to achieve dynamic locomotion. In the same way no research has seemingly been done to analyze the mechanics of a modular robot gait.

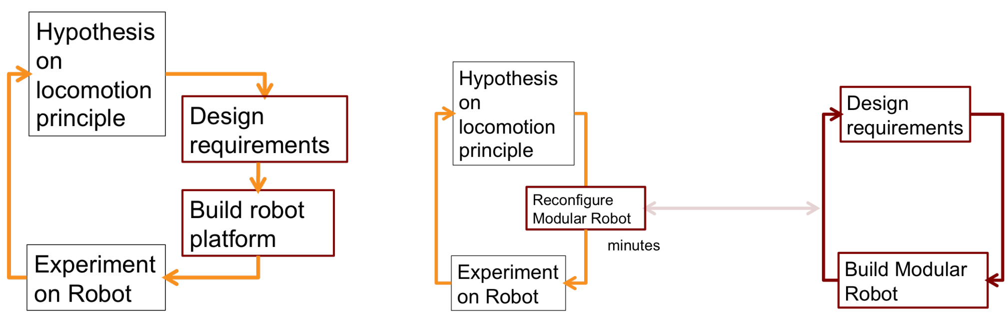
Reconfigurable modular robots have been proposed as a platform to study different locomotion modes in [110], and indeed, hundreds of locomotion modes have been demonstrated [15, 16, 20, 48, 101, 104, 105, 106]. However these gaits are for the most part, statically stable. Moreover, researchers chose not to measure the performance of such robots, such as their speed or their specific resistance. In this work we'd like to specifically investigate dynamic locomotion with modular robots. One group that has also shown significant progress toward this goal is the Locomorph group with their robotic construction kit called LocoKit [56].

1.4 Aim

Researchers today strive to understand dynamic locomotion, and to find a practical solution for making machines capable of moving over rough terrain. Furthermore, there is also the need to improve the speed and efficiency of traversing uneven terrain with these research platforms. As we saw above, one popular way of solving these problems is by building machines that locomote using the mechanical principles one desires to study. For example, Marc Raibert hypothesized that running is like bouncing on a spring [75]. By building a robot that runs using this principle, he gained many insights into the dynamics of running with compliance. We can learn from what works for the mechanical structure (a springy leg) or by seeing which control algorithms succeed or fail on such a dynamic system. As a result, the field of dynamic locomotion research is very hardware oriented. Being able to quickly build robotic prototypes changes the scientist's workflow and improves productivity and innovation. One approach to building robots quickly and rapidly exploring different morphologies is the use of Modular Robots.

The typical work flow in dynamic locomotion research is illustrated in Figure 1.3a. The scientist (graduate student) comes up with a locomotion hypothesis, then formulates requirements through analysis and simulation, spending weeks, months, or even years building a physical robot platform (Jonathan Clarke, personal communication, July 11, 2012). Only then can such researcher finally run his or her experiments on the robot, before returning to the original hypothesis. By the time this whole cycle has finished, months, years – even whole Ph.D. thesis – may have elapsed.

In our research, we propose the use of reconfigurable modular robots as the hardware platform for dynamic locomotion research. Rather than reinventing the wheel and building a whole new robot from scratch for each locomotion hypothesis, one can instead quickly reconfigure the modular robot in minutes and iterate through the hypothesis-experiment cycle rather quickly. The result is a different research process that allows not only for a *faster* experiment cycle, but also for one that runs *independently* of the build cycle, so that it can run in parallel with the experiment cycle (as shown in Figure 1.3b).



(a) Traditional Approach (b) Modular Approach

Figure 1.3: Research Work Flow

1.5 Thesis Proposal Outline

This dissertation follows the roadmap outlined in Figure 1.4. Chapter 2 describes design of both hardware and software used in the CKBot modules. Chapter 3 will demonstrate the ways in which one can effectively use these modules to rapidly build robot prototypes in three scenarios: emergency response, product design and dynamic locomotion research. Out of these three scenarios we chose to focus on the utility of our method in dynamic locomotion research and the next two chapters thus describe two examples of the hypothesis-experiment cycle. Though many gaits

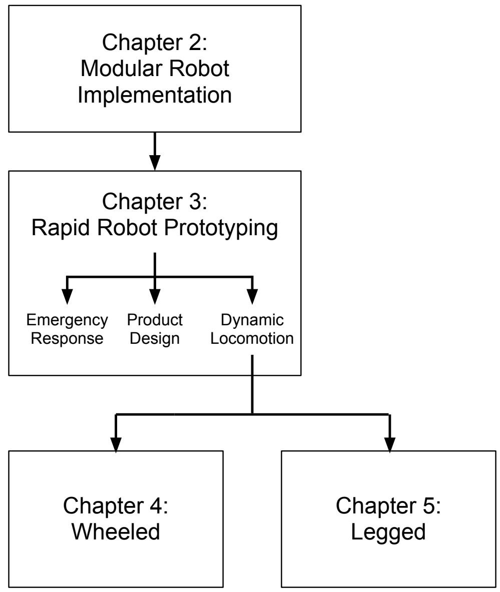


Figure 1.4: Thesis Roadmap

have been implemented using CKBot, we focus only on two of particular interest. Chapter 4 describes a wheel-like locomotion in which CKBot is in a loop configuration like a wheel with no spokes continuously deforming its rim such that it rolls forward. Chapter 5 describes a legged mode of locomotion in which CKBot is in a snake-like configuration with the addition of passive compliant legs. Finally, Chapter 6 provides a small summary, short discussion, and potential future work.

I have tried to describe topics in a manner that will hopefully be interesting to designers, engineers as well as scientists. Chapter 2 will likely be most interesting to engineers as it details the technical implementation of CKBot. Designers will be most interested in the rapid robot prototyping design process and philosophy in Chapter 3. Chapter 4 and 5 is the most scientific, and will be of most interest to researchers in locomotion.

Alternatively, one can look at the chapters in light of the workflow illustrated in Figure 1.3b. Chapter 2 falls under the build cycle where all the engineering work happens and Chapters 4 and 5 can be categorized in the experiment cycle which is the work of scientists.

1.6 Contributions

The main contribution of this thesis is to show that chain-style modules can be a useful tool for rapidly building robots and more specifically for building lower resolution physical prototypes for dynamic locomotion.

This required the development of a modular hardware that can indeed achieve dynamic locomotion and was easy to reconfigure. This was achieved using CKBot, a modular robot that has fast, high-torque modules and a fast, robust, manual reconfiguration mechanism.

Furthermore, to show that the CKBot system is indeed a useful tool for dynamic locomotion research we provide two case studies as an example of the type of analysis that can be done.

First we develop and analyze a dynamic rolling gait – the fastest and most efficient gait in the Modular Robot community to date.

Second we develop and analyze a legged-type of locomotion –the first example of a compliant legged robot with an undulatory backbone driven solely by body articulation.

Chapter 2

Design and Development of CKBot

Modular Robots are systems that promise to be versatile, robust, and inexpensive [108]. We aim to explore the feasibility of using such systems as tools for rapid robot prototyping. The CKBot modules have been designed with the intent to do this specifically for emergency response, product design exploration, and dynamic locomotion research. We describe their implementation in greater detail in this chapter.

The design goals of the CKBot modules are two-fold. Firstly, the system should decrease development time. By using building blocks that provide actuation and have a standardized interface for programming motions, we can protoype robotic solutions more quickly. Secondly, the resultant hardware/software system needs to encourage an exploratory process in which different ideas can be rapidly tried and iterated. To achieve this we leverage the modularity of the system and enable rapid reconfiguration. Additionally, the software provides an interpretive environment with a command line interface so that code can be explored quickly on a cluster of modules.

We have made seven design iterations of CKBot over the past seven years which explore the different functional requirements of the modules which we categorize as: reconfiguration, actuation, power and communication. While most efforts in the Modular Robotics community have explored technologies for self-reconfiguration [107], we stress the importance of manual reconfiguration for the applications of prototyping. Actuation will be provided by servos to enable cost reduction. Power will be supplied locally using lithium polymer batteries or tethered using a DC power. When tethered, we reduce the amount of current carried across modules by converting locally to a lower voltage. We have explored two methods of wiring to transport power and communication across modules. Communication is achieved on a global bus within a cluster of modules as well as locally between modules.

In addition to hardware, a software interface that allows the user to develop quickly and encourages exploration is of equal importance. A graphical user interface has been developed which lets users without a programming background teleoperate a configuration of modules. This was particularly evident in robot competitions we've held in which competitors learned the interface and implemented a task in a matter of hours [102]. A scripting interface has shown to be more of a learning curve, but provides a powerful programmatic interface [78]. It's scripting nature also encourages exploration since code can be tried on the fly and the user does not have to follow the traditional compile-debug cycle.

2.1 CKBot Hardware

Modular Robots can be categorized in chain- and lattice-style. Chain-style modules form articulated serial chains. They tend to be more practical and applicable to standard robotic tasks. The lattice style modules are restricted to moving in a specified grid. This makes planning the motions for reconfiguration easier since the space of possible motions is smaller, but they are less practical for the tasks we are interested and for that matter less applicable to traditional robot tasks in general

[66][107]. Choosing practical application in favor of self-reconfiguration, CKBot is desigend to be a chain-style modular robot.

CKBot modules have two different form factors as shown in Table 2.1: a *U-bar module* in the column labeled V1.2 and an *L7 module* in the column labeled V1.3. The two modules are very similar. Both have a one-rotational degree of freedom, but have slightly different kinematic structures. The U-bar is made up of a U-structure and a bar; the L7 is built using an L-shaped and 7-shaped structure. These differences allow the modules to connect on different rotational axes. For example, each module can rotate ±90 degrees around a single axis. Attaching a U-Bar in between two modules one can pitch two modules with respect to each other. If one puts a L7 module instead, the two modules can pitch or roll with respect to each other depending on the orientation of the L7 module in between.

The frame of V1.1 through V2.1 are made out of of acrylonitrile-butadiene-styrene (ABS) and can be easily built using a lasercutter and Epoxy. In nominal, or zero degree configuration, the module resembles a cube 60mm on each side. We have used hobby servos in different sizes. The mid-range is 30mm from the rotational axis to the bottom of the servo, not including the mounting brackets which were lopped off. V2.2 was made out of laser cut sheet aluminum and uses a Dynamixel EX-106+ servo. This was a larger servo and in zero degree configuration the module's dimensions are 60x60x90mm.

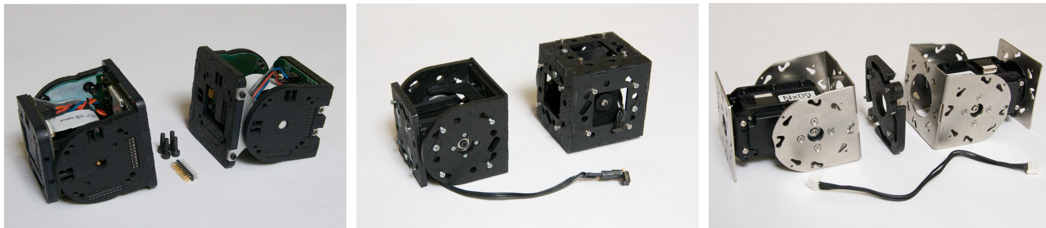
2.1.1 Reconfiguration Mechanism

One of the issues in building rapidly reconfiguring systems is making the bonds between modules strong. There is typically a trade-off between strength and ease of reconfiguration. The classic example is the popular Lego toy system. Lego is a heterogeneous reconfigurable modular building system, with pieces that require little effort to assemble, but can also come apart easily under significant mechanical loads. On the other hand, systems like Bioloid are assembled using screws. While these are very strong bonds they take a long time to put together.

The method with which the CKBot modules can be connected have gone through three different design changes as shown in Figure 2.1. In the first version modules were connected using M3 screws. One module has four faces. Each face is patterned with four through holes and four tapped holes each. Modules can be connected by alligning the tapped holes with the through holes. This can be done in one of four allignments at 90 degree intervals. Connecting two modules together nominally requires eight screws, but four are sufficient. Manufacturability would be better if we used nuts and bolts to connect alligning through holes, however assembly time would be longer and more difficult in handling modules, screws, and nuts, and the tools required to hold the nut and turn the screw. Using tapped holes featured inside the modules still proved easy to manufacture.

The second design introduced a system we call the ModLock mechanism. Each face featured a hole pattern and protruding screws. In this method the user could instantaneously snap the modules together, while still creating a strong bond between the modules. No screws or tools were required to put the modules together. More effort was required to manufacture these modules, because each face of the module contained five screws. However the typical trade off between strength and ease of assembly did not have to be made. In our third version, these screw features were put in a separate connection plate and only the hole pattern had to be featured into each module face. This made manufacturing much easier.

The ModLock mechanism using the connection plate is shown in Figure 2.2. To connect two modules a *male connector plate* is used to connect the two female patterns. Effectively, it operates as two gendered connections, in which any system module holds a female connector and a male connector couples the female patterned modules together. Each male connector has four flat head screws embedded into the plate. The screws however are not turned. Rather the head geometry acts as hooks. To connect two modules together, the screws are aligned to fit into four custom



(a) Using conventional screws to connect modules

(b) ModLock mechanism in faces of modules. No attachment plate needed

(c) ModLock mechanism in separate attachment plate. Easier to manufacture

Figure 2.1: CKBot Manual Reconfiguration Mechanisms

shaped *screw insertion holes*. The plates are pressed together so that the heads go through the screw insertion holes (Fig. 2.3a). The faces are twisted 12° to align and tighten the male connector plate assembly with the female pattern. A stainless steel *locking dowel* on a compliant lever snaps into a *lock groove* to prevent the faces from untwisting. This is shown in Figure 2.3b. Once the faces are tightened with respect to each other and the locking dowel slides into the lock groove, the connection between the male connector plate and the female connector is complete. A remaining female connector is secured to the other side of the male connector plate to join the two female connectors together.

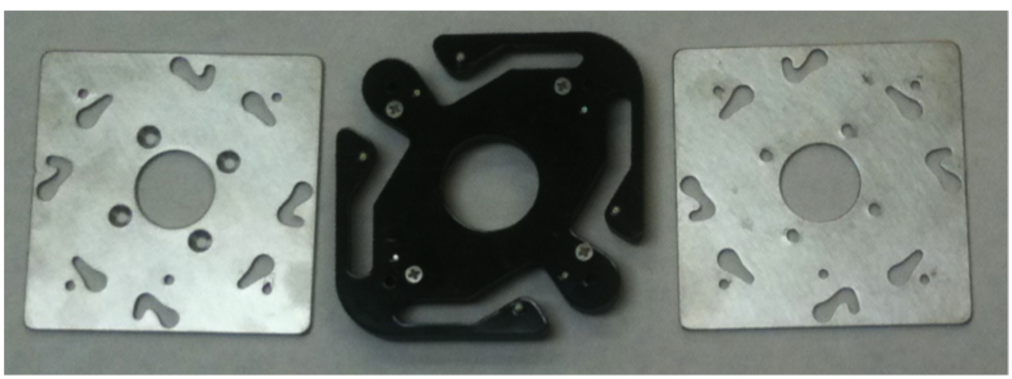


Figure 2.2: ModLock pattern

To disconnect the modules, a user pushes on opposing compliant levers to disengage the locking dowels. He or she then twists the male connector plate assembly in either direction to disengage the screw heads. Modules can then be effortlessly

(a) Male connector plate assembly and female pattern are pushed together

(b) Connectors twist to a lock

(c) Bi-directional unlock

Figure 2.3: Method of connection

pulled apart.

2.1.2 Actuation

We used hobby servos as they provide great power density while still being very small and lightweight. In addition, hobby servos are far less expensive than other COTS motors, such as those provided by Maxon. Unfortunately, hobby servos proved less reliable and lacked the tolerances one might find in its more costly Swiss counterparts.

We have used the following servos: Airtronics 94359, Hitec HSR5990TG, and Dynamixel EX-106+. See Table 2.1 for specifications. These hobby servos are priced at \$79.89, \$199.85 and \$499 respectively at a quantity of one in 2011. With this particular set of three, as they increase in price, they also increase in power consumption and torque output, but they decrease in speed. One of the limiting factors of a Modular Robots is torque, or more precisely, specific torque: a number specifying how many modules can be cantilevered. Modules with greater torque output can create larger arms which enable more robotic tasks, which enables making more practical robots. Speed in actuators is important to be able to take advantage of inertias when performing dynamic locomotion.

2.1.3 Power

A module can be powered with either batteries or a tether. The ports carry two power lines of 7.4V and 24V. When using tethered power, a higher voltage of 24V is carried on the power bus across the modules. It is regulated down locally to 7.4V. to power the servos of the module. This strategy lets us carry less current across modules. When untethered battery packs are conected locally and current of all modules does not need to go through the entire serial chain.

We have explored two wiring strategies: internal and external. All V1 modules were *internally* wired. These modules have four PCBs, one on the internal side of each of the four faces. All four PCBs are connected electronically using wiring or solder joints. Each module also has eight identical electrical connection ports, seven around four faces of the robot and one that is internal. These ports can be used to electrically connect the modules together on the outside of each face using a double sided pin header as shown in Figure 2.1a. V2 modules had to be wired *externally*. Modules have one or two electrical ports on the inside of the modules. The user has to manually connect power and communication using wires. This looks more messy, but simplifies the design of the modules. Simplifying design usually results in modules that are more robust.

2.1.4 Communication

There are two types of communication on the module: global communication, based on CANbus, and local communication (between two modules) through IR pairs on each face, based on serial communication. The global bus is called the Robotics Bus [38], which is based on CAN (Controller Area Network) [31]. It is passed through the electrical connections of the 20-pin sockets on each face, as shown in Figure 2.4. We run the CAN bus at 250Kbits/sec for reliability even though it can run up to 1Mbits/sec. A global bus is easy to use and allows us to easily browse modules and design gaits.

Local communication is achieved through infra-red (IR) pairs on each face as shown in Figure 2.4. Communication is achieved using serial communication at a baud rate of 9600. A custom designed opamp filter allows us to calibrate the sensitivity and filter out noise. Local communication allows each module to talk to its neighbor, so that it discovers which module is connected to which. This operation has been useful in detecting the configuration of the connected structure.

A wireless communication bridge has been made using XBee [28] for Version 1.1 through 2.1. Wireless communication for Version 2.2 was achieved by setting up an ad-hoc wireless network using a Gumstix Overo board.

Servos have different kinds of input. The angular position of the airtronics 94359 is set using PWM. The Hitec HSR 5990TG supports both PWM and a bi-direction serial communication bus on a single line. The EX-106+ servo communicates on a RS-485 bus with a master-slave arrangement.

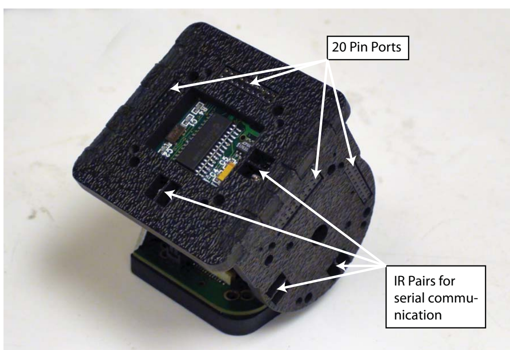


Figure 2.4: CKbot V1.3 U-bar module labeled

2.1.5 CKBot Module Iterations

The first design (V1.1) features a single servo, a PIC microcontroller and a CAN transceiver. A ribbon cable was used to transmit power and communication from one side of the module to the other. This ribbon cable proved to be the most common failure mode, sometimes breaking itself in minutes of use through repeated bending.

Version 1.2 aimed to improve reliability as well as manufacturability. Using pads

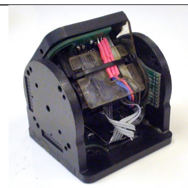
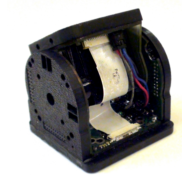
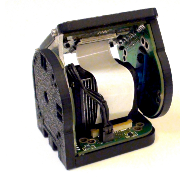
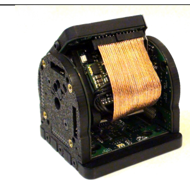
Version	V1.1	V1.2	V1.3	V1.4
Picture				
Servo	Airtronics 94359	94359	Hitec HSR5990TG	HSR5990TG
Torque	1.4Nm	1.4Nm	2.94Nm	2.94Nm
No load speed	0.10 sec 60deg	0.10 sec/60deg	0.12sec/60deg	0.12sec/60deg
Cable	Ribbon	3 inch FFC	3 inch FFC	High flex ribbon

Table 2.1: Four iterations of CKBot V1. These type of modules have internal wiring and electrical ports on each face which expose power and communication. Modules can be connected using a pin header. They don't require external wiring.

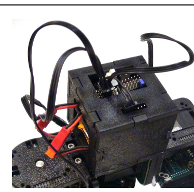
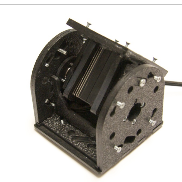
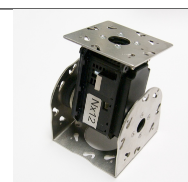
Version	V2.0	V2.1	V2.2
Picture			
Servo	HSR5990TG	HSR5990TG	Dynamixel EX-106+
Torque	1.4Nm	1.4Nm	10.5Nm
No load speed	0.12sec/60deg	0.12sec/60deg	0.143sec/60deg
Cable	Hitec servo cable	Hitec servo cable	Dynamixel cables

Table 2.2: Three iterations of CKBot V2. These modules have one or two electrical ports inside, and need to be wired externally.

at the edges of the board and applying solder in between to connect the boards together, we could electrically the assemble modules more quickly. By replacing the ribbon wire with a Flexible Flat Cable (FFC), we were able to increase the lifetime of the modules which are designed for repeated bending. We also added IR functionality to allow local communication.

We changed the servo in Version 1.3 from the Airtronics 94539 to the Hitec HSR5990TG, doubling the amount of torque, at a comparable no-load speed. In addition the servo was opened and the MCU inside was reflashed allowing us to run our own control loop and the servo at regimes the factory code does not.

In Version 1.4, the FFC cable was replaced with a high-flex-life ribbon cable. In addition, the IR ports were replaced with IrDA: by doing so, communication could be modulated onto a higher frequency and be transmitted over IR pulses. This allowed for faster, more robust, and more long-range communication. This version used a ATXMega for IRDA communication and PIC for CANbus.

In all Version 2 iterations, modules required external wiring. Version 2.0 was built with a Pololu servo controller, which would command all servos directly, using a ZigBee wireless module attached to command the Pololu [79]. All servos had to be wired directly to the central Pololu controller in a star shaped configuration.

Version 2.1 took advantage of the bi-direction serial communication bus over a single line. In this configuration the modules could be daisy chained, though still required external wiring.

Version 2.2 used Dynamixel servos. External wiring could also be daisy chained and it's communication protocol was much more sophisticated compared to the HSR5990TG.

2.2 CKBot Software

All the iterations of Version 1 have on board computation in the form of a 8 bit PIC micro controller. Even with such a small amount of computing power there are several reasons why having the functionality of local computation on each module is a useful feature in rapid robot prototyping with modules. Firstly, it enables implementation of introspection and secondly it can expose an interface that is consistent across all modules. Both these properties can speed up the programming of the robot.

Adding an on board controller to all modules, rather than talking to each servo, camera, or other sensor directly allows us to store information about modules in a unified manner. The module can provide information about its capabilities and expose parameters that can be read or written to. This type of introspection then allows the user or the client to query properties of an individual module. The user does not need to be familiar with the specifications of the module beforehand, but can discover them through the client. Similarly a whole cluster of modules can be inspected via the client interface without having a priori knowledge of what is connected. For example, the user can query a servo module and discover it has capability to set its angular position. It also exposes sensors who has parameters that can be set such as frequency at which feedback is read and reported. The list also shows its local communication capabilites.

Secondly, the added on board controller can provide a standardized interface to communicate to all modules which simplifies code at the client side. Being able to use the same software to communicate with all actuators and sensors makes for cleaner code and shorter development time.

Thirdly, the added on board controller can use local communication to detect how modules are connected and thus the robot's overall configuration. Local communication is provided by seven IR pairs in all four faces of the modules. Having knowledge of the morphology may allow the client to be smart in aiding the user by looking up gaits implemented with this morphology [67]. This feature holds promise to speed up development time, and make programming motions quicker, but still requires further development to be of practical use.

One of the trade-offs of adding an on board controller is that more electronics need to be physically accomodated inside each module. In Version 2 we explored talking directly to servos and having no on baord computation to simplify module design. Hobby servos generally only expose PWM input to set angular position. In Version 2.0 we use a Pololu Controller which has 8 ports which supply PWM to individual servos. The pololu board has a serial interface. Version 2.1 takes advantage of a serial communication bus that is also enabled on these servos and exposes a limited set of properties: get/set angular position. Lastly, the Dynamixel EX-106+ is a high end hobby servo already exposes a rich set of properties via a RS485 bus.

2.2.1 Robotics Bus Protocol

The communication protocol in Version 1 modules is called Robotics Bus [38]. It is built on top of a CAN bus [31], a low-bandwidth protocol which we run at 250Kbits/sec and is lightweight so it can be implemented on PIC18f2680 microcontrollers [61]. These are inexpensive eight bit micro controllers with only 3.3KB of RAM memory. Robotics Bus is designed to support distributed processing. One of its unique features, making it particularly useful for CKBot and modular robots in general, is its browseability.

It is a global bus protocol. Modules that are connected on the same communication line show up as inidividual nodes and have a unique Node ID. Only one node can transmit at a time while all receive the transmission at the same time. A diagram of this topology is shown in Figure 2.5. Modules that are on the same CANBus are called a cluster of modules.

The Robotics Bus protocol describes four types of messages: heartbeat messages,

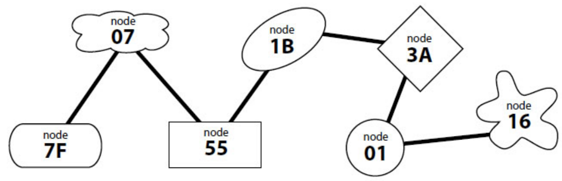


Figure 2.5: A Cluster of Robotics Bus Nodes

dictionary request messages, dictionary response messages, and process messages. Heartbeat messages are sent out once a second by each node. Typically at startup, the client listens for two seconds on the bus to discover which nodes are connected on a cluster. A node's parameters also known as objects can be configured using dictionary request and response messages. Process messages are short messages to send raw data with little protocol overhead to be used for messages that are used frequently and time critical.

Each node features an Object Dictionary as shown in Figure 2.6 which lists the features of a given module. In our example the node is a Camera Controller. It has a Robotics Bus Object named Tilt and Pan that can be read and written to. These Robotics Bus Objects can be set and queried using dictionary request and response messages. Each Object has a hex number associated with it that denotes its place in the memory. Each object contains a segment that holds a hex number to link to the next Robotics Bus Object. A client can thus walk through the dictionary to obtain all of the Objects assuming the first Object starts at 1,000h.

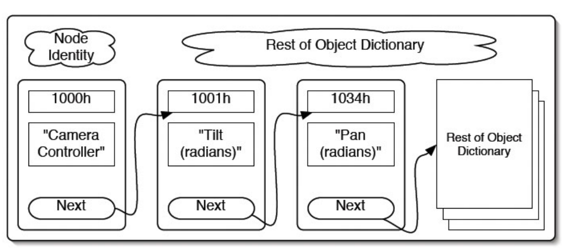


Figure 2.6: Robotics Bus Object Dictionary

2.2.2 CKBotIX Client

A client written in Python is used as a configuration tool, and a programmatic interface for designing behaviors such as executing locomotion gaits [78]. It communicates with a cluster of CKBot modules using the Robotics Bus protocol. By taking advantage of the interpreted environment of Python, code can be run immediately without having to go through the traditional edit/compile/execute cycle.

The interpretive environment allows code to be generated dynamically during runtime. We exploit this feature by dynamically generating *setter* and *getter* methods for the Objects in a node's Dictionary. A command line interface is provided using iPython which supports tab-completion and allows the user to automatically discover properties of modules in a cluster like so:

```
>>> c = Cluster()
>>> c.populate(3, {0x88:'front', 0xB9:'middle', 0xC1:'rear'})
>>> c.at.head.set <press tab>
c.at.front.set_pos
c.at.front.set_feedback
>>> c.at.front.set_pos(4500)
```

We instantiate a Cluster class which can automatically query the Robotics Bus dictionary and discover properties of the modules using a populate() method. This dynamically generates a logical view of the cluster in which modules can be named for convenience. In this example we have named them 'front', 'middle', and 'rear'. This allows the user to quickly try out code, look at data, and change variables. Additionaly these generated functions are also valuable in speeding up code development because they can be used in scripts.

The client is written to implement the Robotics Bus protocol in an asynchronous manner. The client may put in requests for description strings from different modules. The nodes will respond when they are ready and when the bus is free. The host receives these responses out of order and is responsible for collecting the segments.

Once all of the segments of a object have been received, they can be collected and reported to the user. The advantage of is that these calls are non-blocking. The client is free to execute other code while it is waiting for responses. This strategy leverages the many processors running in parallel in a cluster of CKBot modules. For example, Figure 2.7 shows the sequence diagram of asynchronous requests of description strings from three CKBot modules. The vertical axis shows time sequence of messages as they occur, and the horizontal dimension shows interaction between the different instances. In this case the instances are three modules, the client which is usually the laptop and the user. Note that the requests are non-blocking. Other code can be executed on the client side shown in red while modules respond with object segments shown in green. The update() method returns number of outstanding requests. Once all segments of all description strings have been received and collected in the buffer, update() returns 0. The client, then collects all the segments and reports the requested description strings.

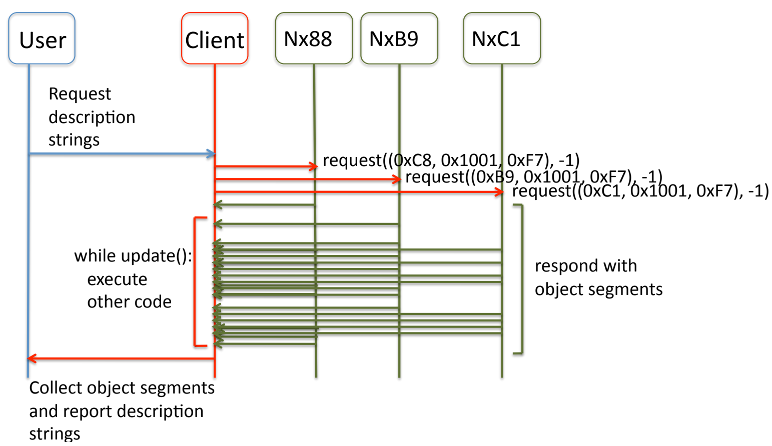


Figure 2.7: Sequence diagram of an asynchronous request of description strings from three modules.

In previous work a graphical user interface called CKBot GUI was developed. While the CKBotIX Client is extremely useful for power users, the CKBot GUI allows users to interact with point and click, rather than with text commands. In robotics competitions in which we novices used CKBot modules, we have found that this interface is extremely useful for the casual user [102]. Module's parameters can be configured, simple periodic motions can be programmed and teleoperation implemented using a joystick. These can be set up faster through the GUI compared to the CKBot IX client, but more advanced behaviors can not be implemented. The CKBot IX client runs on Ubuntu Linux, Mac OS, and Windows. The CKBot GUI, however, runs only on Windows. Future work will involve a graphical user interface built on top of the Client.

2.3 Conclusion

CKBot hardware modularity is driven by need for rapidly exploring morphologies. We believe to further this task, the manual reconfiguration mechanism is extremely important. We have presented a mechanism that doesn't need to make the typical trade off between speed of assembly and bond strength.

The CKBot software is written to provide an interpretive command line interface supporting tab-completion. This is very useful for trying out code quickly. Asynchrony of the protocol takes advantage of the fact that modules have on-board computing power. A GUI enables basic configuration and teleoperation of a cluster of modules, but nothing custom or more advanced than what has been implemented in the GUI can be programmed.

Local communication is implemented to enable detection of the configuration followed by selection a corresponding gait table from a library [67] which maps motions to isomorphic configurations. Future work could make this functionality more useful in rapid robot prototyping by letting this feature aid in development of gaits not in the library.

Chapter 3

Rapid Iterative Prototyping

3.1 Iterative Design Everywhere

Numerous examples of innovations that were the result of rapid iterative prototyping can be found in engineering [29], design [88], even in theatre [109]. The Concorde Supersonic Jet started out as a cardboard airplane [33]. The first two prototypes of the Open Prosthetics Projects were made using Lego [69]. James Dyson built 5,127 prototypes over five years before finally settling on the model for his first vacuum cleaner [29]. That design went on to become the U.K.'s best selling vacuum cleaner in only 18 months, revolutionizing an entire industry. Dyson, in fact, was only an art student – and one who didn't even know how to use a CAD program. To this day Dyson engineers still make their first models out of cardboard, glue and tape. It's quick, simple, cheap. And it seems to work

Others have followed in Dyson's low-tech footsteps. Take for instance Ideo, a design consultancy firm that evangelizes the use of low-resolution prototypes as a source for innovation. David Kelley calls it "Thinking with your hands." The methodology is more formally known as "Design Thinking" [13]. But at IDEO, a playful iterative approach to problem solving is foundational to their culture. For one of the projects, for example, the team worked up a prototype using a roll-on deodorant. This later became the first commercial computer mouse for the Macintosh. In effect, the team learned its way by building prototypes. But prototyping doesn't just solve straightforward problems. It opens up new solutions through serendipity and luck, allowing one to work more efficiently.

IDEO also believes that prototyping is a way of communicating. A message is likely to be lost when reading a report. Build a prototype and not only does it communicate your idea, but it also persuades its audience of its value. Prototyping is worth a thousand pictures [50].

The d.school at Stanford University is devoted to teaching the basic principles of design thinking in which prototyping is a large part of the process. They empphasize to just start building. Even if you're not sure what you're doing, the act of picking up some materials (paper, tape, and found objects are a good way to start!) will be enough to get you going. Don't spend too long on one prototype. Move on before you find yourself getting too emotionally attached to any one prototype [88].

Workflow has perhaps been best studied in the field of Computer Engineering. Extreme Programming [4] or Agile Planning [23], for instance, embrace iterative development. They emphasize frequent releases in short development cycles. A quick prototype is developed and continuously refined. At each iteration, when designers and engineers can see which developments have worked and how end-users have responded, design modifications can be made and value added. This process differs markedly from a more traditional Waterfall approach, a sequential design process in which progress flows from one phase to the next only when the preceding phase has been completed and perfected.

Software lends itself particularly well to the build-quickly and iterate methodology because of the reusability and modularity of its materials. Code is easy to rewrite or put into separate classes and can be done quickly. Such features, however, are not yet available for building robotic hardware. Creating hardware that is modular and reusable will give the ability to quickly prototype and iterate through numerous ideas and design variations and follow a methodology of places like IDEO.

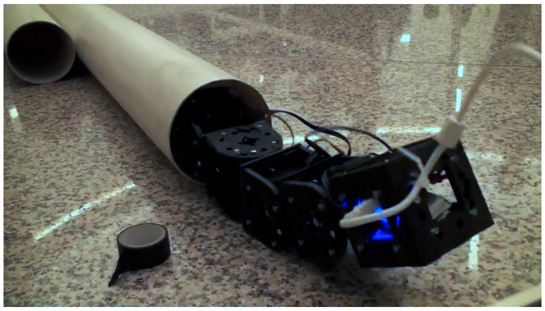
3.2 Rapid Exploration of Robot Morphologies

We have demonstrated that modules are useful tools for rapidly building robots in several scenarios. The first scenario is in the annual Planetary Contingency competition in which robots were built in three hours to solve a robotic task that required mobility and manipulation capability. Our second example comes from product design. There we used our modular robots to create prototypes that serve to convey a message and explore feasibility. Third, we have used modular robots in robotics research to demonstrate different types of locomotion principles. This section will describe the different morphologies and the time required to build them.

3.2.1 Planetary Contingency Competition

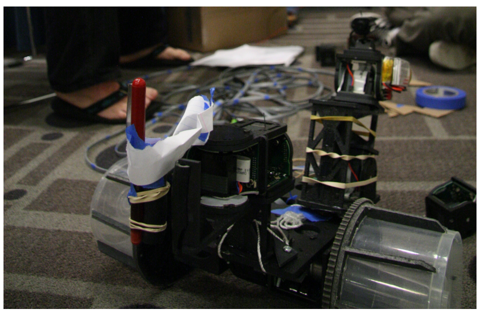
The Planetary Contingency is an annual competition that simulates an unexpected problem occurring at a planetary habitat, in which a robotic solution must be quickly developed and deployed. Teams are allowed to use only what they can carry in a container with outside dimensions summing to less than 150cm, and weighing 25kg or less. For example, a container 70cm long, 50cm wide, and 30cm tall has a total dimension of 70+50+30 = 150cm, and would be within the size limits. These limitations make the event more challenging, but also simulate the very real space and weight restrictions enforced on space missions. The actual unexpected problem to be solved is announced, the day of the competition, and must be solved on the spot using only what each team has packed in its container. Tasks were solved using the limited set of tools, and improvised in a short amount of time [102].

Figure 3.1a shows a snake configuration that was made to crawl through a three-inch pipe. Figure 3.1b shows a device made from tape, cardboard and rubber bands. It shoots a grappling hook up, before winching itself up. Two cylinders hold the

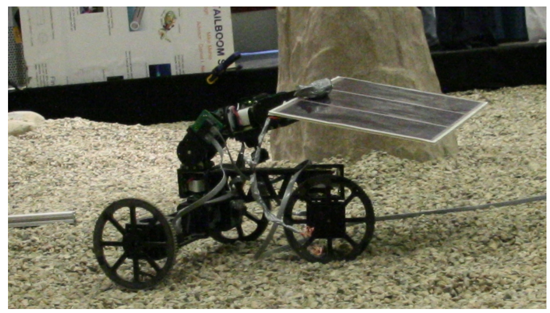


(a) snake through tube

(b) grappling hook



(c) wheels out of CD cases



(d) manipulating a solar panel

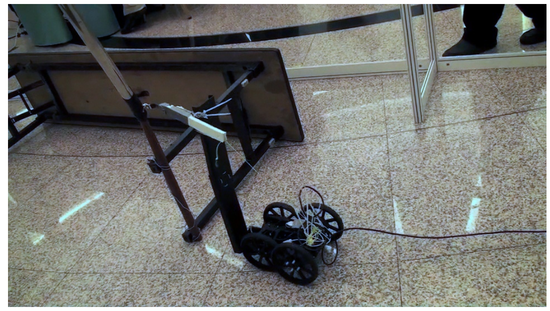
Figure 3.1: Robots improvised to solve tasks in the Planetary Contingency competition at the 2012 ICRA conference in Shanghai

spool of wire. Figure 3.1c shows a mobile manipulator with wheels made out of two CD casings and foam core treads to increase traction on the gravel. Figure 3.1d shows a mobile manipulator grabbing a solar panel (a foam core panel) and carrying it back to the base station.

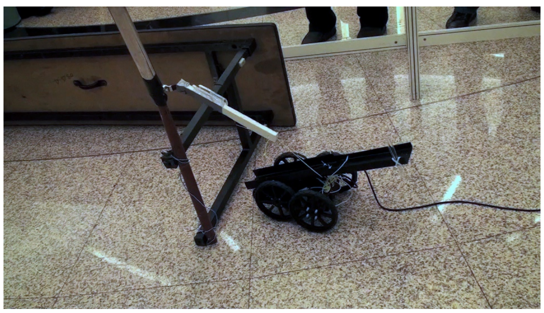
In Figure 3.2a through 3.2d the task was to turn of a brass ball valve. Besides the difficulty of being able to operate a robot to manipulate a valve, the task was even more challenging thanks to the torque required to rotate the valve. A wheeled base with an arm was constructed to attach a foam core tube to increase the lever arm on the handle. The robot could then drive and pull on the lever arm that was attached by a wire.

The competition demonstrates that robots can quickly be constructed to a custom morphology that solves a task requiring mobility and manipulation using manually reconfigurable modules.

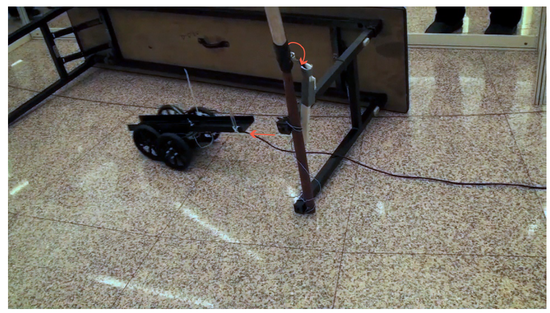
(a) approach valve



(b) attach foam core strut



(c) to increase lever arm



(d) pull to turn off valve

Figure 3.2: This sequence shows an example of a creative improvised solution during the Planetary Contingency competition at the 2012 ICRA conference in Shanghai.

3.2.2 Product Design

In product design, mockups are used as low-fidelity prototypes to convey an idea rather than a finished product. Foamcore and yellow foam are tools of choice, but can only create static objects. Willow Garage is a robotics company which is interested in putting robots in the home. At his company, we have explored the use of modular robots as a way to rapidly build functional mockups. We were interested in seeing if robots could enter the home and transition robotics from an industry tool to a consumer product. Below are a few of the ideas and prototypes that came out of that process.

SuperDolly

The Super Dolly is a powered, wheeled platform with a large flat payload surface. It has a low profile, and is similar in appearance to a standard furniture dolly. In comparison to a furniture dolly, which rides on four passive casters, the Super Dolly rides on two active wheels and two passive casters. The Super Dollys payload surface is force-sensitive: the user pushes gently on the payload object to cause the Super Dolly to drive in a desired direction.

A prototype was created by replacing two casters of a regular dolly with actuated wheeled modules. A board was placed on top with Acetal in between to decrease friction. A Space Navigator^{TM} is a six degree of freedom input device which can sense displacement in all three positions and orientations. This device was placed between the board and the dolly to measure displacements of the board. Bungee chords were added to act as springs that would provide the returning force to reset the Space Navigator to its zero position, when no forces are being applied. This setup allows us to detect forces imparted by the user in the horizontal plane, while the placement of COM on the payload does not matter. Also, forces could be imparted indirectly to the payload, and not only to the SuperDolly structure. A Gumstix board took readings from the Space Navigator and sent velocity commands to the wheel modules. A rubber mat on top of the board (not pictured) provided friction to better transmit forces to the board via the payload.

PlatformBot

The Platformbot concept is a general purpose mobile manipulator with a tray. A hinge in the base can be used to adjust its height, to fold flat to fit underneath a couch or to raise itself to manipulate objects on tall surfaces.

The prototype consists of a thirteen modules and was assembled in thirty minutes. Two wheel modules were attached to three UBar modules on each side of the base. Additionally, two modules were used for the manipulator. One day of programming could demonstrate teleop of the robot to slide an object onto its tray and to demonstrate its maneuverability as well as its ability to traverse stairs.

Mobile Bases and Manipulators

We explored a variety of mobile bases using foam core as structural material. Figure 3.5a shows the minimum degrees of freedom for a holonomic base. We used two wheeled modules and a module for a vertical rotational degree of freedom that is off axis. This was also achieved with a three-caster and a four-caster base. We used foam core and hot glue as materials. They proved easy to use and could be cut into any shape.

Mobile manipulators were similarly explored using PVC as a structural material. This proved to be a cheap and fast way to make human-scale structures. PCV pipes, which come in standard sizes and could be cut to length using a pipe cutter. Different kinds of fittings are also available for different types of joints. These materials are easily acquired at a hardware store.

VacuumBot

IRobot and Neato have shown the beginnings of a consumer market in task-specific robots– products such as the Roomba, a shoe-box sized machine that can autonomously move through the house and do a mediocre job of vacuuming the floor, in the $350-$600 price range. It might be argued that one of the main tasks that a general-purpose robot should perform in the home is vacuuming. What this task gets us for free is an air pump system for a vacuum cup end effector and blower system, as well as a license to periodically roam the house which provides a platform for developers to write apps such as room diff and home monitoring/security.

The vacuum of the air pump can be used periodically to vacuum the carpet; the end effector can dust the couch. While it is roaming the house, a Kinect sensor can be used to create a database of the objects in the house. Out-of-place objects on the floor can be picked up and collected in bins. Once the bin of collected objects becomes heavy, the blower source in the air pump can power air bearings under the bin which reduces friction for transporting heavy bins. One of the design challenges in putting a robot in the home, unlike industrial robots, is the requirement that the robot have a small foot print and low weight requirements to navigate clutter in the house and enable consumers to pick up the robot if it's in the way. Given this constraint, in-home robots have a greater difficulty exerting smaller external forces, unlike the larger heavier robots we are used to seeing in industry environments. In this instance, vacuum could provide, in addition to gravity, an additional normal force.

Our prototype consists of an off-the-shelf vacuum cleaner. We used foam core to create a bin. Two wheeled modules and a 3 module arm were attached using two-part polyurethane foam. A webcam using a $8 spoon ladle made for an omni directional camera. This prototype was made in three hours and was meant merely to convey the rough idea while exploring looks and size. It was capable of being driven by teleop and provided a 360 view with its camera.

Our second prototype offered higher fidelity. Wheeled modules were mounted using wood screws. The arm linkage consisted of foam core. We made a vacuum cup out of a rubber ball. A valve was 3D printed to allow it to select between vacuuming the carpet and actuating the vacuum cup. A kinect was also mounted and later replaced with a Hokuyo laser scanner. This prototype was able to autonomously pick up objects.

Conclusion

Building consumer robot prototypes for product design is hard. Size, rigidity, looks, all matter. The prototypes end up looking like Frankenstein machines. Such mechanical monsters are useful for gauging feasibility, telling a story or conveying an idea internally. They were not, however, used for validation interviews with outsiders because they would be too distracting. For the validation interviews with outsiders we used static foam mockups. It is clear that those mockup aren't functional. It would be easier for the interviewees to use their imaginations and not get hung up on the details.

3.2.3 Locomotion Principles

Numerous locomotion principles have also been demonstrated using our CKBot system. Figure 3.7 shows a subset of these. On the top row are examples of wheeled like locomotion. On the bottom row are legged types of locomotion.

Figure 3.7a shows a wheeled vehicle contructed using five modules and four passive wheel attachments. Figure 3.7b shows a snake-like configuration; every module appears at a 90 degree angle from each other. Unactuated wheels are attached on each side. Forward movement is achieved by undulating the body. Lateral friction of the wheels forces the body to be propelled forward. This follows the same design principle as the Hirose snake, except that its modular architecture can arbitrarily be shortened or extended. Figure 3.7c shows a ten-module configuration. The modules are arranged in a loop and the robot can roll itself forward.

Figure 3.7d shows a four legged configuration. The legs are constructed out of thermoplastic material. They can be heated then bent to the desired shape. Figure 3.7e shows a quadruped with foam legs. Figure 3.7f shows a hexapod with fiberglass legs.

3.2.4 Build Times

Table 3.1 summarizes all of the prototypes and denotes the amount of time it took to build and program each configuration.

Our timings were recorded during assembly as much as possible. However some were estimated afterwards. While these robots may only be low fidelity prototypes,

Robot	Iteration	Assemble [hr]	Program [hr]
Planetary Contingency robots	N/A	3	3
Superdolly		1 day	1 day
PlatformBot	v1	0.5	2
	v2	2	2
Base	4 caster	2	3 days
	3 caster	2	0.5
	1 caster	1	0.5
Mobile Manipulator	V1	3	1
	V2	3	1
	V3	3	1
LegBot		5	1
VacuumBot	teleop	3	1
	autonomous	7 days	6 days
Wheeled vehicle		2 min	0.5
Wheeled snake v2		3 days	3 days
Closed loop rolling	10 modules	6	6
	8 modules	3	1
	12 modules	3	1
	14 modules	3	1
Legged robots	ABS lasercut	8 hours	3 hours
	thermoplastic	8 hours	3 hours
	foam	94 mins	49 mins
	fiberglass	72 seconds	4 hours
	spring steel	3 hours	1 hour

Table 3.1: This table shows how much time it took to assemble all the components and program a behavior for each prototype

they were all constructed in no more than three days, some as few as 30 minutes. Structural components were made out of cheap materials, while actuation was achieved by CKBot modules. We measured time starting from when we decided to implement the idea. Building a module, for example, was designed without having a particular morphology in mind and was considered as one of the tools. That time was therefore not included.

3.3 Tools for Robot Prototyping

Several key technologies enabled us to rapidly build these prototypes. We will put them in three functional categories: Actuation, Connection and Structure. Examples of each main modular components are shown in Figure 3.8.

3.3.1 Actuation

The most basic building block of the system is the UBar module as shown in Figure 3.8a and the many design iterations have been described in detail in Chapter 2. They provide actuation by means of a servo with a 180-degree range of motion. Using only this type of modules, we can already explore numerous different morphologies, such as snake-like or rolling locomotion. A system made up of these types of modules is generally also known as chain-style modular robot. A second type of module which is very similar in its make up is the CR (Continuous Rotation) module. These provide a continuous rotational shaft that can be used to actuate wheels. This is shown in Figure 3.8c and 3.8g. The third module is the Control module which is designed to act as master to control all the modules and provide power for itself as well as the UBar and CR modules.

3.3.2 Structure

The last category of our modular system addresses how to rapidly create physical structure. In most instances, the structure of the robot needs to be stiff and light. Reducing the weight of the structure minimizes the torque required by the motors. A stiff structure will increase accuracy of the end-effector position given local joint commands for example. However, in some cases, like dynamic legged robots, it is advantageous to have a structure that is compliant.

Fixed trusses, as shown in Figure 3.8e can be used as structural material in between modules. One of the three connection mechanisms is embedded in each fixed passive truss as described in the previous subsection, making it possible for a user to connect each one to another as well as to other modules. These trusses are made to be as strong and as stiff as possible. They are designed as a T-beam meaning it's cross section is shaped like a T. Triangles are cut out of the surfaces leaving interior diagonals, like a Pratt truss, to save weight.

Thermoplastic trusses are made out of ABS (Acrylonitrile butadiene styrene). The material is a thermoplastic with glass transition temperature of 100° . This allows the user to apply heat using a heat gun or heat stripper, thus changing the shape of the truss on the fly.

The process used to create a legged robot in which the legs can be shaped on the fly is shown in Figure 3.9. Trusses can be heated as shown in Figure 3.9a. Once the material has reached its glass transition temperature, we can bend the truss to our desired shape, Figure 3.9b. In this case, we made a C-shaped leg, reminiscent of the legs used on the RHex robot. The ModLock mechanism is embedded at each end to allow easy assembly of modules and trusses.

Fixed trusses provide us a stiff structure. The thermoplastic trusses give us the ability to create any shape on the fly. Foam trusses give us the ability to do both. Figure 3.10 shows how to build a quadruped using two-part insulation foam. First, a paper mold is folded. Then we could attach plates that have the ModLock connection embedded including hooks that the insulation foam can hold on to. We then mix and pour in the two-part solution and let it harden. Cutting paper and folding it allows us to quickly create any desired shape.

3.3.3 Connection

To enable rapid prototyping it is worth given consideration to how these actuation and structural pieces are connected. The conventional and most robust method is using screws. We use a method called the ModLock mechanism which has been described in more detail in Chapter 2 which is a robust connector that can connect modules instantly.

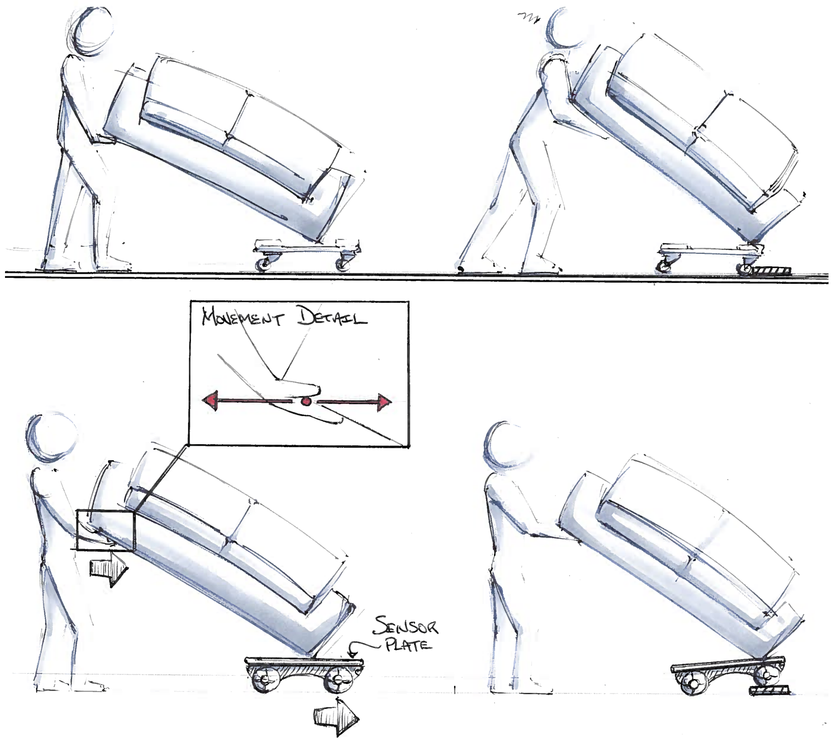
3.3.4 Wiring

In traditional chain-style modules each face has electrical connections so that the modules can be attached to each other without external wiring. By using trusses, modules are not connected face to face anymore. This requires external wiring to share communication while providing power to the modules. With the foam and thermoplastic trusses there are no fixed lengths between actuator modules. A retractable cable can cope with this variability and allows for different lengths.

3.4 Conclusion

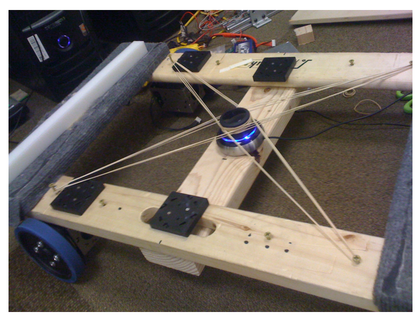
Prototyping and iterating has shown to be useful in many fields to drive innovation. Likewise, being able to quickly and cheaply build robots could prove a useful way for robotics to move the field forward. We have proposed a modular system that divides actuation, struture, wiring into separate reusable components. We proposed methods to rapidly create stiff or compliant physical structures for the robot.

Our system is a starting point for partical tools for emergencies in which a robotic solution is needed or in task explorations for robots in product design. The system has proved most promising, however, for the study of locomotion. This can be attributed, perhaps, to the limitations of our particular system, rather than a property of the modular approach or philosophy. In the planetary contingency competition, structures were scaled down to make the task achievable using our modules. For product design, mockups were used to understand the size, feasibility and to convey a story. Aesthetics of the prototypes however proved to be a deal breaker for presenting them in user studies.

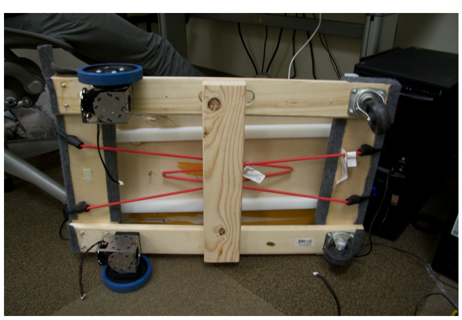


(a) Sketch

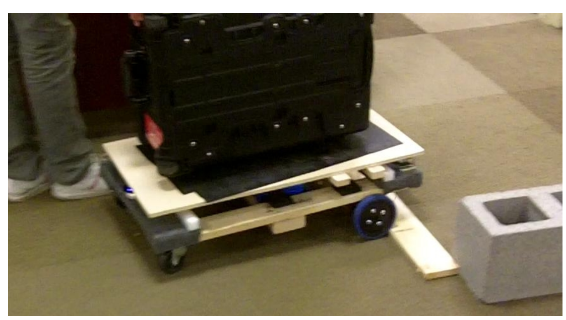
(b) Prototype isometric



(c) Prototype Spacenav

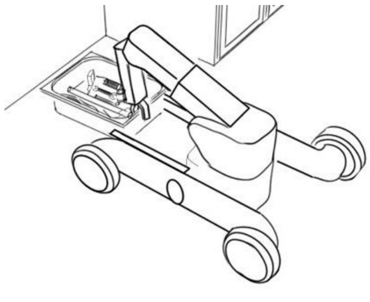


(d) Prototype bottom

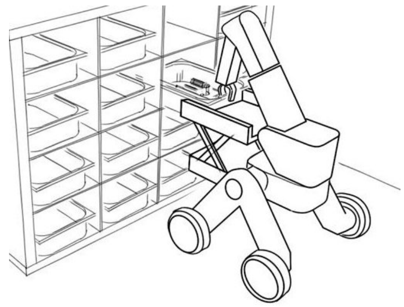


(e) Over a threshold

Figure 3.3: SuperDolly concept drawing and prototype

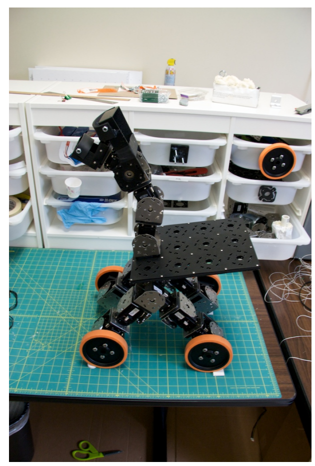


(a) Sketch flat

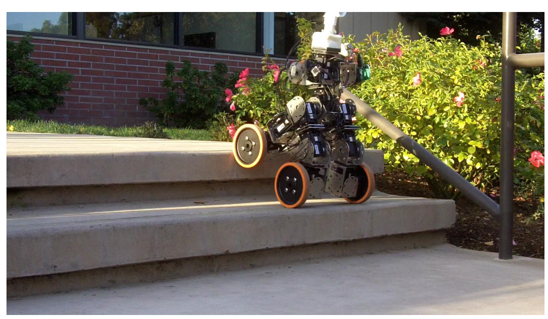


(b) Sketch extended

(c) Prototype flat

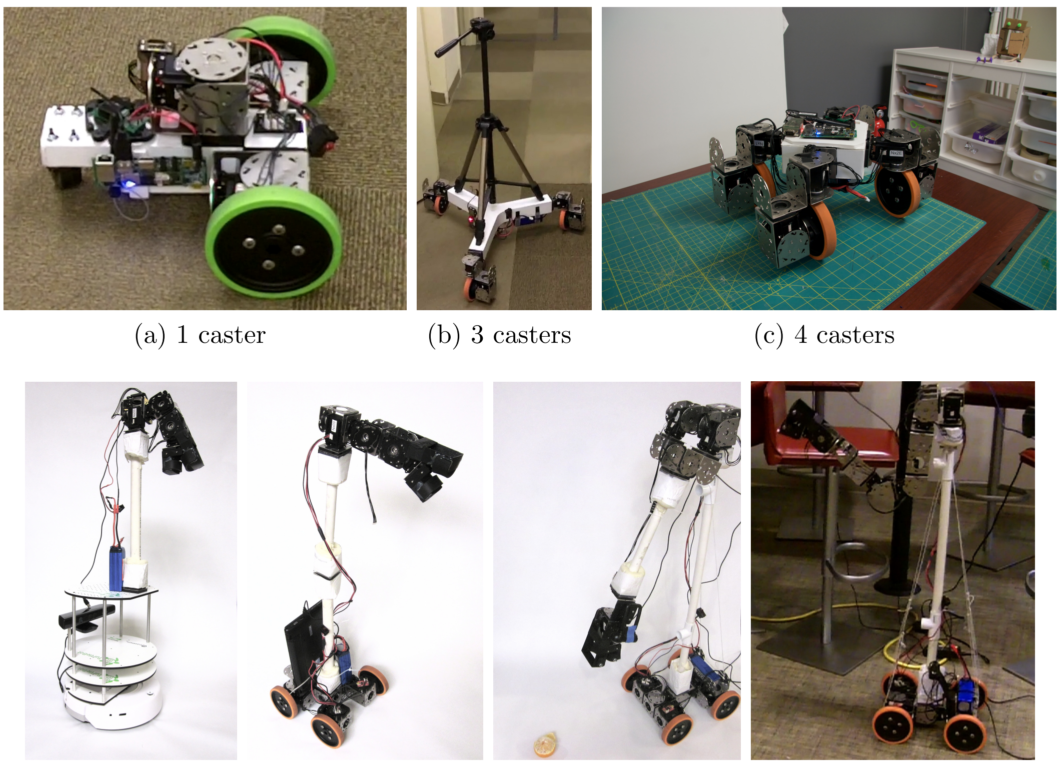


(d) Prototype extended



(e) Prototype ascending stairs

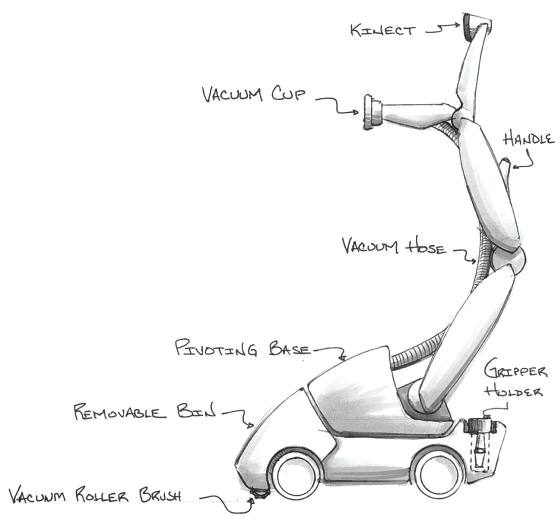
Figure 3.4: PlatformBot concept drawings and prototype



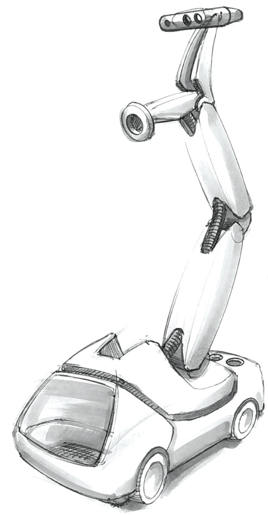
(a) 1 caster (b) 3 casters (c) 4 casters

(d) manipulator 1 (e) manipulator 2 (f) manipulator 2 (g) manipulator 2

Figure 3.5: Mobile bases and manipulators



(a) Profile Sketch

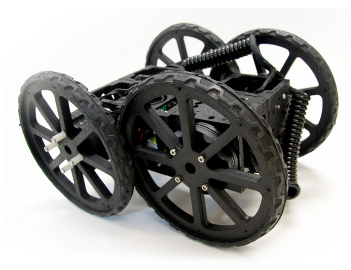


(b) Isometric Sketch

(c) Prototype 1

(d) Prototype 2

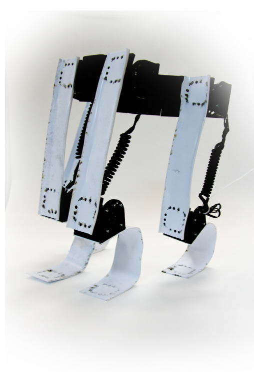
Figure 3.6: VacuumBot concept drawings and prototype



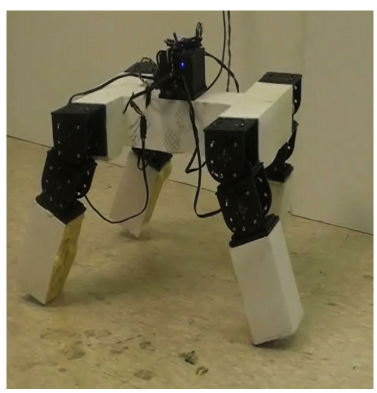
(a) Wheeled vehicle Driving

(b) Wheeled snake slithering

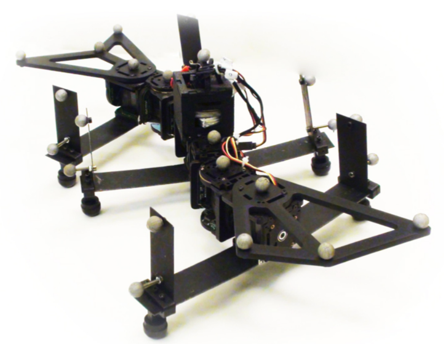
(c) Closed loop rolling



(d) Quadruped Raised 1

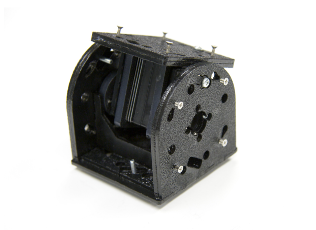


(e) Quadruped Raised 2

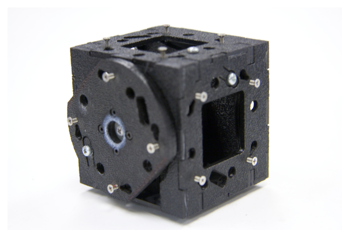


(f) Hexapod Sprawled

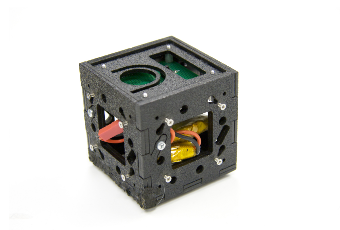
Figure 3.7: Selected configurations demonstrating locomotion principles with various generations of CKBot



(a) UBar

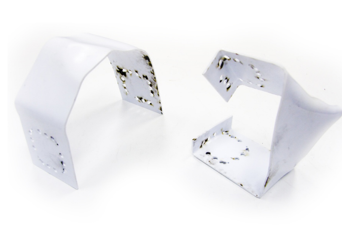


(b) CR (Continuous Rotation)

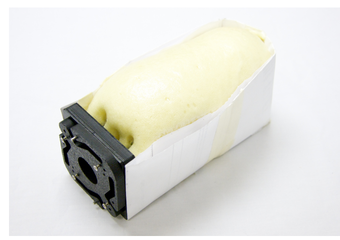


(c) Control

(d) Fixed Truss



(e) Thermoplastic Truss

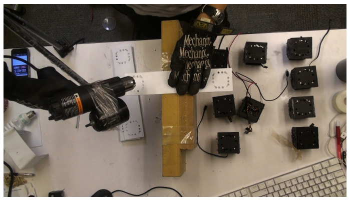


(f) Pourable Foam

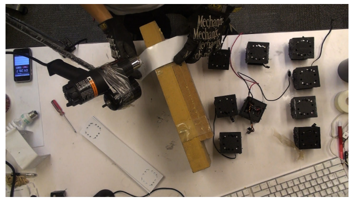
(g) Wheel

(h) Retractable Cable

Figure 3.8: CKBot V2 Modules

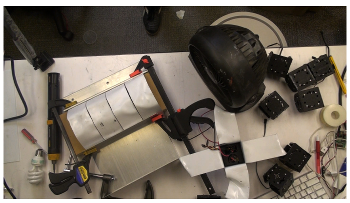


(a) Apply heat to trusses



(b) Wait till it reaches glass transition temperature, then bend

(c) Bend more legs

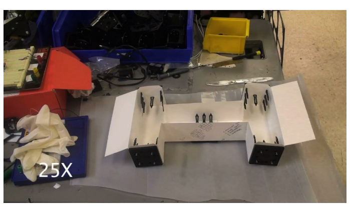


(d) Let cool

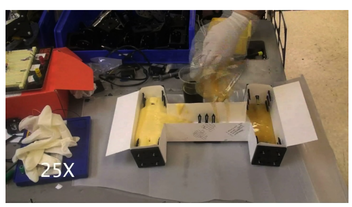
(e) attach trusses to CKBot modules

(f) Program walking gait

Figure 3.9: Creating a Quadruped robot using CKBot modules and thermoplastic trusses

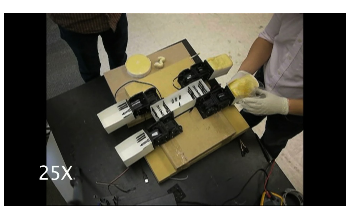


(a) Create paper mold with ModLock plates

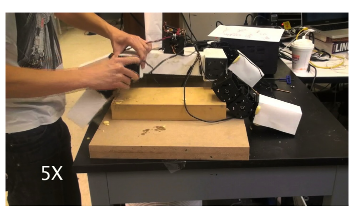


(b) Pour in 2 part solution foam. This will be the body

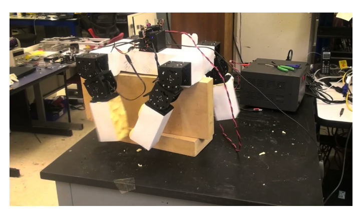
(c) Foam rises and harden



(d) Create 4 legs using similar method



(e) Attach to body and CKBot modules



(f) Program walking gait

Figure 3.10: Creating a Quadruped robot using CKBot modules and foam

Chapter 4

Development and Analysis of a Dynamic Rolling Configuration

This section will focus on a rolling loop configuration using a sensor-based feedback controller to achieve dynamic rolling. The robot senses its position relative to the ground and changes its shape as it rolls. One of the major findings is that more elongated shapes achieve higher terminal velocities than rounder shapes and at 1.6m/s this is the fastest gait yet implemented for an untethered modular robot. The control scheme is scalable to an arbitrary number of modules, shown here using 8 to 14 modules.

4.1 Introduction

This work was originally motivated by the need to find locomotion modes that would allow modular robots to cover large distances at reasonably fast speeds. We envision modular robots reconfiguring and using this mode of locomotion primarily in situations where they need to traverse a long path. While the research presented here has been applied to one particular robot, we believe that the controllers and ideas developed here can be easily extended to other rolling robots. In particular, we have demonstrated that the controllers easily scale with the number of modules on the robot. Thus, it should be easy to use the controllers for robots with different sizes and possibly very different actuation modes.

In the loop configuration shown in Figure 4.1(a), the motion of the robot is like that of a tank tread. On flat terrain this gait is currently considered to be the most efficient as well as the fastest configuration under some conditions [101] (this has not been proven for the general case though). It has been implemented on various robots in [48, 93, 101]. In most implementations, the motion of these gaits was not dynamic, *i.e.* there was no inertial component to the motion. Rather the motions were purely *kinematic*; changes in geometry cause locomotion, stopping the changes in geometry also stop locomotion. There is a limit to the rate at which this motion can accelerate and stay kinematic. Accelerating too fast causes the loop roll backwards as shown in Figure 4.1(c). Its motion stops being kinematic and causes the robot to undulate in space. This however does not produce the intended behavior of forward locomotion. In this work we aim to produce a controlled dynamic rolling mode as shown in Figure 4.1(b). These different modes of locomotion are also illustrated in Extension 1.

In [46], Kamimura et al. implemented an *open-loop* dynamic rolling gait using CPGs (Central Pattern Generator) where the weights for the CPGs were determined using simulation. A deformable robot was actuated by SMA (Shape Memory Alloy) coils in [93] to manipulate the shape into stable and unstable deformations for crawling and jumping. In [58], Matsuda and Murata proposed a robot whose links formed a closed chain where the actuators control the stiffness of a spring in each joints. This allows them to adjust the stiffness in each joint and drive the robot forward. In [90], a dynamic simulator was used to generate and simulate a dynamic rolling gait. Feedback was through accelerometers in the robot and an average velocity of about 1 m/s was reported. However, this gait was not implemented on an actual robot and no analytical insight was provided. A dynamic rolling gait was implemented on a Icosahedral modular robot called the Tetrobot in simulation in [57]. However, no implementation details were given for this work.

A loop that rolled dynamically was demonstrated by Duff and Yim at PARC using PolyBot modules, however this work was never published. The loop approximated an ellipsoid. The major and minor axis of this ellipsoid rotated in an open loop fashion, starting slowly and increasing predicting the acceleration of the robot with sensing. Under proper initial conditions the loop would accelerate. In contrast, the use of closed loop control with contact sensing in the work presented here has removed any dependence on initial conditions.

Different kinds of rolling robots have also been built and studied in recent years. Halme et.al. [40] introduced the first truly spherical robot that uses an internal steerable wheel to generate motion. Mukherjee, et.al. [62] proposed motion planning strategies for the Spherobot, a proposed variant on Halme's robot that used internal reciprocating weights. Bhattacharya et.al. [5] used the principle of conservation of angular momentum to generate motion for a spherical robot. They also presented analytical results for control of the robot using an optimal control approach. Shores, et.al. [92] propose a bipedal robot that can also locomote by folding into a circular shape and using internal joint motions to initiate rolling through small shifts of the center of gravity of the robot. The Hexaball project [68]proposes to build a spherical robot with legs that can climb over obstacles and roll by using small motions of its legs.

In this work, we present a new implementation of the rolling loop that is dynamic using sensor-based feedback. Our work differs from previous work in the use of sensory feedback, development of a simplified dynamic model that provides considerable insight for development of control and robust implementation on a prototype robot. Sensory feedback dramatically improves the reliability of this gait (as compared to open loop implementations). In addition, this work presents the fastest gait yet reported by a modular robot.

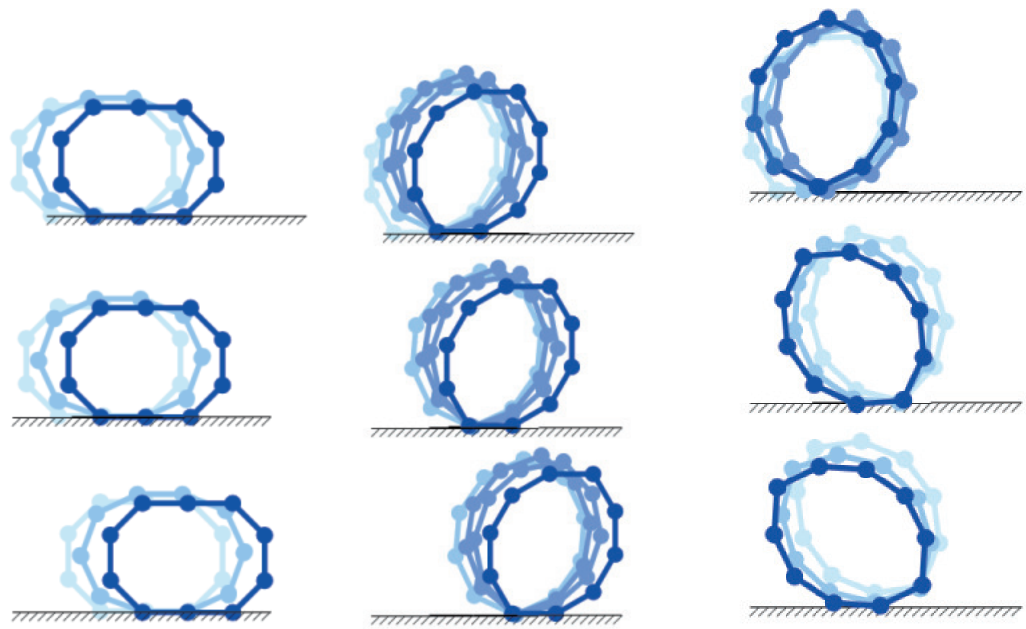
The rolling loop is formed by a closed kinematic chain with many degrees of freedom. A complete model with all the joint degrees of freedom and the closed chain constraint for an arbitrary shape could be built for our robot, but the equations of motion are very complex and would not provide much insight into the dynamics of the system. Further, with so many actuators on the robot, the dimension of the space of possible inputs makes designing controllers non-trivial. Our approach is to simplify the model for the system by restricting the type of controller to track an appropriate shape at *touchdown*, the contact of a module with the ground.

The resulting relatively simple controller gives us better insights into the dynamics of the system. Another benefit of this implementation is that the method scales to any number of modules or joints in a loop, within actuator limitations. In addition to simplifying the control algorithm, our approach also offers better insight into the dynamics of the system.

This paper is structured as follows. In Section 4.2, we present the robot used in this work. In Section 4.3, we introduce the main idea behind *dynamic rolling* and compare it with *kinematic rolling*. In Section 4.4 we present a four-bar like model for the robot that simplifies the analysis of dynamic rolling. In Section 4.5, we propose the framework used for control. The experimental setup is described in Section 4.6. In Section 4.7, we present theoretical results derived using this model and experimental results with 10 to 14 module rolling module loops. In Sections 4.8 we follow up with a discussion on insight gained from the results.

4.2 The Robot Configuration

The configuration used in this work is shown in Figure 4.2. To form a loop, each module is attached end-to-end using screws and the two ends are then screwed together to form a loop. It is possible to form loops in other ways. For example,



(a) Kinematic Rolling (b) Dynamic Rolling: Ideal Case (c) Dynamic Rolling: Loop turns back on itself

Figure 4.1: Different modes of rolling.

instead of daisy-chaining head to tail each module: (head-tail)(head-tail)(head-tail) the modules could be attached head-head: (head-tail)(tail-head)(head-tail). This is the configuration that Superbot [16] and MTRAN [48] use. When tested with CKBot, this configuration does not do as well in taking advantage of the rounded structure of the modules and thus the motion is not as smooth. As a result this was likely to be less efficient and was not tested extensively.

A separate microcontroller board, the **brain**, serves as a centralized controller. It plugs into one of the ports on the robot and provides control position commands for all modules. The touch sensors are infra-red proximity sensors that measure reflectance as an indication of distance to the ground. These touch sensors plug inside the module as shown in Figure 4.3. Sensors use empirically derived thresholds

Table 4.1: Technical specifications for a CKBot module.

Property	**Value**
Mass (per module)	138(g)
Size (per module)	W60xL60xH60(mm)
Batteries	Lithium Polymer 7.4V
MCU	PIC18f2580
Servo	Airtronics 94359
Torque	1.4Nm
Reconfiguration	Manual

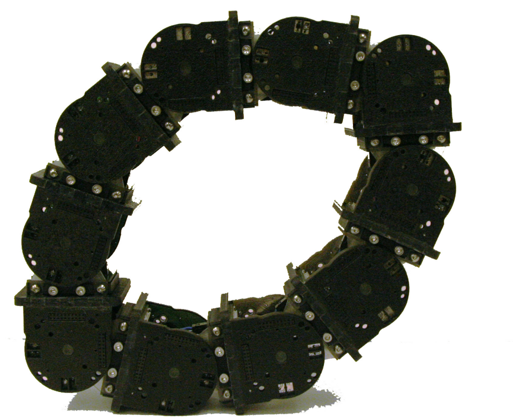


Figure 4.2: Ten CKBot modules forming a football shape.

for different surfaces to determine whether the module they are plugged into is touching down or not. The touch sensors send process messages to the brain upon a touchdown event. The brain then calculates the angles required for each module to track the desired shape and sends these commands to the microcontroller on each module.

While each module is capable of carrying a battery, typically five lithium polymer battery packs were attached to a full loop during testing which would give several hours of run time. If long life performance were required, more batteries (up to 20) could be added to the system.

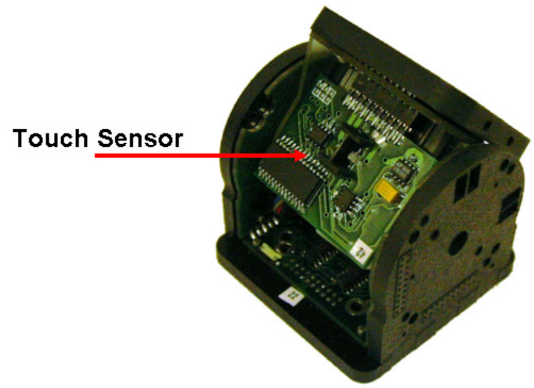


Figure 4.3: An individual CKBot module.

4.3 Kinematic vs. Dynamic Rolling

Statically stable locomotion is a term that is often used to characterize robot gaits. At any moment in a statically stable gait, the robot could stop moving its joints and the robot would not fall over. The projection of the center of gravity is always maintained to be within the convex hull of the ground contact points. Dynamic locomotion characterizes robot gaits in which the inertia of the robot plays an important role in the locomotion. In general, gaits (which are assumed to be stable) are either statically stable or dynamically stable, but not both. Traditionally static and dynamic stability refers to legged robot gaits. When applied to rolling gaits things become less clear. An automobile has its center of gravity always within the convex hull of its four tires. If it moves slow enough, the inertia of the vehicle can be ignored and it might be said that the vehicle is statically stable. However, if it gained any significant speed, the inertia cannot be ignored and the vehicle might be said to be dynamically stable. The line delineating the two conditions is not clear. In the case of loop robots, we refer to the gaits in which inertia plays no role as kinematic rolling. Here, the equations of motion can be determined directly from the geometry (no mass terms).

Table 4.2: Gait table for kinematic gait (all angles are specified in degrees).

60	60	60	0	0	60	60	60	0	0
60	60	0	0	60	60	60	0	0	60
60	0	0	60	60	60	0	0	60	60
0	0	60	60	60	0	0	60	60	60
0	60	60	60	0	0	60	60	60	0

4.3.1 Kinematic Rolling

A kinematic rolling gait is implemented by repeatedly moving the shape of the loop such that the long axis rotates. This motion is similar to the motion of a tank tread. One rotation of the long axis corresponds to one cycle of the gait. The frequency of rotation is directly proportional to the speed, *i.e.* stopping the tread causes the whole robot to stop. For a closed loop robot like the one used in this work, one typical loop shape has two lines of modules one on top of the other attached by an intermediate set of modules forming arcs as shown in Figure 4.1(a). A kinematic roll for this configuration is executed by smoothly interpolating the joint angle of each module to the joint angle of the neighboring one in the loop. This type of motion can be easily represented using a *gait table* [101].

An example gait table for a kinematic rolling gait for a 10 module loop robot is shown in Table 4.2. The neighboring columns of the table correspond to neighboring modules in the loop. The rows of the table correspond to steps (or time). The elements of the table are the joint angles for the corresponding module at the corresponding time. Note that there are only five rows in the gait table since the gait cycles back to the first row after the fifth.

One thing to note is that between rows only four modules change joint angles. This table can be scaled to larger numbers of modules by increasing the number of modules with 0 degrees (the straight parts). As the numbers get larger there would still be only four modules which change joint angles.

4.3.2 Dynamic Rolling

Unlike the kinematic gait a dynamic gait continues to move the robot even after all joints have stopped moving, *i.e.* a dynamic gait utilizes momentum. To create a dynamic rolling motion for a modular loop robot, one approach is to move the center of mass beyond the pivot point for the module currently on the ground as shown in Figure 4.1(b) and Figure 4.5. This results in a moment contribution from the weight of the robot in the direction of rolling and the robot accelerates in that direction.

The motion of the robot can be separated into two phases: (1) a shape change where the robot changes shape to the new desired shape that increases the distance between the center of mass and the ground contact point and (2) a falling phase where the robot's shape is frozen and the robot behaves essentially like an inverted pendulum pivoting about the contact point (bringing the center of mass closer to the ground contact point). The start of the first phase occurs as soon as a new touchdown is detected. This paper will show that the first phase results in a slight deceleration and then an acceleration while in the second phase the robot is continuously accelerating towards the next touchdown.

This motion is clearly not statically stable as the center of gravity is never within the convex hull of the ground contact points. One could say the robot is continuously falling. If the joints in the loop were to lock in place the robot would not instantaneously stop but would rather continue falling. Since the robot is shaped like a loop, as long as it falls in the plane formed by the loop, it is never in a position where it cannot move (i.e. the way a legged robot may catastrophically fail if it falls over). One way to view this method of control is that of a modified gait control table where the speed of motion between rows of a gait table is based on sensor feedback.

4.4 Analytical model

We make a simplifying assumption that serves to reduce the complexity of the dynamic analysis for the robot. We choose a "backbone" curve [21] to which we map the modules. We restrict this backbone curve to a shape that is formed by joining two equal arcs of a circle whose sector subtends an angle less than 180 degrees. This results in a shape which resembles an American football as shown in Figure 4.2. In the limit, as the two arcs approach 180, the shape reduces to a circle. The modules of the robot approximate this backbone curve by fitting the position of the joints to lie on the arcs.

The shape can be defined using a single parameter θ_a, the angle between the modules at the top and bottom apex of the shape (Figure 4.4). All the other joint angles are equal to each other (to say θ_s) and can be derived in terms of θ_a from Equation 4.1

$$2\theta_a + (n-2)\theta_s = 2\pi, \tag{4.1}$$

where n is the number of modules in the loop.

It will be clear from our choice of control strategy in Section 4.5 that local shape changes of the robot will involve only four modules moving at a time just as the kinematic gaits in Section 4.3.1. We can thus simplify the model of the loop to that of a floating four-bar mechanism hinged at the contact point during the entire motion of one touchdown to the next touchdown. In this model the four moving modules represent the four joints. The two longer arcs (nodes 2 through 5 and nodes 7 through 10) represent two of the links of the four-bar while the other two links (comprising node 1 and 6) are made up of single modules. Reducing the model in this manner to the one degree of freedom four bar linkage means that the shape of the robot can be parameterized using a single parameter, θ_a (or similarly θ_s). This framework is shown in Figure 4.4.

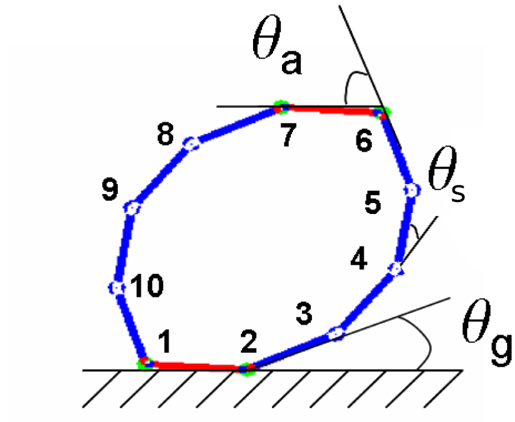


Figure 4.4: Four-bar model used for analysis. The two longer arcs (nodes 2 through 5 and nodes 7 through 10) freeze their joints and can be considered rigid. The thus represent two of the links of the four-bar while the other two links (comprising node 1 and 6) are made up of single modules.

The equations of motion for this simplified version of the robot are derived using a standard method by first defining the Lagrangian for the system and deriving Lagrange's equations. The generalized coordinates used in the analysis are the apex angle θ_a and the global angle made by the robot with the ground θ_g (Figure 4.4). Each module is considered to be a thin rod of length 0.06 m with mass 0.138 kg (from Table 4.1). The resultant equation for the evolution of θ_g can be expressed in the form:

$$\ddot{\theta}_g = f_1(\theta_a, \theta_g)mg + f_2(\theta_a, \theta_g)\tau. \tag{4.2}$$

where m is the mass of a module and g is gravity. Note that the first term on the right hand side essentially collects the terms that are linear in mg while the second term on the right hand side collects all the terms linear in τ. f_1 and f_2 are functions of the two angles θ_a and θ_g and constant parameters including the length of the module l, its mass m and the mass moment of inertia (I_0) of the module about its rotational degree of freedom. f_1 and f_2 are presented in detail in the Appendix.

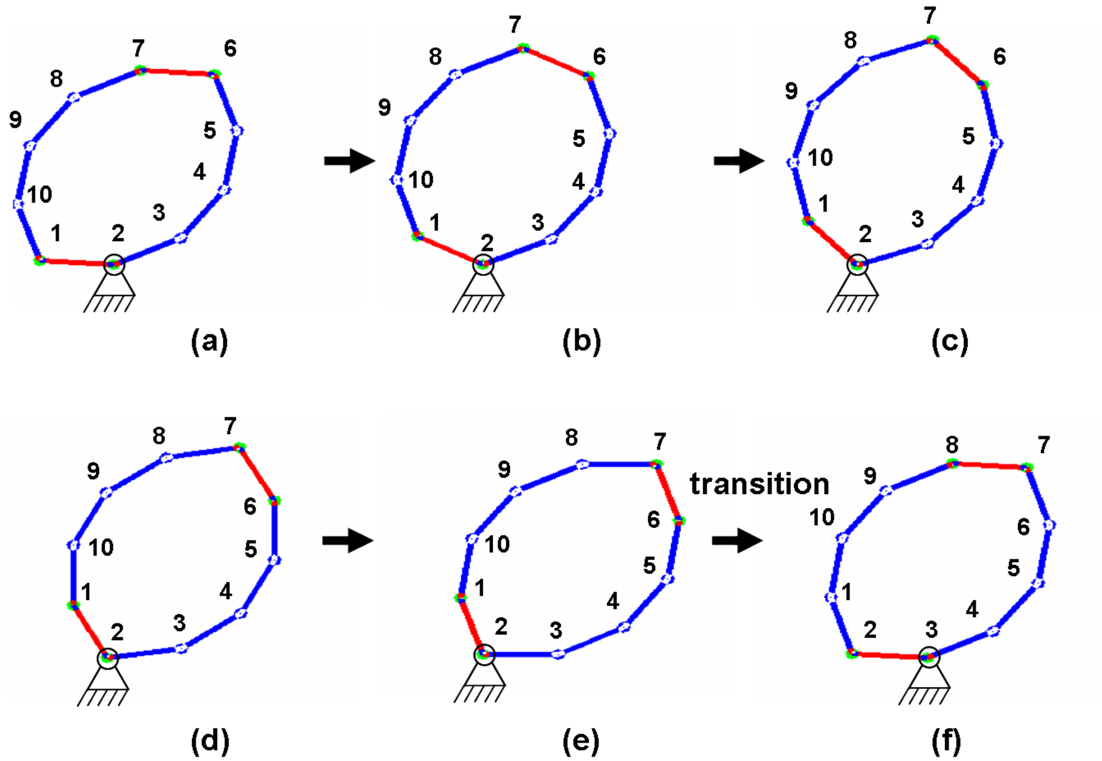


Figure 4.5: Different phases of the rolling motion illustrating the effect of shape change at touchdown and subsequent falling motion of the robot. Modules change shape in (a) - (e) note apex nodes 1,6 change to 2,7 and start to pivot rigidly about node 3 (f).

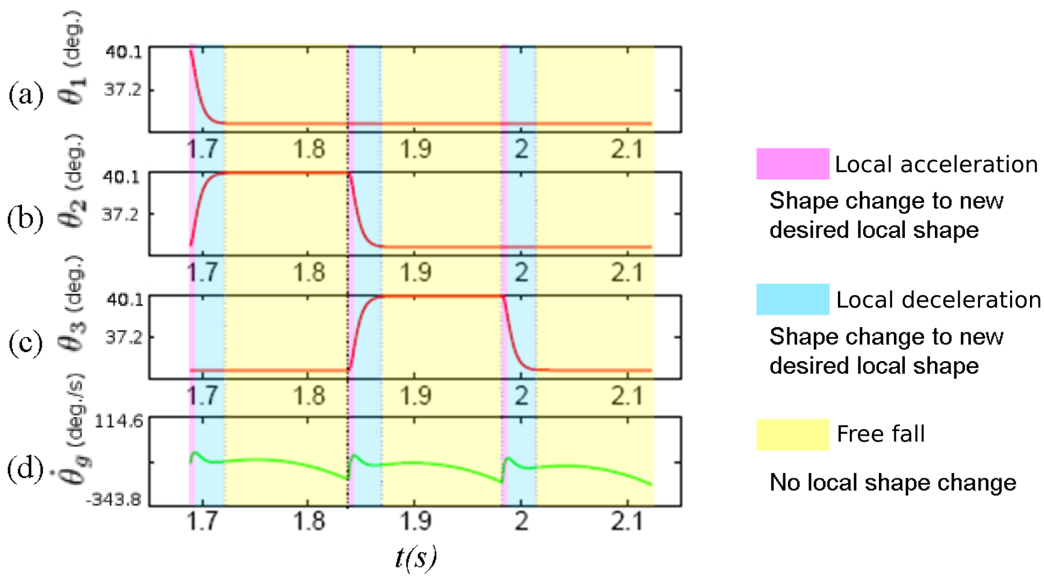


Figure 4.6: Simulation results: (a) Joint angle for Module 1 (b) Joint angle for Module 2 (c) Joint angle for Module 3 (d) Angular velocity of the robot over time interval corresponding to three consecutive module touchdowns.

Equation 4.2 shows that there are two contributing terms to the angular acceleration of the robot: (1) the moment due to gravity about the point of contact with the ground and (2) a coupling term arising from the coupling of θ_a and θ_g. While the moment arm due to gravity is always towards the direction of rolling, the direction of the coupling term is initially against the direction of rolling. Thus, the robot first decelerates during the beginning of the shape change phase (as shown by the local deceleration part of the graphs in Figure 4.6) and then accelerates due to a change in sign of the coupling term in the remaining part of the shape change phase (shown in local accelerasion phase in Figure 4.6). The coupling term is initially against the direction of rolling since it is applying torque at the appropriate modules to set the shape change in motion. It then reverses sign to slow down this motion to lock its joints in place before the free fall phase. After finishing the shape change phase, it then continuously accelerates in the free-fall phase (shown in free fall phase in Figure 4.6) solely due to the influence of the moment-arm due to gravity. As the robot rolls faster, the duration of the free-fall phase gets shorter. Beyond a certain speed, it is possible that the robot is unable to go through its complete shape change before touchdown in which case the simplified four-bar model we use in our work is no longer valid. This corresponds to touchdown happening at a time before the free fall region in Figure 4.6. Since our model is no longer valid in these cases, we do not report or use these results for further analysis.

We define a *step* of the gait as the sequence of events between consecutive touchdowns of two adjacent modules. At touchdown, we reassign the nodes to the different links based on the global positions of the nodes. This is illustrated in Figure 4.5. In Figure 4.5(a) joints at 1, 2, 6 and 7 form the joints of the four-bar linkage. After the transition to Figure 4.5(f) the four bar is represented by the joints 2, 3, 7 and 8 and joint 3 becomes the pivot point around which the fourbar linkage is hinged.

When the module comes into contact with the ground, a transition condition is defined at impact of the module on the ground. Joint angles and the position of the robot stay fixed at transition while velocities are transformed using the transition conditions. The transition condition relates the angular momentum L_- of the whole body of the robot about the new pivot point on the ground just before impact with the angular momentum L_+ of the whole body about the new pivot point after impact. Using a coefficient of restitution η (found empirically to be 0.94 on carpet flooring), the transition condition is given by the momentum transfer equation 4.3 on impact.

$$L_+ = \eta L_-. \tag{4.3}$$

Thus, at each step energy is lost with each impact based on η. Also at each step energy is input to the system by the motors as the loop changes shape. The energy input, to a first order, is constant with each step, however, the energy lost is a function of velocity (as a component of momentum). So, it is logical to propose that as the system accelerates from zero velocity, a terminal velocity will be reached where the energy input to the system is equal to the energy lost, assuming a stable steady state.

4.4.1 Scalability

The particular choice of parameterization made for the controller earlier in Section 4.5 has the advantage of making the controller *easily scaleable* to configurations with a different number of modules. This has an advantage in designing controllers for modular robots since it reduces design and computational requirements for control and makes the controller invariant to the number of modules in the loop. Consider, for example, Figure 4.7 where a loop robot with n modules is shown. In Figure 4.7(a), the apex nodes are 1 and m where $m = 1+n/2$. The link joining nodes 1 and 2 (Link 1) and the link joining nodes m and $m+1$ (Link 2) form two links of the four-bar used for analysis and control. The third link (Link 3) is formed by a combination of the links joining joints 2 through m and the fourth link (Link 4) is formed by a combination of the links joining joints $m+1$ through n and 1. Thus, a multi-degree of freedom rolling loop with n modules can be reduced to the same four-bar linkage used for analysis.

Note that all the joints in Link 3 and Link 4 are stationary during the shape change phase and the only joints that move are the 4 joints that attach Link 1 and Link 2 to Link 3 and Link 4. The control scheme relies on position control of the servos to maintain the shape of Links 3 and 4, even though they are not moving there is some power consumed to maintain this shape. As the number of modules in these links get larger the weight of the modules will cause larger draws on power, even exceeding the torque limits of the actuators. One interesting property is that at higher speeds, centrifugal forces may reduce torque requirements saving energy. In the limit, as a shape approximates a circle, gravity and the ground reaction forces will apply vertical compressive forces. Counteracting these forces, centrifugal forces apply radially outward.

The scalability of the controller to different number of modules was tested by implementing the controller on rolling loop configurations with 8, 10, 12 and 14 modules.

4.5 Control

In Section 4.4, we made a simplifying assumption that allowed parameterization of the desired shape at touchdown using a single parameter, the apex angle θ_a. The controller used for dynamic rolling can now be described by specifying a new desired shape for the robot at touchdown such that the robot is falling forwards with respect to the pivot point describing the contact of the robot with the ground. This corresponds to designating node 7 and 2 in Figure 4.5(a) as the new apex angles of the shape. When a new desired shape is specified the loop changes shape as is illustrated in Figure 4.5(b)-(c)-(d). Once it reaches the new desired position,

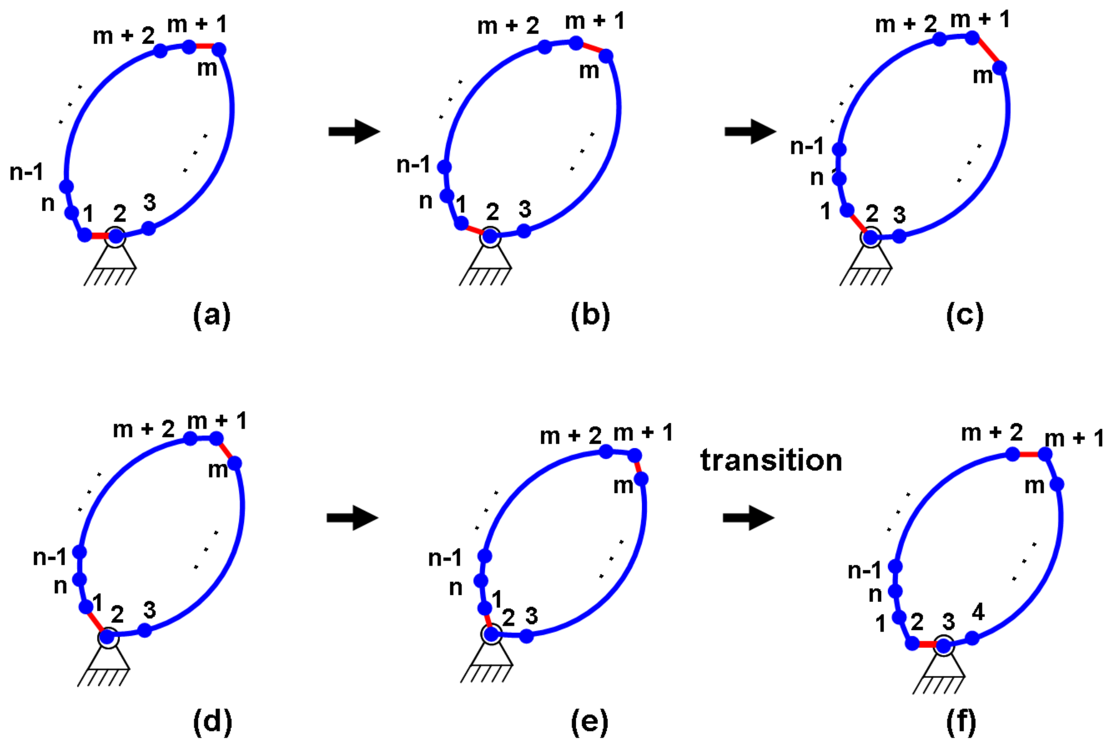
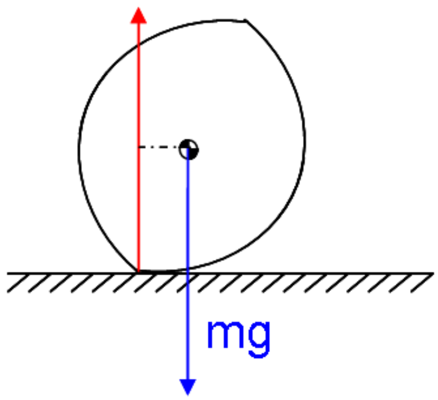
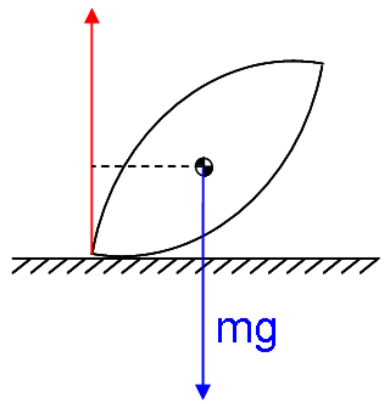


Figure 4.7: Scalability of the controller to different number of modules.



(a) Moment arm corresponding to a rounder shape at touchdown.



(b) Moment arm corresponding to a more elongated shape at touchdown.

Figure 4.8: Effect of shape on moment arm at touchdown.

the local shape does not deform anymore. Now, the robot undergoes a pure falling motion (Figure 4.5(d)). The robot falls like an inverted pendulum until node 3 touches down on the ground (Figure 4.5(e)).

Shapes that are more elongated (corresponding to higher values of θ_a) will result in a larger moment arm and higher angular acceleration. However, the amount of shape change (represented by the net change of joint angles) is also higher. Rounder shapes correspond to a smaller value of θ_a and will result in a smaller moment arm and smaller amount of shape change. Figure 4.8 shows the effect of the choice of shape on the moment arm due to gravity. Figure 4.8a shows the smaller moment arm corresponding to a rounder shape and Figure 4.8b shows the larger moment arm corresponding to a more elongated shape. We should expect that more elongated shapes will give us higher accelerations while rounder shapes may be more efficient. We will examine the effect of the desired shape on the speed of the robot by varying the parameter(θ_a).

This shape control is implemented by the brain board sending the angular positions to corresponding modules over the RoboticsBus at 60 Hz. Each microcontroller on each module generate PWM signals to the servos which then use a highly tuned PID position control to maintain or attain the commanded position.

4.6 Experiments

4.6.1 Terminal velocity and specific resistance

One of the main objectives for the experiments is to see if the trends proposed from the analysis of the model hold true, namely

1. The robot achieves a terminal speed during rolling and this speed increases with increase in θ_a.

2. Rounder shapes are more efficient.

Reflective markers were placed on the robot to track a single module and its joint angle by a high speed motion capture system (VICON). Measurements were recorded at a speed of 100 Hz and a resolution of 0.1 mm. The overall workspace of the VICON however was limited to 3 m $\times$ 3 m. To determine the terminal velocity the position of the robot was measured manually from video to increase the available workspace. Each trial to determine the terminal velocity consisted of two parts. In the first part a running start of 4 m was given to the robot to allow it to reach terminal velocity. No measurements were taken in this part. In the second part, the robot would continue rolling and position was determined manually from the video footage by marking time stamps as the robot crosses markings on the carpet spaced at 1 foot intervals. The field of view of the camera covered only the second part of the trial. No significant accleration was seen in this second part.

The desired shape of the robot was specified using the parameter θ_a for a 10 module robot. θ_a was varied between 36^0 to 90^0. $\theta_a = 36^0$ represents a shape where $\theta_a = \theta_s$ and there is no change in the shape of the robot while $\theta_a = \pi/2$ represents an elongated shape where the amount of shape change in the robot between touchdowns will be very high. It was found that shapes with (θ_a) greater than 70^o could not be tracked accurately by the controller. However, results for these values are still reported here.

Specific resistance (ϵ) measures the energy cost of locomotion per unit distance and robot weight and is thus a good measure of efficiency [97]. It is calculated as follows:

$$\epsilon = \frac{P}{mgv} \tag{4.4}$$

where P is the average power input to the robot, m is the total mass of the robot, g is the acceleration due to gravity and v is the average speed of the robot. Specific resistance is a natural measure for the second claim above, *i.e.* that rounder shapes are more efficient by consuming less energy per unit distance.

Experimentally measuring specific resistance requires the measurement of the power consumed by the robot and the average speed achieved by the robot over the corresponding run. A robot with 10 modules and 5 lithium polymer batteries has a mass of 1.7 kg. Normalizing the power consumption in this manner with respect to both the speed and mass of the robot allows meaningful comparison between robots of different sizes and speeds.

4.6.2 Motion on inclines

Our initial studies showed that the robot works well on level terrain, but for this gait to be really useful in space exploration, search and rescue or any real world scenario we wish to show that it behaves well on non-flat terrain as well. Examining traversal on an inclined terrain is a step towards more unstructured terrain. It should be obvious that rolling down an incline is possible (e.g. just by maintaining a circular shape) however, traversing up is not as clear. Experiments were performed going up a slope on an incline of 5 degrees and down a sleep with an incline of -5 degrees. This incline angle was chosen due its prevalence as the requirement for wheelchair access. Rolling motion up a long incline is a good measure of the robustness of the controller to a constant source of disturbance while rolling down an incline tests the controller's ability to react to faster touchdown events.

Terminal velocities and power consumption were measured and compared with behavior on level terrain. Multiple trials were carried out for each shape and incline on the same carpet to maintain consistency across trials.

4.6.3 Scalability of the controller

As noted earlier, the particular parameterization chosen for the rolling loop makes it easier to scale the controller to loops with different number of modules. This was tested by implementing the controller on rolling loop configurations with 8, 10, 12 and 14 modules. The controller maps the links and joints of all these configurations onto the four-bar like *backbone curve* used earlier for analysis. The user chooses the value of θ_a and θ_s can be easily determined from θ_a using Equation 4.1. The controller then designates the module touching the ground and the one diameterically opposite as the apex and sets all other joint angles to a constant value of θ_s. The control algorithm for n modules is thus the same as the one used for the loop with 10 modules. This demonstrates the versatility and scalability of the controller.

4.6.4 Speed control

Results from the terminal velocity experiments showed that more acceleration occurs in the falling phase at greater apex angles. This led to experiments to demonstrate arbitary speed control with a human specifying the desired speed using a joystick in real time. The robot could be sped up by increasing the apex angle specified by the controller. Braking motion to slow the robot was achieved by designating the module in front of the current touchdown module as the apex of the new desired shape. Snapshots are shown in Figure 4.15 and in Extension 2.

4.6.5 Experimental setup

Experiments were carried out on multiple surfaces, but the results reported here are for carpet flooring. The choice of surface on which the robot rolls has a visible effect on the speed of the robot. The robot was slower on thick carpet than on a thin carpet placed on a marble floor where the fastest run times were achieved. The choice of flooring also affects the performance of the IR touch sensors. Thresholds for the sensors were set manually on different floor surfaces to achieve the best performance.

Ground truth data was provided by the high speed motion capture system (VICON). The VICON motion capture system provides measurement of pose of one of the modules and one joint angle of the robot at a high speed (100 Hz) and sub-millimeter accuracy. This allowed comparison between the actual and desired trajectories of the joints on the robot and let us verify that the controller triggers the correct module on touchdown at the correct time.

4.7 Results

4.7.1 Tracking of desired joint trajectories

Figure 4.9 plots experimental tracking results for one of the joint angles of the robot and also the global position of the robot. Figure 4.9(a) shows the height of one of the modules and it should be noted that the crests in the z positions in Figure 4.9 represent touchdowns for the module diametrically opposite the tracked module while the troughs represent touchdowns for the tracked modules themselves. The joint angle of this module is shown in the middle figure and we can verify that the module reaches $\theta_a = \pi/4$ and goes back to $\pi/6$ and that this motion is triggered upon touchdown of the module *i.e.* when the z position is at a minimum. Note that the duration where the module holds the apex angle θ_a is very short.

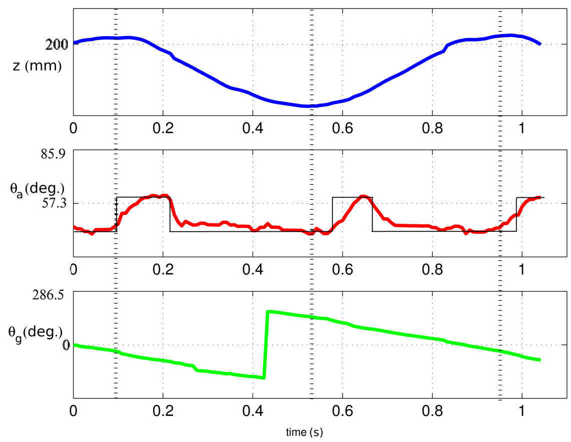


Figure 4.9: Experimental results: Tracking results for one module using a motion capture system: z represents height of tracked module above the ground for one cycle, α represents joint angle of tracked module, α_g represents pitch of tracked module with respect to global reference frame (the discontinuity in the data is because of a jump from $-\pi$ to π).

4.7.2 Terminal velocity and specific resistance

Figure 4.10 plots simulation and experimental results for the final speed of the robot for different desired shapes at touchdown. As predicted in the analysis of the model, the observed behavior of the system was that a terminal velocity was reached. In addition, as the desired shape becomes more and more elongated (corresponding to increase in the value of θ_a), the terminal velocity achieved by the robot increases. Also as the desired shape grows elongated, the angular acceleration of the robot in its free fall phase also increases thus resulting in a higher terminal speed. Shapes with an apex angle greater than 70^o cannot keep up with the speed because the servo cannot move fast enough to reach the next shape before the next touchdown.

Below a certain magnitude of shape change, the robot has zero terminal velocity. In other words, even when given an initial velocity there exists acertain $\theta_a < \theta_{critical}$ at which point the robot will eventually roll down to a stop. Where $\theta_{critical}$ is defined as the apex angle with which the robot has just enough energy to continue motion and

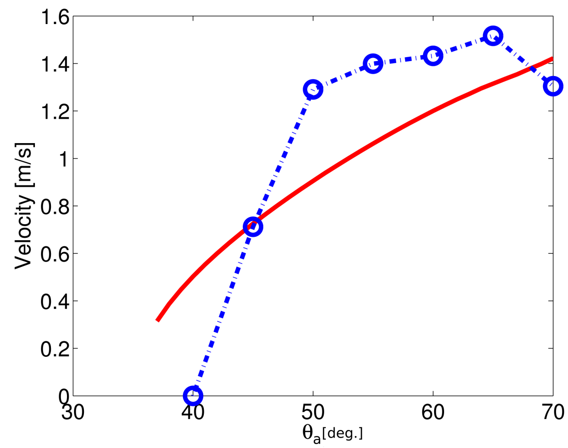


Figure 4.10: Terminal velocity vs. θ_a. Simulation shown by a solid line and experimental results by a dotted line (with (o)'s).

continue rolling after touchdown rather than rolling back to the previous touchdown. Note in the continuous case (with infinite modules) $\theta_{critical}$ approaches 0 at which point it is a perfect circle. Geometric observation shows that when $\theta_a = 37^o$, the center of gravity sits over the new touchdown point. In simulation, with values of θ_a less than 37^o, the robot slows down to a halt even if it has some initial momentum. For $\theta_a \geq 37^o$, the robot is able to sustain its momentum in simulation and roll continuously. Experimentally, the robot does not achieve continuous motion unless $\theta_a > 40^o$. The experimental terminal velocities are close to the predicted velocities.

Figure 4.11 plots simulation and experimental values for the specific resistance for different desired shapes. The power determined analytically should be lower than the actual power input to the robot, because the simulation only takes into account the power used to change shape. This is shown in our results. The experimental measurements show larger specific resistance than the theoretical measurements in all trials. More importantly the trend stays the same, *i.e.* rounder shapes exhibit lower specific resistance and are more efficient.

Another estimate of energy efficiency of a gait is the amount of travel in joint space that each module must move in order to move forward. This is measured by the difference between the two angles $\theta_a - \theta_s$. By this measure, rounder shapes also use less energy than the more elongated ones as $\theta_a - \theta_s$ is smaller. It is worthwhile noting that, based on this measure, dynamic gaits with rounder shapes are also more efficient than kinematic gaits. Maintaining any velocity using a purely kinematic gait typically requires a large traversal of modules in joint space while, once some speed has been built up, dynamic gaits can be sustained using smaller effort in joint space.

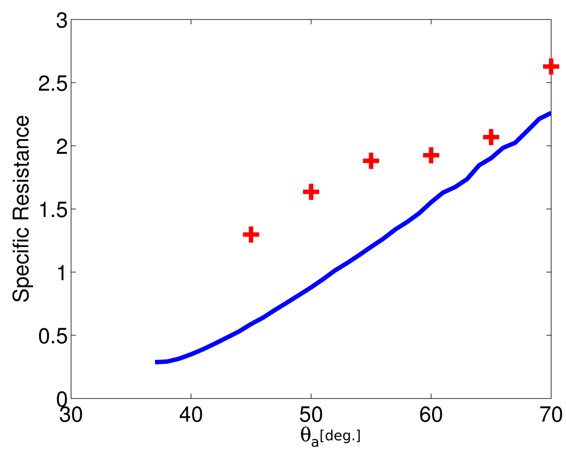


Figure 4.11: Specific resistance vs. θ_a. Simulation results shown with a solid line and experimental results with a '+'.

To compare these numbers with those for a kinematic gait, specific resistances for different dynamic rolling gaits as well as kinematic rolling gaits are plotted against terminal velocity as shown in Figure 4.12. The kinematic rolling gait has higher specific resistance than all the dynamic rolling gaits which implies that the amount of energy used to move a unit distance is lower in dynamic rolling than in kinematic rolling, which is what one would expect. For completeness, the electrical power consumed by the total robot is presented in Table 4.3 for the kinematic as well as the dynamic gait. This gives an indication of the absolute power consumed by the robot over time. However, specific resistance represents a normalized non-dimensional measure of efficiency that can be better used to compare the performance of this robot with other similar locomotion systems.

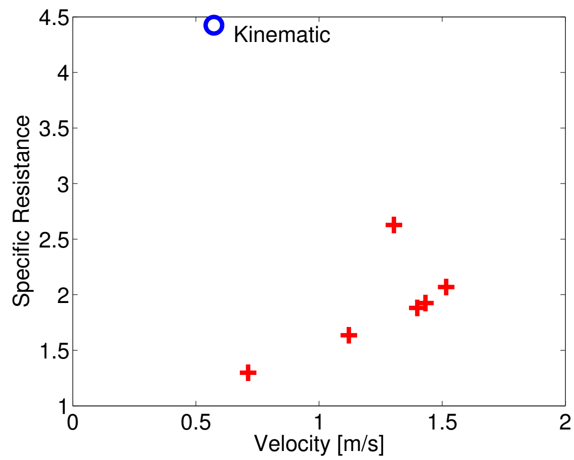


Figure 4.12: Experimental results: specific resistance vs. velocity. Dynamic rolling (shown with '+') and kinematic rolling (shown with 'o').

45	1.541
50	3.057
55	4.389
60	4.597
65	5.231
70	5.715
Kinematic	4.232

Table 4.3: Apex Angle (in degrees) vs. Power (in Watts)

4.7.3 Motion on inclines

Figure 4.13 summarizes the terminal velocities of the robot in a dynamic rolling gait on different inclines (-5^0, 0^0, and 5^0). On each incline, values of θ_a between 36^0 to 70^0 were used. Video of the incline experiments can be found at Extension 4.

In the case of a downward slope, rolling motion with a terminal velocity of 0.9 m/s was achieved even for $\theta_a = 36^0$ while on level terrain no motion was achieved for $\theta_a <= 40$ degrees. On the upward slope no motion was achieved for $\theta_a <= 50^0$. The trend of terminal velocity increasing with more elongated shapes is preserved on all the inclines. The terminal velocity also saturates at a lower value for higher slopes of the terrain.

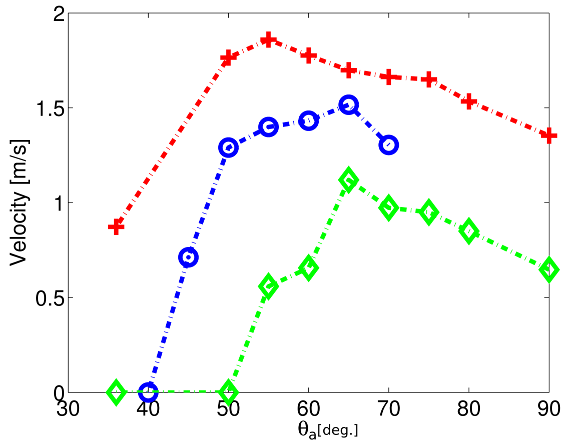


Figure 4.13: Experimental results: Terminal Velocity (in m/s) vs. Apex Angle (in degrees) on 5^o (shown with '+'), 0^o (shown with 'o') and -5^o (shown with '$\diamond$') inclines.

4.7.4 Scalability

Figure 4.14 shows that terminal velocity increases logarithmically with an increase of 3.7 times between 8 and 10 modules. There is only a small increase found between 10 and 12 modules and no significant difference between 12 and 14 modules. The terminal velocity saturates and approaches a limit at 1.6 m/s. Video of the experiments can be found at Extension 3.

To compare terminal velocity between the loop of 8 and the loop of 10 modules we can scale the terminal velocity by dividing by the length of the loop. For the case with 8 modules, this corresponds to a speed of $0.4/(8 \times 0.06) = 0.833$ loop lengths per second, while for the case with 10 modules this corresponds to a speed of $1.29/(10 \times 0.06) = 2.15$ loop lengths per second. For configurations with 12 and 14 modules, the speeds in loop lengths per second are smaller than for the case of 10 modules since the weight of the robot plays a more significant role. It is harder to maintain or change the shape of a robot with more number of modules.

This number is a measure of speed that accounts for the difference in size of the loops and shows that a loop with more joints has a higher velocity in terms of

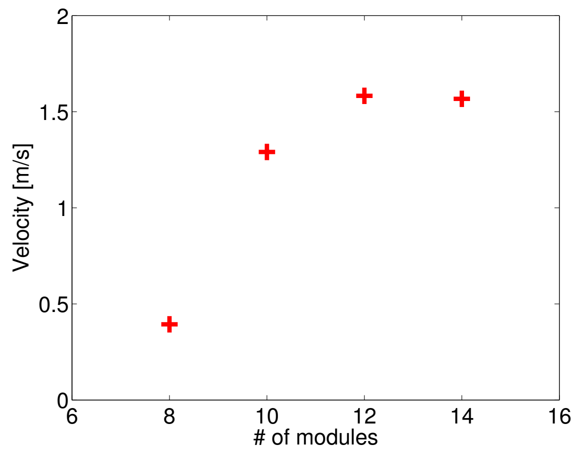


Figure 4.14: Terminal velocity vs. number of modules. Apex angle was 50 degrees in all experiments.

loop lengths per second. A loop with more joints can more accurately accurately approximates the arc of the shape. Thus, these results show that if the arc is more accurately approximated the faster the gait is.

In a loop of 14 modules the servos had enough torque to maintain its shape however saturation still occurred. This could be explained by limitations of the touch sensors that operate at 60Hz and speed of the servos when changing shape. There were no issues with stability on smooth level terrain in the transverse plane with the larger or smaller loops.

4.8 Discussion

One of the major findings of this work is that elongated desired shapes at touch-down for a rolling loop lead to higher terminal velocities. This is shown through a combination of simulation and experiments. The result makes sense intuitively as more elongated shapes create a larger moment arm due to the center of gravity w.r.t. the ground contact point. Because of this greater moment more acceleration occurs in the falling motion and more energy is put into the system at each step, a

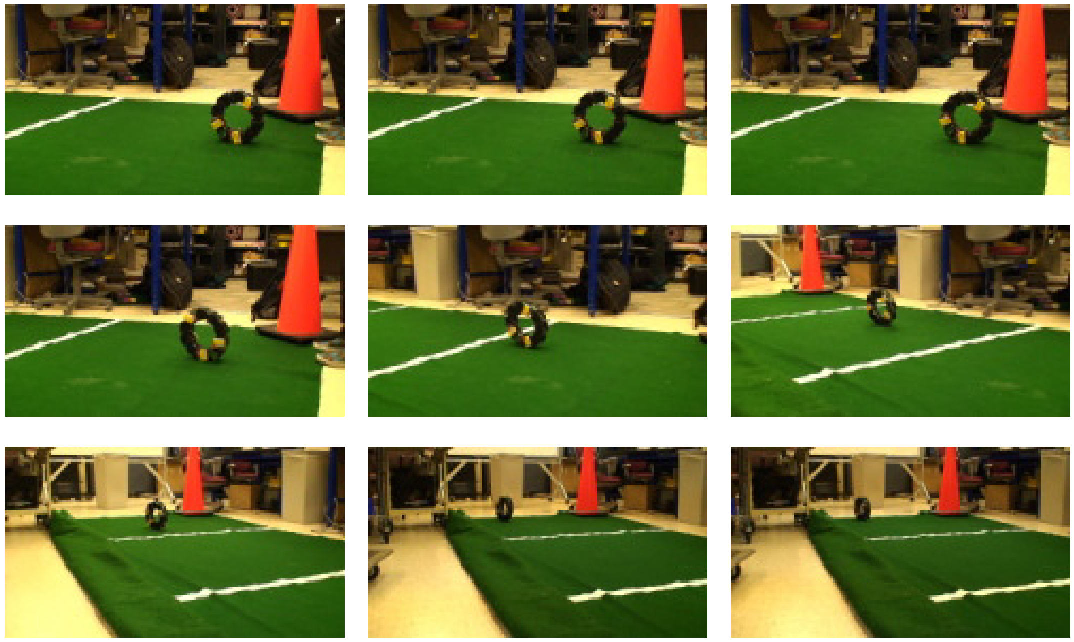


Figure 4.15: Snapshots of the rolling motion. Speed was controlled by a joystick in real time. Frames are in chronological order from left to right, then top to bottom. The motion of the robot is from right to left.

result that agrees with our theoretical predictions. It is interesting to note that the acceleration phase of the dynamic gait is similar to the motion of an inverted pendulum which has been shown in the context of walking to be very efficient requiring no work input to move the center of mass [54]. The fastest experimental gait had a speed of 1.6 m/s (roughly 5.4 body lengths per second for the 10 module robot.) Since the robot can reconfigure to different bodies lengths and use different gaits for different applications, a more apt measure for speed may be to normalize to module size. Using this measure the 1.6 m/s translates to 26 module lengths per second. To the authors' knowledge this is the fastest gait for any untethered modular robot.

Although the experimental and the theoretical results for the terminal velocity are close, the experimental results are consistently lower. There may be several possible reasons for this difference between the predicted and the actual behavior. One of the main reasons is that we have not taken into account friction in the modules and did not build a motor model for the servos. Our model also assumes that the modules can be represented as rods (for determination of inertia parameters). The actual modules however have a complex shape that could have different moment of inertias. Also, the contact point is not an ideal hinge point. An individual module has a more complex shape and comes into contact with the ground at more than one point. Therefore contact dynamics would be a good place for improvement in the model.

The terminal velocities saturated for desired shapes with a high apex angle. Hardware limitations in the current prototype may be partly responsible for this saturation. At a speed of 1.6 m/s, touchdowns occur at about 27 Hz and the hobby servos used in the prototype are unable to track the desired shape changes for speeds higher than this. Limited bandwidth on the communications bus might be another reason for this saturation. We have observed frames representing touchdown being dropped by the controller which could result in the controller's inability to keep up with the desired shape changes.

Conversely, for smaller values of θ_a, the controller was unable to initiate motion in the robot. The desired shape needs to move through a certain angle for the center of gravity of the resultant shape to lie outside the base of support formed by the module on the ground. Thus, motion is only initiated after overcoming this initial load.

In simulation and experiments, we also show that although more elongated shapes lead to higher terminal velocities, rounder shapes have lower specific resistance. This means that more elongated shapes are less energy efficient. The result makes sense intuitively as rounder shapes need to travel less distance in joint space at each step. On the other hand at higher rolling speeds centrifugal forces come into play and modules on the top do not need to fight gravity.

While the most efficient gait may be the roundest one, it is also the slowest to accelerate. One strategy for faster yet still more efficient rolling is to start with an elongated shape to accelerate quickly, then decrease θ_a linearly with speed until $\theta_a = \theta_s$. As it gets faster the shape becomes less oval and more circular. At the limit the shape will be that of a perfect circle which will roll using zero energy when ignoring gravitational effects.

Discrepancy between theoretical and experimental specific resistances are due to limitations in the model as explained earlier. Additionally, the analytical power computed only takes into account the power used by modules that are moving. It does not take into account the power used by modules that do not change their joint angle. However, the critical result to note here is that the trend in variation of specific resistance found through experiments with change in the desired shape matches the trend found through simulation.

Figure 4.16 shows an interesting comparison between the specific resistance of the dynamic rolling gait of CKBot, other gaits for CKBot (including a kinematic rolling gait, a crawling gait and an inchworm gait) and other robotic systems with non-wheeled modes of locomotion like walking. Here, all quantities have been plotted on a logarithmic scale. Ideally, we would like to have as low a specific resistance as possible at as high a speed as possible. This corresponds to being on the lower right corner of the graph. Robots shown in Figure 4.16 range from very energy efficient robots such as the Gravity Walker by McGeer [59] to very fast robots like RHex [97] and iSprawl [51].

It can be seen that the specific resistance for a dynamic rolling gait for CKBot falls within a reasonable range of that for legged systems like RHex and iSprawl inspite of CKBot's lack of powerful actuators. However, the larger number of actuators on CKBot still raises the specific resistance substantially so that it is not as much lower than these fixed configuration non-wheeled robots, as would be expected.

Figure 4.16 also compares the dynamic rolling gait with other modular robot gaits. In Section 4.7, we saw that the dynamic rolling gait improved on the kinematic

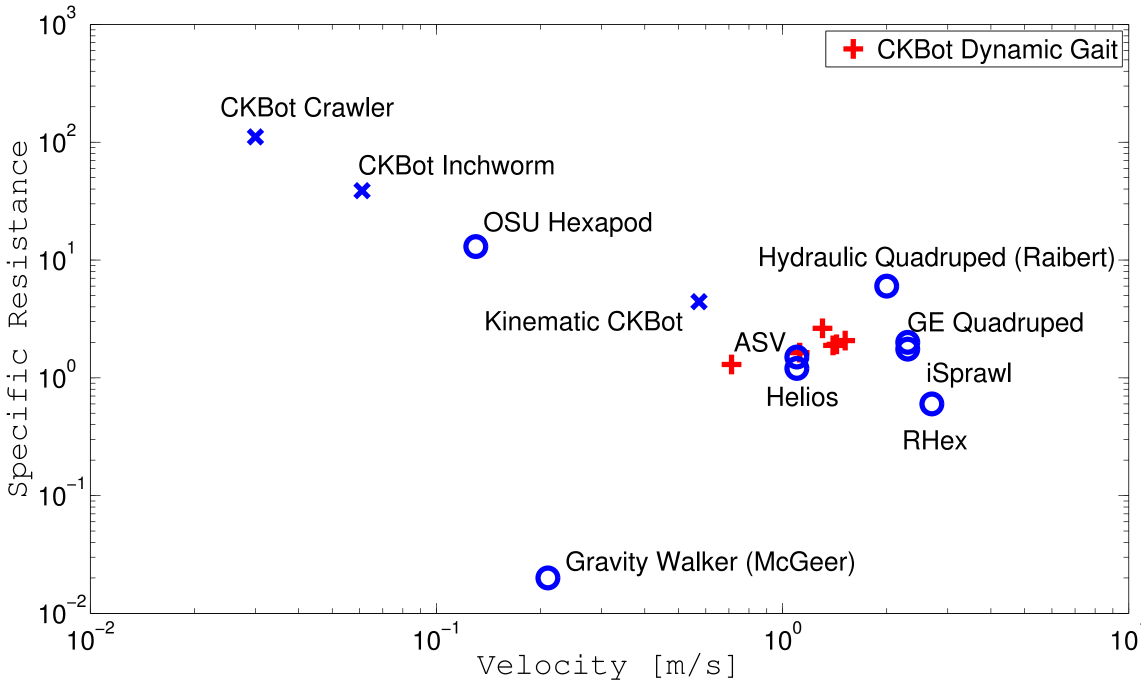


Figure 4.16: Specific resistance vs. velocity for several robotic systems. The data for walking systems included here is from [39].

rolling gait. From Figure 4.16 it can be seen that it is a dramatic improvement on the inchworm gait and crawler gait. The inchworm gait was implemented with 10 CKBot modules as well whereas the crawling gait was implemented with only 2 modules. These gaits are very slow and energy inefficient. The kinematic rolling gait is shown to have greater performance in terms of velocity and specific resistance, but it is the dynamic rolling gait that has pushed modular robots into the same range as walking systems.

The experimental results prove that this rolling gait is successful in traversing up and down inclines. Further, the trends in final measured terminal speeds for these cases match the expected trends, *i.e.* the robot rolls faster downhill than on level terrain and uphill.

We believe that the scaleability of the controller to configurations with different number of modules is a significant contribution of this work. It results from the choice of parameterization made for the controller and greatly reduces the computational complexity of scaling the controller. Thus, if a module or several modules break during a mission the system may continue after a simple reconfiguration discarding the failed modules. Conversely new modules can be picked up and added to the loop and the robot can keep going without having to expend significant resources to recompute control strategies.

The number of modules in the loop also has an effect on the performance of the rolling gait. A loop consisting of twice as many modules, with each module being half the length would have more joints, yet would retain its overall size in terms of length of the loop. As more joints are added to the loop the robot will more accurately approximate the arcs of the shape. The results show that making the arc less discrete will increase the velocity of the robot.

As the loop gets larger and larger the center of gravity of the robot gets higher. This should make the robot more susceptible to falling sideways in the sagittal plane. However, no significant instability in the sagittal plane was detected yet at a loop length of 14 modules in the case of CKBot.

The dynamic gait implemented in this work exploits the *passive falling dynamics* of the modular loop robot. Significant work has been performed in this area for walking robots [25, 26] where controllers are developed to take advantage of the passive dynamics of the robots to reduce torque requirements on the actuators. Indeed, McGeer's gravity walker [59] in Figure 4.16 has the lowest specific resistance amongst robots included in that Figure. Since modular robots have multiple actuated degrees of freedom, controllers that can reduce the requirements on the actuators would present significant benefits in extending the range and duration of operation of these systems.

The scalability of this controller addresses the interesting issue of *scalable dynamics* where models and controllers built for simpler systems can be easily adapted to larger systems. While this reduction to a simpler system was performed manually in our case, it might be possible to develop more general ideas for reducing complex configurations of modules to a simpler *abstracted* model for which controllers are easier to develop. Given the desire to ultimately extend this work to modular robots with hundreds or thousands of modules for which controllers would be extremely difficult to develop, the ability to abstract simpler models will play a significant role in being able to realistically deal with system of these sizes.

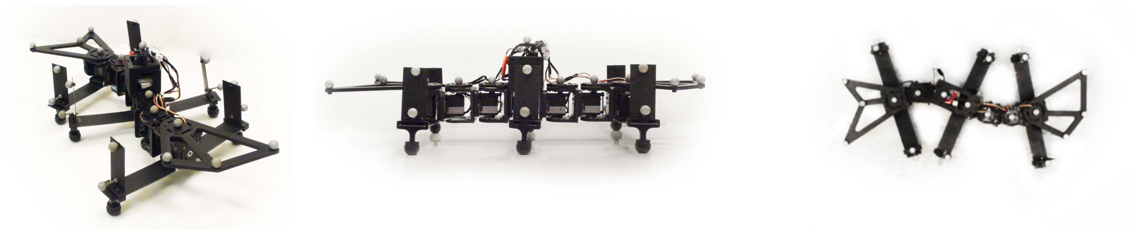
Chapter 5

Development and Analysis of a Dynamic Legged Configuration

As we saw in the last chapter, modules allowed us to analyze dynamic rolling One advantage of the modular design was that they enabled us to scale the robot in size. This in turn allowed us to run experiments while increasing the number of modules in the loop from 8 to 14 modules. In this section we will explore a legged configuration. This type of dynamic locomotion is potentially far richer in its parameter space, and greater in complexity. It is also an area in which we do not yet fully understand how to formulate the optimal design. The modular platform might thus be of particular utility here, not only in its scalability and ability to increase its number of legs like a caterpillar, but also as a method to explore experimentally different kinematic configurations.

5.1 Introduction

In the last decade we have witnessed an enormous increase in the interest in legged robots [74][49], primarily centering around the ability of legged platforms to traverse terrain that is difficult or impossible for wheeled vehicles. For quasi-static legged



(a) Isometric (b) Side (c) Top

Figure 5.1: Images of the Hexapod Configuration

systems, the governing mechanics are well understood [64]. The situation is starkly different for dynamically moving legged robots, where the combination of hybrid transitions, surface interaction and robot deformation often proves challenging to analyze [43][73]. Among the fastest autonomously powered legged robots are the Sprawl [19] and the RHex [3] robot families. Both are bioinspired, applying design principles and parameter ranges obtained from studies of cockroach (*Blaberus sp.* and *Periplaneta sp.*) locomotion.

The method we propose in this work of using Modular Robots to quickly build prototypes offer the chance to rapidly explore the parameter space of these legged robots. Such an approach has been the subject of ongoing work, primarily with a bio-inspired focus [56]. Our work on rapidly prototyping robots with the CKBot platform [77], lead us to explore the possibility of dynamical locomotion with these modules [83]. We present a robot of the same overall type as Rhex and Sprawl – a hexapod with elastic legs, whose mass of 1.6Kg lies in between that of the largest Sprawl and smallest RHex [83] as shown in Figure 5.1.

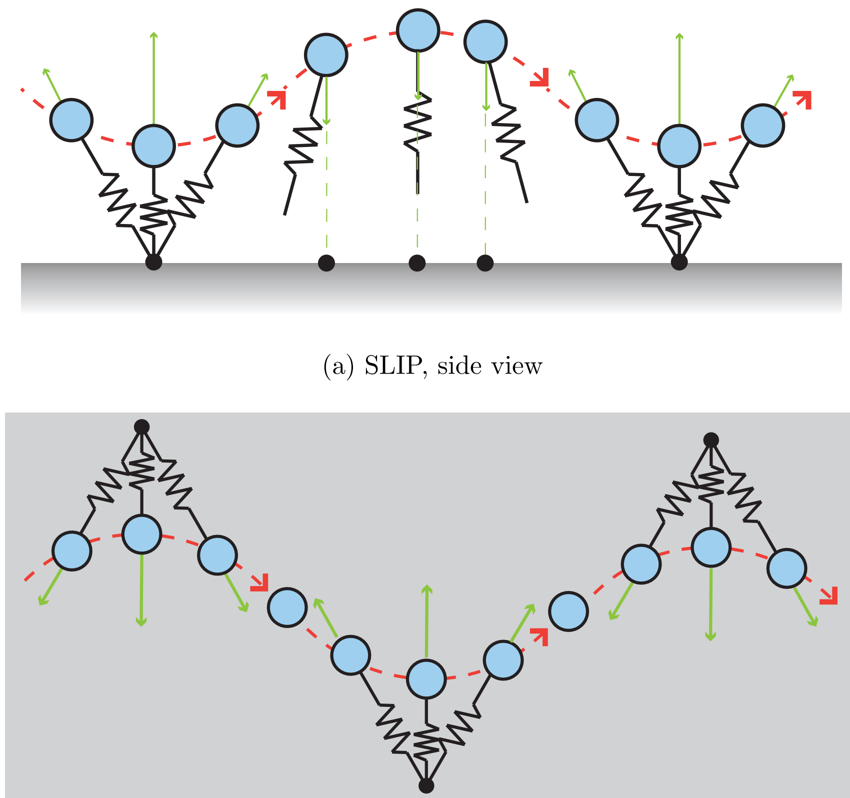
Passively compliant leg attachments are utilized to achieve a dynamic running gait using body articulation. Although the robot design and control strategy are, in principle, scalable to any number of leg pairs like a caterpillar, results are given for a hexapedal robot configuration. This prototype represents the first example of dynamic legged locomotion driven only by body articulation. Locomotion is achieved by actuating the spine of the robot, rather than motors that drive individual legs independently. We present an open loop controller which can be described as a modified Buehler clock with two slow phases instead of one to account for dependent opposing legs. Control parameters are optimized and we provide an analysis of the resulting gait.

5.2 Background

Legged locomotion is very complex, because it results from high-dimensional, non-linear interactions with the ground. Fortunately, simpler models, called templates have been made which resolve the redundancy of multiple legs, joints and muscles by seeking synergies and symmetries [36]. We do not yet fully know build legged robots, but these templates have proven invaluable in uncovering the basic principles and can act as guidelines for design. The primary templates are the sagittal-plane Spring-Loaded Inverted Pendulum (SLIP) [17] and the horizontal-plane Lateral Leg Spring (LLS) [85].

5.2.1 SLIP and LLS

The SLIP model was first proposed in the late 1970s [17]. The model, as shown in Figure 5.2a, consists of a single point mass representing the runner's body attached to a mass-less, prismatic, hooks-law spring that is intermittently attached to the ground via a frictionless revolute joint. This 'leg' is positioned during flight to touch down at a given angle. During stance the spring compresses and the point mass rotates as an inverted pendulum about the foot until the spring extension reaches the rest length and the flight phase begins. For clarity, only a single leg is shown as the left-right symmetry of motion in the sagittal plane renders each steady-state step equivalent. Despite the simplicity of the dynamic model, the SLIP 'template' [36], manages to accurately capture the fore-aft and vertical whole-body ground reaction forces and center of mass motions for a wide variety of running systems. Though running animals exist in a wide variety of shapes and sizes, biomechanical studies of these templates have shown that there are amazing similarities in the underlying dynamics among two-, four-, six-, eight-, and even forty-four legged creatures [10], [35].



(a) SLIP, side view

(b) LLS, top view

Figure 5.2: Characteristic COM trajectory and virtual toe point for the Spring-Loaded Inverted Pendulum (SLIP) and Lateral Leg Spring (LLS) shown in XZ and XY planes respectively. Forces on the COM are shown in green arrows. Resulting ZMP are shown in black dots. Note that the ZMP coincides with touchdown of the virtual toe during stance phase.

The SLIP model has proven useful in explaining the relative leg stiffness of animals of greatly different scales [6], the relationship between leg stiffness, speed, size, and stride frequency [30, 22], how humans adjust their legs stiffness to different surfaces [32], and in predicting how leg re-circulation strategies affect running stability [89]. More recently the SLIP model has begun to serve as a platform for investigating locomotion control schemes including how neuromuscular models are designed to control locomotion tasks [82, 52, 53]. In short, they provide the foundation for our understanding of running. Insights gained from these models have, in turn, led to the development of fast running robots. The first dynamic legged machines built by Raibert and Hodgins mimicked the pogo-stick morphology of the SLIP model and showed that artificial legged systems can be both fast and agile [74]. More recently, insect-inspired sprawled-posture hexapedal robots with compliant legs such as Sprawlita and RHex have been built, enabling enable fast running over rough and unknown terrain [18, 2]. These robots have demonstrated fast locomotion (up to 2.7m/s for RHex and 15 bodylengths/s for iSprawl) are able to run stably over natural terrain and over nearly body-height obstacles.

While SLIP provides a model describing dynamics in the sagital plane, the LLS template provides a model which describes dynamics in the horizontal plane. The LLS model similarly consists of a mass-less, prismatic, hooks-law spring representing the leg, attached to a point mass representing the body as shown in Figure 5.2b [85], where the spring intermittently attaches to the ground as a frictionless revolute joint. In this model, gravity does not play a role. For each half stride, the fore-aft force begins at zero, then decelerates the body, decreases to a minimum, then accelerates the body, increases to a maximum before returning to zero. The lateral force begins at zero, increases to a maximum, then goes back to zero. This results in the COM moving forward and oscillating side to side, as shown in Figure 5.2b [86]. Lateral characteristic motion of the center of mass is shown in a red dotted line and leg touch downs of stable LLS motion are denoted by a black dot. We see leg touch downs alternate to the left and right of the COM.

Unlike the SLIP model, the LLS model has been relatively less observed in experimental legged robotics. In fact, the only physical instantiation of LLS running is described by Shill et al [91], in which the robot is specifically designed for the purpose of embedding horizontal LLS dynamics. The bottom of the robot has ball bearings on which it glides over the ground to simulate flight phase. A pair of springs act as legs which are attached to a piston which is driven by a crank-slider mechanism, each powered by a single motor. The rotation of the hip joints can also be controlled by a second pair of motors. The absence of motion in the sagital plane obviously means there are no sagital SLIP dynamics.

5.2.2 Zero Moment Point

The zero moment point is a popular concept often used in the control of bipedal robots. It is the point on the ground at which all ground reaction forces balance all the forces acting on the robot during motion to maintain dynamic equilibrium. Because the sum of all moments of active forces with respect to this point is equal to zero it is named the zero-moment point (ZMP) [95].

We find the ZMP by finding the intersection of the acceleration vector to the ground plane. We see the ground contact point of the virtual toe of the SLIP or LLS model coincides with the location of where the ZMP is stationary. The ZMP has a velocity when its COM is moving during the aerial phase. With respect to the SLIP and LLS templates, Figure 5.2 shows that during the stance phase, the ZMP (shown using black dots) coincides with the touchdown location of the virtual toe. When in flight, the ZMP is directly below its COM, because no forces other than gravity act on the point mass.

By observing the touchdown location of the virtual toe, we can determine to what extent the robot follows SLIP- or LLS-like dynamics. In LLS-like dynamics, the virtual toe alternates to left and right of its COM during the left and right tripod stance, respectively, within a single stride.

5.2.3 Phase Estimation from Multi-Dimensional Time Series

In this work we look at the motion of the robot over many strides and trials, thus requiring a method for averaging data. We use the phaser algorithm for estimating the phase of our system to be able to view data in a cyclic manner as opposed to time [76]. Decomposition into a Fourier Series then lets us represent a typical model signal.

5.3 The Robot Configuration

Unlike the cockroach-inspired robots whose leg motions are primarily in the saggital plane, our robot also borrowed from the body-plan of hexapods (*Scolopendridae sp*), combining a laterally bendable body and laterally projecting legs. The configuration of the modules presented in this chapter is shown in Figure 5.1. Seven modules form the body or spine of the system which is articulated to achieve locomotion with three attached passive compliant fiberglass cantilever plates, each of which acts as a pair of legs. We attached a "foot" (a normally offset rubber pad) on each side of the leg cantilever. The design achieves an effective 2-DOF motion for each of its six feet with only seven motors.

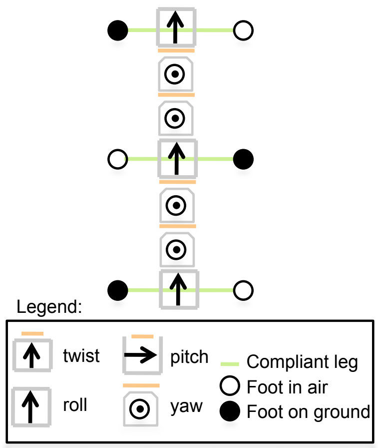
Figure 5.3a shows this configuration in the form of a diagram with all of its module's joint angles at an angle of zero. Three of the modules have a pair of legs connected. Let's call these the *hip* modules since they can roll the leg in the vertical plane with respect to the robot's local frame. In between each of these *hip* modules are two modules that allow the body to undulate in the transverse plane. Let's call these the *body* modules. To pose in a tripod stance the robot must use the *hip* modules to put alternating feet on the ground. Feet that are on the ground are shown with filled circles. Feet that are in the air are shown with open circles. To take a step the robot must use its *body* modules to undulate the body and propel the body forward. This is shown in 5.3b, where the robot takes a half stride in the left tripod stance.

Note that in a tripod stance, assuming each ground contact point at the foot acts as a ball joint, the robot imposes a four-bar linkage constraint in the horizontal plane. The constraint is superimposed on an image of the robot in Figure 5.3c. Assuming no slip at ground contacts, θ and ϕ are dependent variables enforced by the four-bar constraint as shown in Figure 5.3d. The link labeled λ is the leg length projected on the horizontal plane. This is, however, the horizontal leg length, which we can obtain from leg length L of the leg, using the roll angle of the leg γ so that $\lambda = (L/2) * cos(\gamma)$. To prevent feet from moving with respect to each other at ground contact, we have to solve for $\phi = f(\theta, \gamma)$.

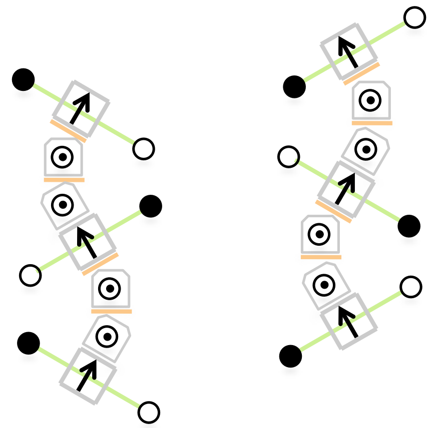
5.3.1 Local Reference Frame

When developing a high-level controller it is useful to have a robot base origin in which to describe the leg position. On a robot like Rhex, for example, it is clear how to select an intuitive local frame. The robot can be thought of as a cuboid with three rotating legs on each side. Local frames on that cuboid with origins at the attachment points of the legs is an obvious choice. On our Hexapod configuration, however, there is no obvious location for the local reference frame because the body is not a rigid body, but rather undulates like a snake while all of its joint angles are oscillating.

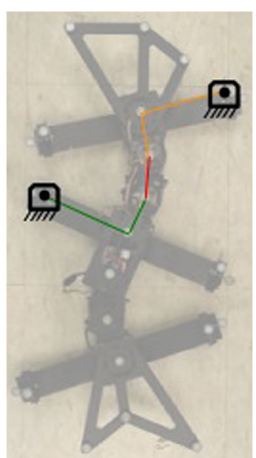
Figure 5.4 illustrates our method for fixing a reference frame onto an undulating body. Call each point where a leg attaches to the body b_i. Fit a line through all b_i, as shown in Figure 5.4a with a red dotted line. We place reference frames with origin at each b_i with the y axis pointing to the next b_i, as shown in Figure 5.4b.$y_i = b_{i+1} - b_i$. We determine a z-axis by crossing all body markers and normalizing. $v_i = b_i p_j . z_i = S y_i \times v_i . x_i = y_i \times z_i$ as shown in Figure 5.4c.



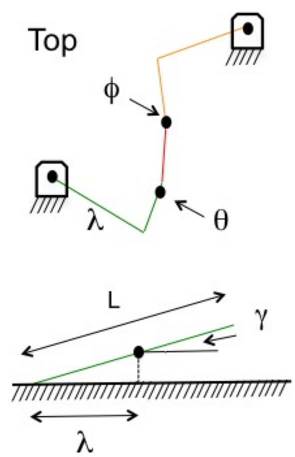
(a) Hexapod configuration with all joints set at angle zero



(b) Top view of the robot taking a half-stride in its alternating tripod gait. The filled circles denote a foot on the ground; an empty circle denotes a foot in the air. In this figure, forward motion is up and the robot is shown to take one step forward.



(c) This figure shows the four bar linkage imposed by the ground contacts in tripod stance between the front right foot and middle left foot.



(d) Degrees of freedom in the four bar linkage constraint

Figure 5.3: Diagrams of the Hexapod Configuration

(a) Fit curve through leg attachment points

(b) Set origins

(c) obtain z-axis

Figure 5.4: Method for fixing a local reference frame to a undulating body.

5.4 Control

Using the hexapodal configuration, we have been able to implement many different modes of locomotion such as walking, running, turning while running, turning in place, as well as sideways movement, rolling and caterpillar motion using different controllers.

To achieve running, the six-legged robot uses a simple alternating tripod gait. In an alternating tripod gait, the front and rear legs on one side move together with the middle leg on the other side. One step of the motion is shown schematically in Figure 5.3b in the horizontal plane. In the diagram the front and rear legs on the right side and the middle leg on the left side provide a triangle of support. As can be seen in Figure 5.1, a pair of legs is formed from one piece of fiberglass. By twisting the three pairs of legs relative to each other, one end of the fiberglass piece will point downward and the other end upward. This motion shown is achieved by changing the angles of the module attached to the leg pair. We'll call the angle of the *hip* module β_{hip}. The joint angles of the modules in between can move legs in the horizontal plane; we'll call these β_1 and β_2. A *hip* module together with the *body* module in front and the *body* module behind (total of three modules) are called a *segment*.

We then developed a controller for a segment in the local frame, described in the previous section. We call the local coordinates of the leg: θ_{yaw} (forward and backward motion) and θ_{roll} (up and down motion). One can think of these coordinates as moving the foot in the sagittal plane of the robot. We allow a bend to the spine that will prove useful later for turning gaits, and which we will set by a variable we call θ_{bend} angle. We can transform these local coordinates into the joint angles called β_1, β_2, and β_{hip} to be sent to the *body* and *hip* modules of a segment by inspection:

$$\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_{hip} \end{pmatrix} = \begin{pmatrix} 2 & 0.5 & 0 \\ -2 & 0.5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{yaw} \\ \theta_{bend} \\ \theta_{roll} \end{pmatrix}.$$

If we use a sinusoidal pattern on the yaw of the leg and a cosine pattern on the roll, then we can get a elliptical foot trajectory:

$$\begin{pmatrix} \theta_{yaw} \\ \theta_{bend} \\ \theta_{roll} \end{pmatrix} = \begin{pmatrix} sin(\phi + ecc) * A_{yaw} \\ bend \\ cos(\phi) * A_{roll} \end{pmatrix}.$$

By varying the amplitudes A_{yaw} and A_{roll} we can elongate the foot's trajectory in the respective directions. Changing *ecc* will shear the leg trajectory. Note that the controllers are with respect to phase ϕ, not time. The phase is the output of our Buehler clock controller which we will describe next.

Leg Trajectory Controller Variables	
A_{yaw}	yaw angle amplitude of both legs
A_{roll}	roll angle amplitude of both legs
ecc	eccentricity of both legs
$Bend$	bend of body

Buehler Clock Controller Variables	
t_c	period of rotation for both legs
t_{sl}	duration of slow swing for left leg
t_{sr}	duration of slow swing for right leg
ϕ_{sl}	sweep phase duration for left leg
ϕ_{sr}	sweep phase duration for right leg
ϕ_{ol}	sweep phase offset for left leg
ϕ_{or}	sweep phase offset for right leg

Table 5.1: This table shows leg trajectory controller variables for the Hexapod's Buehler clock, which controls two legs and thus prescribes two slow phases.

5.4.1 Buehler Clock Controller With Two Slow Phases

We achieved stable forward locomotion, by using an open-loop controller, and setting a desired leg trajectory driven by a Buehler clock. This type of controller was first implemented on the RHex robot. In a conventional Buehler clock, there is only one slow phase as there is only one stance phase for each leg. In our hexapod robot, however, the left and right legs are coupled to each other. We therefore run both legs on one Buehler clock and employ two slow phases, one for each leg, as shown in Figure 5.5. In the hexapedal configuration, the robot has three Buehler clocks, one for each pair of legs. The front and last pairs are in phase and the middle leg pair runs half a cycle out of phase.

The traditional Buehler clock is parametrized by four variables: period of rotation, duration of slow swing phase, angle of slow swing phase and sweep angle offset. Our Buehler clock requires seven parameters because there are two sweep angles, two angle offsets, and two duty cycle parameters, one for each of the two legs, as shown in Table 5.1. Each clock still only has one period parameter t_c. Control of

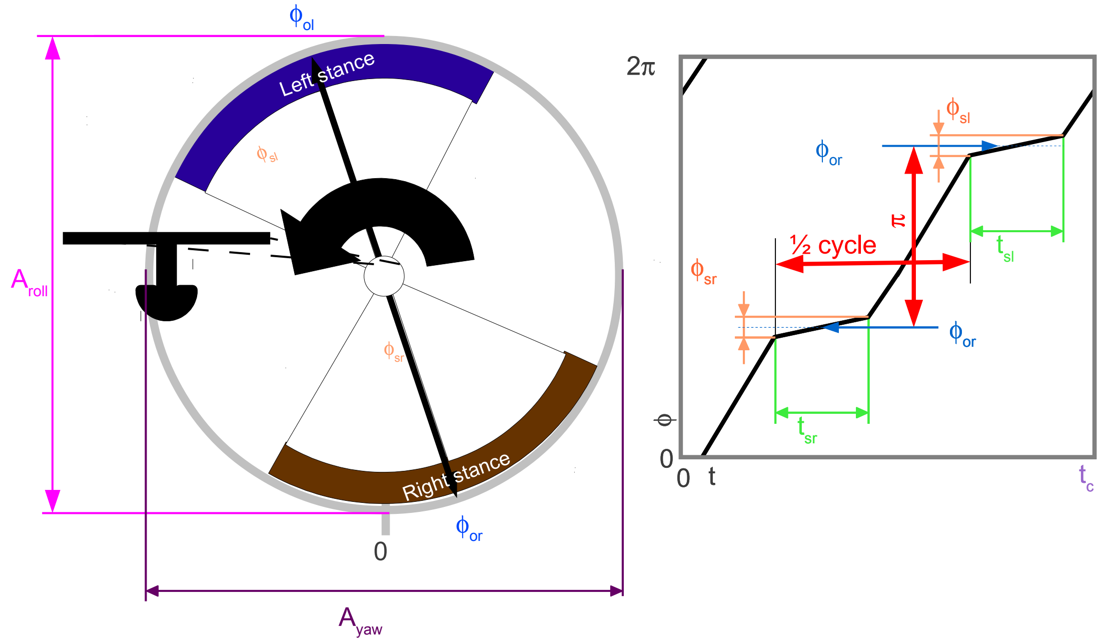


Figure 5.5: Buehler clock with two slow phases. For clarity, eccentricity *ecc* is set to zero and roll and yaw amplitudes of the legs, A_{roll} and A_{yaw}, are set equal to each other.

the gait timing is achieved by modifying these parameters.

5.4.2 Turning while Running and Turning in Place

Turning while running is achieved by adjusting the ratio between sweep angles ϕ_{sl} and ϕ_{sr}. Turning in place can be achieved by controlling the foot trajectory in a figure eight instead of an ellipse like so:

$$\begin{pmatrix} \theta_{yaw} \\ \theta_{bend} \\ \theta_{roll} \end{pmatrix} = \begin{pmatrix} sin(2 * \phi) * A_{yaw} \\ 0 \\ cos(\phi) * A_{roll} \end{pmatrix}.$$

5.4.3 Leg design

Biomechanists have long known that dynamically locomoting animals share some non-obvious commonalities [27]. Running gaits in particular share the property that ground reaction forces sum up to those of a virtual monoped ("pogo-stick"), whose center of mass dynamics are described by the SLIP model [7]. These gaits can be compared across animals by choosing non-dimensional variables to describe the motion: a non-dimensional force $\frac{F}{mg}$ and a non-dimensional leg deflection $\frac{\Delta l}{l}$. The ratio of these is a non-dimensional stiffness, that is broadly conserved across all animals that exhibit alternating trot-like gaits, even when these range in mass from grams to tons [8]. Cockroaches, whose running behavior has been carefully studied, have been found to occupy a stability "sweet spot" in the parameter ranges of the LLS model [84][87].

The RHex robot [3] was built to capitalize on the insights obtained from these biomechanical studies, by using simple legs whose stiffness fell within the biologically observed range. To guide the leg design for our robot we also looked at leg stiffnesses found in nature [9], where a wide range of animals are found to have an average relative leg stiffness of 7.82 for polypedal trotters and 9.76 for bipedal runners. We hypothesize that this same range will give our robot the most natural SLIP like gait. We have developed two sets of legs. One that is close to the relative stiffness found in nature with a relative leg stiffness of 12.4. We made another set of legs that were more compliant with a relative leg stiffness of 2.67. We name each set of legs the *stiff* and *soft* set respectively. We used the convention of [10] to calculate relative leg stiffness of the monopod's leg using $k_{rel} = \frac{\frac{F}{mg}}{\frac{\Delta l}{l}}$. Substituting $k = \frac{F}{\Delta l}$ we get $k_{rel} = \frac{kl}{mg}$ where k is the combined stiffness of all legs in contact with the ground during the stance phase. With our tripod gait this would be the combined stiffness of all three legs between the robot and the ground.

Initially legs were made out of ABS for its ease of manufacturing, but repeated use would cause fractures. We eventually selected S2-6781 pre-preg fiberglass (from Applied Vehicle Technologies, Indianapolis, IN), as the material of choice for several reasons including its relatively low density and Youngs modulus, high yield strength, comparatively high specific strain energy capacity and low material cost. In addition to these properties, composite laminates expand the available design space by offering the ability to vary the Youngs modulus value by adjusting orientation of the plies. [37].

We assume that the leg functions as a cantilever beam and that vertical force is linear to displacement. Applying force at the toe we find stiffness $k = F/x$. The total mass of the robot m is 1.6Kg. Each leg in tripod stance should see a third of this load. We apply a mass of 0.5Kg and 0.212Kg and measured deflection. Fore-aft forces at the feet caused the legs to twist. We determined the torsional stifness by applying the same masses and measuring the angle of rotation.

	stiff	soft middle	soft front/rear
k [N/m]	649	153.3	88.7
k_τ [Nm/rad]	2.0	1.4	1.1
k_{rel}	12.4	2.67	

Table 5.2: Stiffness measurements in the vertical direction k and torsional stiffness k_τ for individual legs. These are used to determine relative leg stiffness k_{rel} of the whole robot in a tripod stance.

5.5 Experiments

5.5.1 Automated Gait Optimization

As described in section 5.4, gait parameters can be varied for the forward running gait. A joystick interface allows the user to easily change these parameters, tuning the robot's gait for speed or efficiency. However, even a parameter space of two variables provides for a cumbersome manual search; manually tuning more than two variables is virtually impossible. We use an automated gait tuning method similar to [97] and use a Nelder Mead simplex based gradient search. Automated tuning of the RHex gait parameters brought its speed to $2.7m/s$ – slightly over 5 body lengths per second [96]. This approach is able to work in a noisy environment and where gradients can not be calculated at each point. It incurs, in principle, the least experimental cost per step of any of the other "direct search" (derivative-free) methods. For this method, an initial simplex is required, which we derived from the parameters of a gait that had been tuned manually.

A VICON system streams the locations of robot markers. A Python program computes the mean x,y and heading of the robot. The robot was kept on track by a human operator using a joystick. Pressing a button would start and end a trial. The optimization attempted to minimize time divided by distance between robot position at start and end of each trial by changing the gait parameters. The gait parameters that were optimized were frequency, leg roll amplitude, leg yaw amplitude (achieved via body bending), sweep angle and duty cycle (relative duration) of the stance period.

5.5.2 Motion Capture Data Set

After optimization we collected tracking data from all the markers while running the robot back and forth. A total of 52 trials were conducted with an average of 372 frames and an average distance of 3.0m (minimum of 303 frames and 2.1m). From one end to the other end of the arena counted as one trial. Trials were segmented by thresholding velocity of the robot at 0.5m/s as shown in Figure 5.6.

The arena contained 16 cameras running at 100fps, and covered a carpeted area of 5m x 5m. We removed marker association errors using a combination of manual and automated tools. An average of 1.4% of markers were occluded in a trial and

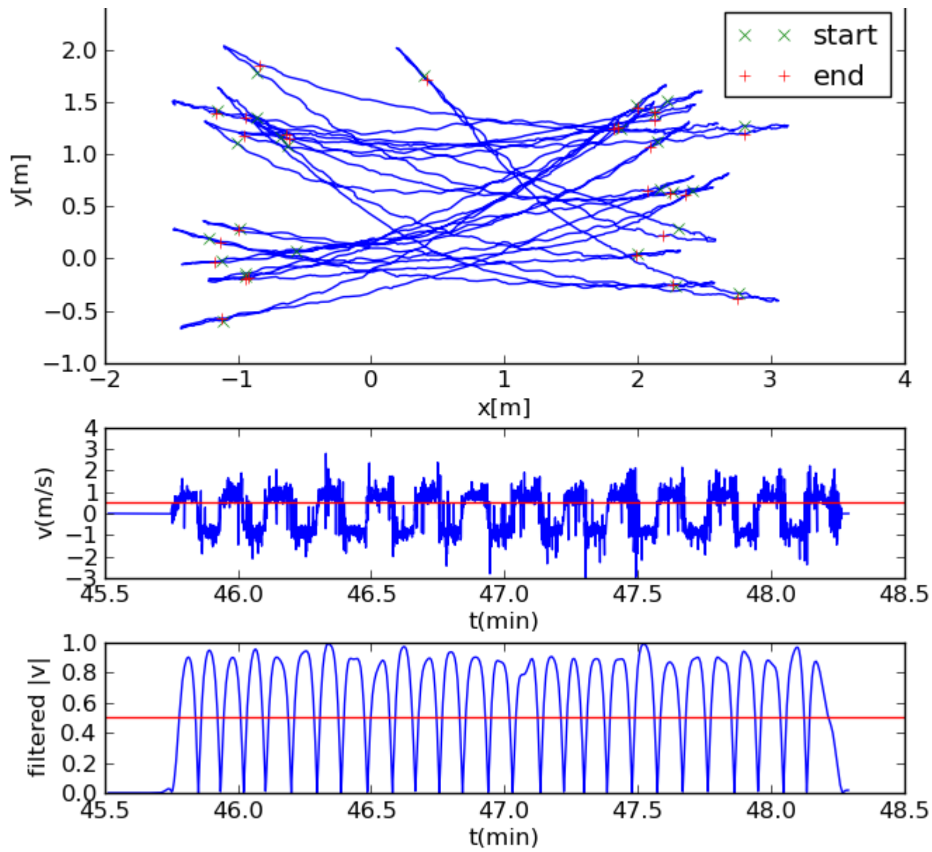


Figure 5.6: Segmenting trials. The top Figure shows a top-down view of the robot as it runs back and forth in the world frame. The green x denotes the start of a trial. The red + denotes the end of a trial. Velocity is shown in the middle figure. To detect the start and the end of a trial, we filter the magnitude of the velocity and threshold at 0.5 m/s as shown in the middle figure.

were linearly interpolated.

Marker positions and labeling are shown in Figure 5.7. Each module has a marker placed on top and is numbered 0,5-9. Assymetric plates with four markers are placed on the head and tail labeled 0-4 and 9-12 respectively. At each foot are three markers called Top (16,22,28,13,19,25), Corner (17,23,29,14,20,26), and Side (18,24,30,15,21,27).

	soft forward	soft backward	stiff forward	stiff backward
mean $\|v\|$	0.67 +/- 0.09	-0.69 +/- 0.06	0.7+/-0.1	-0.7+/-0.1
trials	13	13	14	13
mean frames	370	374	422	410
min frames	204	295	321	361
max frames	459	444	493	461
mean distance	3.2	3.3	2.6	2.7
min distance	2.4	2.9	1.4	2.1
max distance	3.8	3.7	3.3	3.3
mean strides	13	12	13	13
min strides	10	11	11	10
max strides	15	14	15	16

Table 5.3: Statistics of four datasets. Two sets of legs, "soft" and "stiff", were run forward and backward. Mean, smallest and largest number of frames and distance traveled are shown for each dataset.

5.5.3 Detecting Stance Phase

To detect stance phase, we must first be able to determine when a single foot is in contact with the ground. Figure 5.8a shows a typical trajectory of a corner marker of an individual leg in the xz plane. We detect ground contact using two selection criteria. First, we threshold on the velocity in x of a corner marker. Velocity was filtered by a second-order butterworth filter with a cut-off frequency of 0.3Hz. The second criterion was selecting points with z within 15 mm above a base line. The baseline was obtained by taking the median of the z height of all the legs and filtering with second-order butterworth filter with a cut-off frequency of 0.01Hz which is shown in Figure 5.8b with the depicted by a red line and the height of all six corner markers plotted in multiple colors.

5.5.4 Zero-Moment Point of Virtual Leg

We determine the position of the center of mass by averaging the positions of all the body markers. $\mathbf{p}_{CoM} = \frac{\sum_{0 \leq i < n} \mathbf{p}_i}{n}$. We differentiate that twice and use a second-order low-pass filter and cut-off frequency of 0.25Hz after each differentiation to

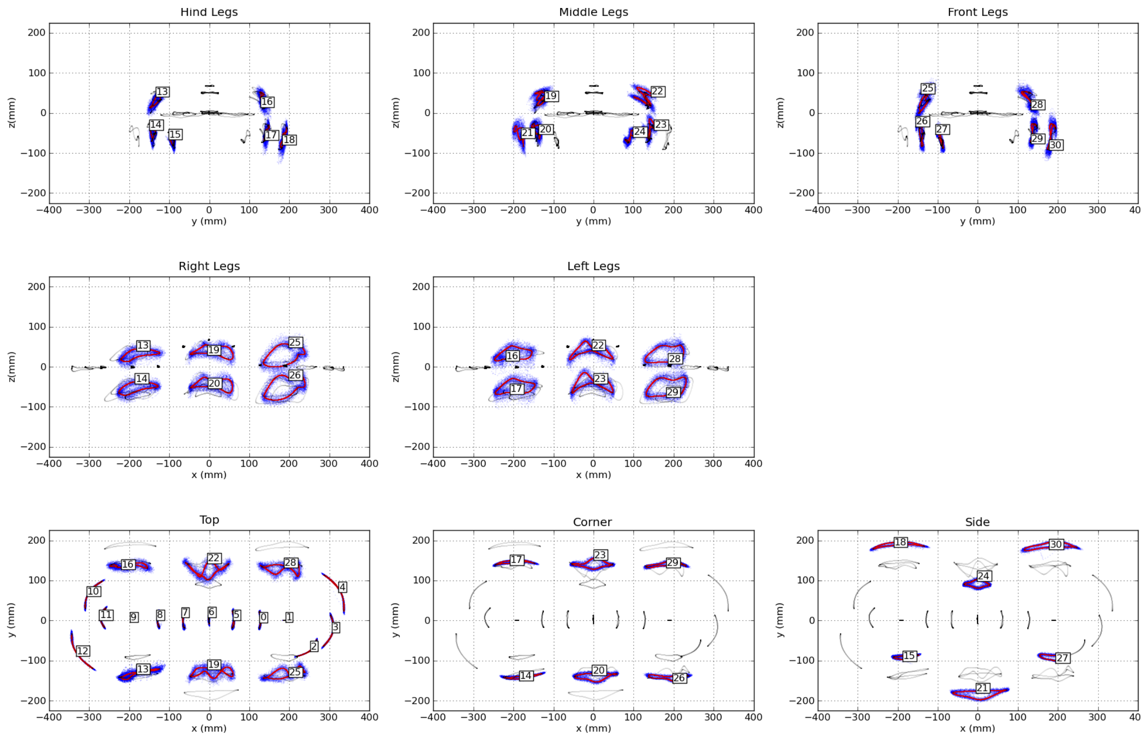
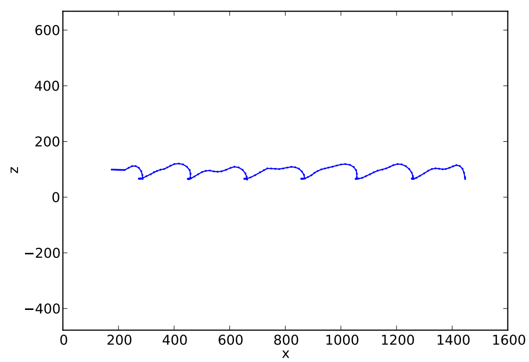


Figure 5.7: Numbering of Vicon markers in different section planes. The first row shows the YZ plane (viewed from the front), the second row shows the XZ plane (viewed from the right), and the third row shows the XY plane (viewed from the top).

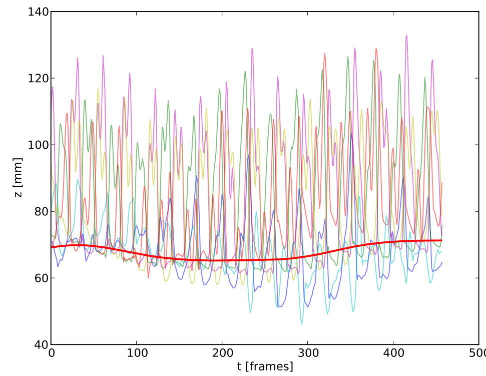
obtain the acceleration vector $\mathbf{a}_{com} = \frac{d^2}{dt^2}(\mathbf{p}_{CoM})$. The ZMP point of the virtual leg is derived by finding where the acceleration vector intersects the ground plane and goes through its COM. $\mathbf{p}_{CoM} + \lambda * \mathbf{a}_{com} = \mathbf{p}_{ZMP}$, where $\mathbf{p}_{ZMP} = [\ x \quad y \quad 0\ ]^T$. We solve for λ, then obtain $\mathbf{p}_{ZMP}$, and eliminate all ZMP points which aren't stationary $|\frac{d}{dt}(\mathbf{p}_{ZMP})| > v_{thres}$ where threshold $v_{thresh} = 15$ m/s.

5.5.5 Leg Trajectories

We used the fore-aft motion of the legs in the local frame centered around the mean to reconstruct the dynamical phase of the gait limit cycle using the Phaser algorithm [76], allowing us to reinterpret our data as a function of phase. The fore-aft motion of the rear right leg was used as the poincare section function for a convenient place



(a) Typical trajectory of a corner marker of an individual leg in a single trial. The trajectory is plotted in the xz plane in the world frame. The robot moves from left to right.



(b) The red line shows the base line computed by taking the median height of all six legs, before filtering using a second-order butterworth with a cut-off frequency of 0.01Hz. The toe height of the other legs is shown in multiple colors.

Figure 5.8: Detecting ground contacts using 2 criteria: x velocity and z height.

to determine when the phase equals zero. We then constructed Fourier series models of the marker trajectories with respect to phase ϕ.

5.6 Results

5.6.1 Gait Optimization

The results of the gait optimization confirmed the importance of a Buehler clock. We start with a hand-tuned gait, only varying gait frequency, and the roll and yaw amplitude of the leg trajectory. Without implementing a Buehler clock and thus leaving out the slow and fast phases, we achieved a speed of 0.4m/s with the "stiff legs". An automated optimization of this gait tuning with these parameters led to a speed increase of 20% to 0.5m/s. We observed that this was a less bouncy gait with less aerial phase, and yet more fluid in comparison with the hand-tuned gait. Adding a Buehler clock with slow and fast phases to the parameterization yielded an additional 21% increase to 0.58m/s after optimization. Note that this is an overall

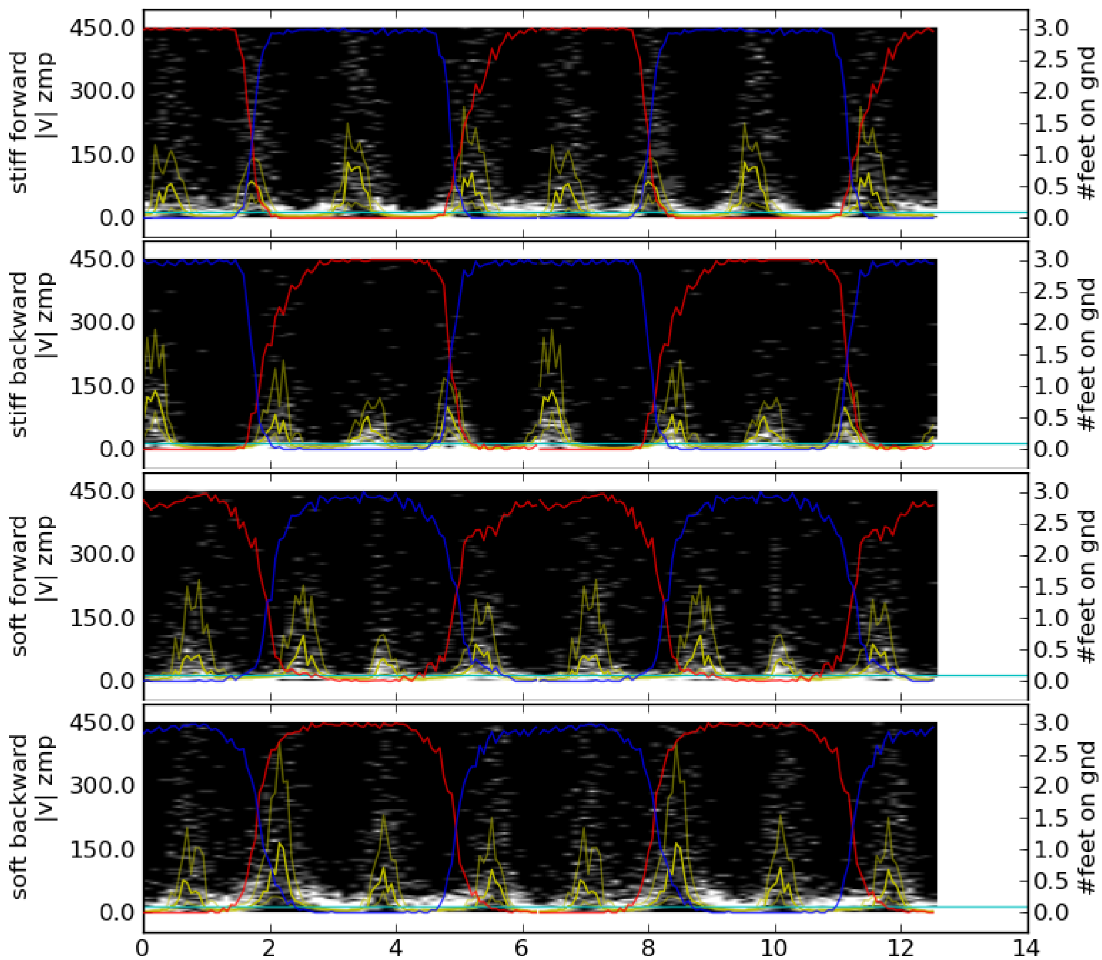


Figure 5.9: This figure shows the speed of the ZMP location in the world frame on the left y-axis with respect to phase in white. We computed median, upper and lower quartiles shown in yellow. We used the ZMP points where the median velocity is slow, below a threshold of 15 m/s shown in a cyan line. The red and blue lines show the number of legs of the right and left tripod respectively that are in stance phase on the right y-axis. Note that slow ZMP points are found at full tripod stance and between half strides, which is where we transition from one tripod to the other.

Leg	Freq. Hz	Roll deg	Yaw deg	Sweep deg	Duty %	Speed cm/s	K_{tri} N/m	L mm	H mm
Soft	3.1	33	11	22	19	78	331	152	68
Stiff	3.5	22	12	20	18	67	1950	137	43

Table 5.4: Gait parameter optimization results of soft and stiff legs.

speed increase of 45% in comparison with the hand-tuned gait.

Optimizations were performed on both sets of legs and terminated with gait parameters, as shown in Table 5.4. We observed that soft legs give 16% higher speed with a 11% lower gait frequency.

5.6.2 Specific Resistance

Specific resistance was measured on a hand-tuned gait using the set of stiff legs. Current was recorded as a measure of power and velocity was determined by measuring time for the robot to travel between two lines that were placed three meters apart. We performed a total of 10 trials, but one was discarded due to operator error in that trial. Specific resistance was measured to be 3.7±0.3, at a speed of 0.55±0.05m/s.

5.6.3 Leg Trajectories

We determine the local reference frame of the robot and plot the markers attached to the backbone and the leg in this local frame. Figure 5.10 shows a side, front and top view. As expected, we see more deflection in the xy plane of the leg corner markers.

5.6.4 Stance Phase

Figure 5.11 shows when ground contacts occur for each foot for a typical trial. This particular figure shows the data for a set of stiff legs. The top three denote the three legs from the left tripod. The bottom three rows denote the legs from the right tripod. They are colored black if touchdown has occured; white means there is an

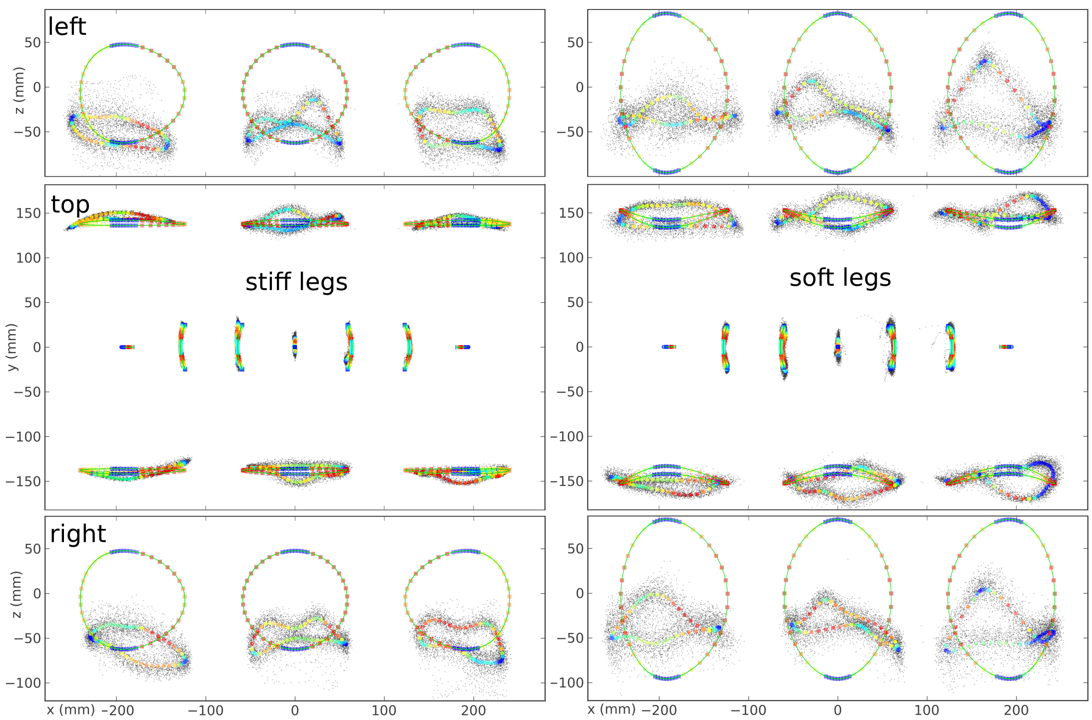


Figure 5.10: Plots show trajectories of body and leg corner markers, with colors indicating speed in 3D space (red faster, blue slower). We plot the positions predicted by a rigid body model (square color markers on a solid green line), the observed positions (point cloud of black points) and a Fourier series model of the observed positions (round color markers). Plots show side (XZ) views of left and right legs, as well as top (XY) view.

aerial phase. Touchdowns are plotted against time in frame number on the x-axis. We see a solid alternating tripod stance. The robot is either in left tripod stance or right tripod stance. The figure also shows that there is hardly any aerial phase.

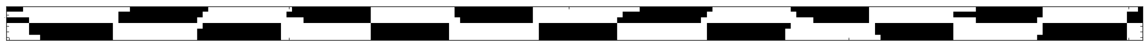
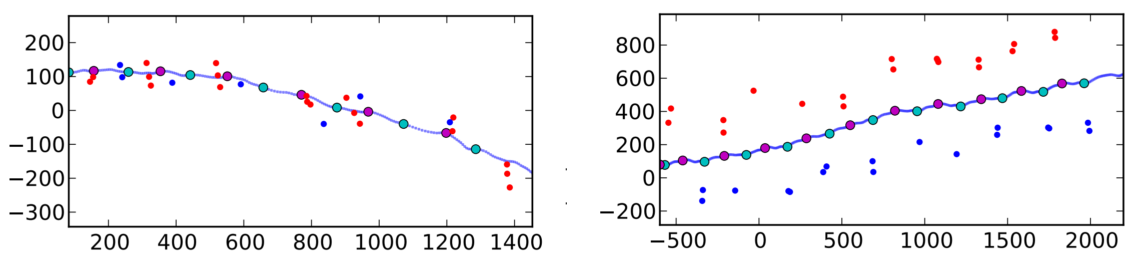


Figure 5.11: A typical trial of ground contact of feet vs time in frames. Black denotes touchdown, white denotes aerial phase. The top three rows denote the left tripod. The bottom three rows denote the right tripod

5.6.5 Zero Moment Point

Figure 5.12 plots the ZMP positions of the robot in the world frame. The blue line shows the motion of the center of mass of the robot. The motion of the robot is from left to right. Shown in red are the left tripod ZMP locations, in blue are shown the right tripod ZMP locations. The location of the corresponding COM is shown in alternating purple and cyan. Figure 5.12b shows that the ZMP swings much more side to side compared to the stiff legs shown in Figure 5.12a.



(a) stiff legs

(b) soft legs

Figure 5.12: Comparison of ZMP and COM motion between soft legs and stif legs in the world frame. The robot is moving from left to right. The figure shows only one typical trial of each set of legs running in the forward direction.

Table 5.5 shows a point cloud of ZMP positions during stance phase in the local frame of all four data sets: stiff and soft legs running forward and backward. Red and blue dots mark the ZMP at stance phase for the left and the right tripod,

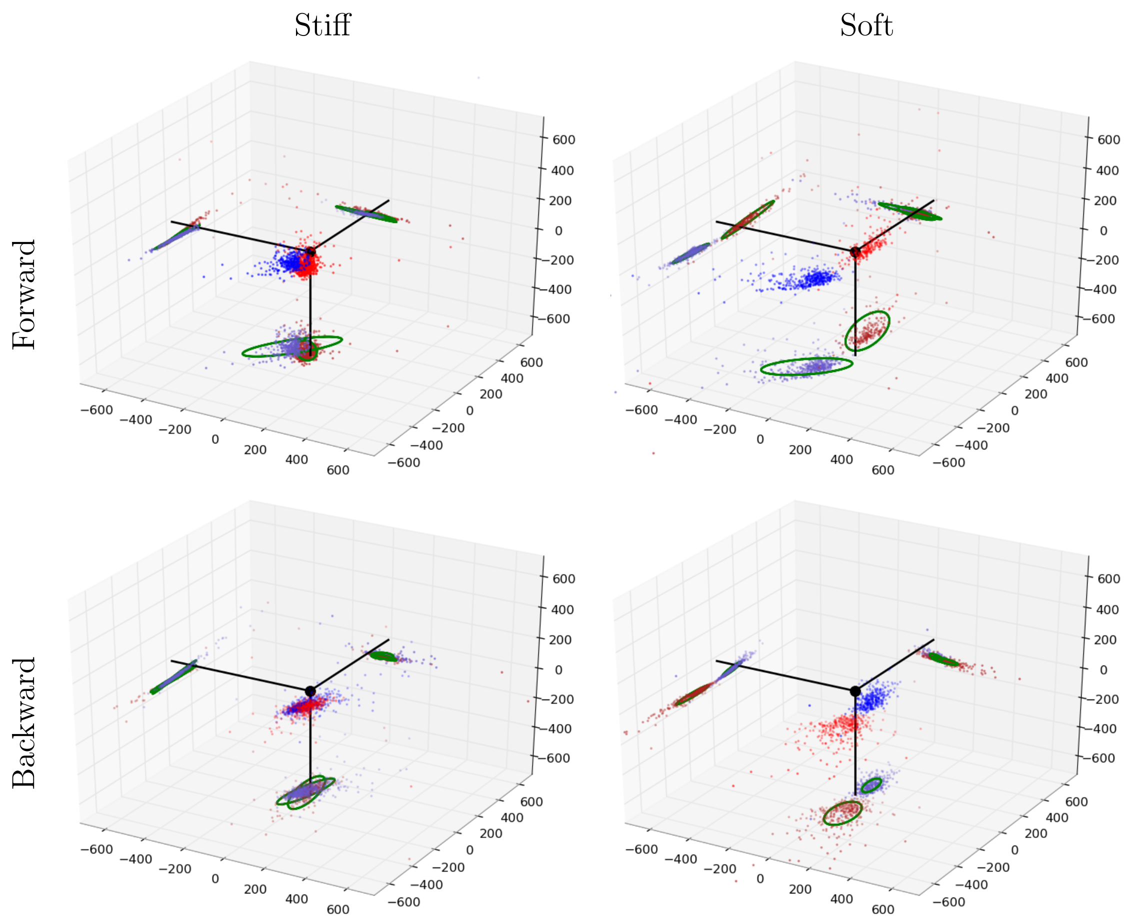


Table 5.5: Comparison of the ZMP between soft legs and stiff legs in the local frame running forward and backward. The ZMP is plotted in red and blue. Green ellipse denote the standard deviations along the principal axes. For clarity, projections of the ZMP onto the planes are shown in dark red and dark blue. Center of Mass is denoted with a black circle and axes extending to the planes are drawn in black.

respectively. These points are projected on the three planes in a darker red and darker blue. We draw axes projected from COM to the planes for clarity.

Table 5.5 show that the stiff legs have their ZMP centered underneath its COM forward and backward and that soft legs run with their ZMP to each side of its COM. This is true for both running directions, both forward and backward.

5.7 Discussion

There are two main findings with the hexapod configuration. First, soft legs run at a higher speed than stiff legs, even though these were closer to the relative leg stiffness found in animals. Second, we found LLS-like dynamics, something which has not yet been reported in any other robot with intermittent ground contact. This could be attributed to the morphology of the robot, which has a laterally sprawled posture.

5.7.1 Specific Resistance

Figure 5.13 compares the performance of a hand-tuned dynamic running gait with our robot's dynamic rolling gait as well as other robots in terms of speed and efficiency. It can be seen that the dynamic running gait denoted in a green $\star$ has lower speed and efficiency compared to the dynamic rolling gaits denoted in a red $+$. The smallest dynamic rolling gait contained eight modules, where the running gait used seven modules. This could be because rolling is more energy efficient than running on flat ground, but is perhaps an overgeneralization and too simple an explanation since there are many factors that contribute to the speed and efficiency metric. Also, remember that Tad McGeer's Gravity walker is the most efficient robot on this chart.

Compared to other legged robots, such as RHex and iSprawl, our Hexapod is still slower. A possible explanation is that we used simple hobby servos and use significantly less powerful actuators. Nonetheless, beating other robots was not our prime goal. In the modular approach, we have specifically favored shorter build time at the cost of performance. The advantage of the modular approach is that we can build our prototypes quickly. The disadvantage is that we use motors which were not specifically selected for this task. What is available to us is a homogeneous set of motors with fixed size actuators. Perhaps a more interesting comparison point then is the performance of Loco-kit, the only other modular robot whose specific resistance and speed is reported in the literature. With a specific resistance of 9.65 at a speed of 11.8 cm/sec [56], we exceed their Loco-kit's performance in terms of both efficiency and speed.

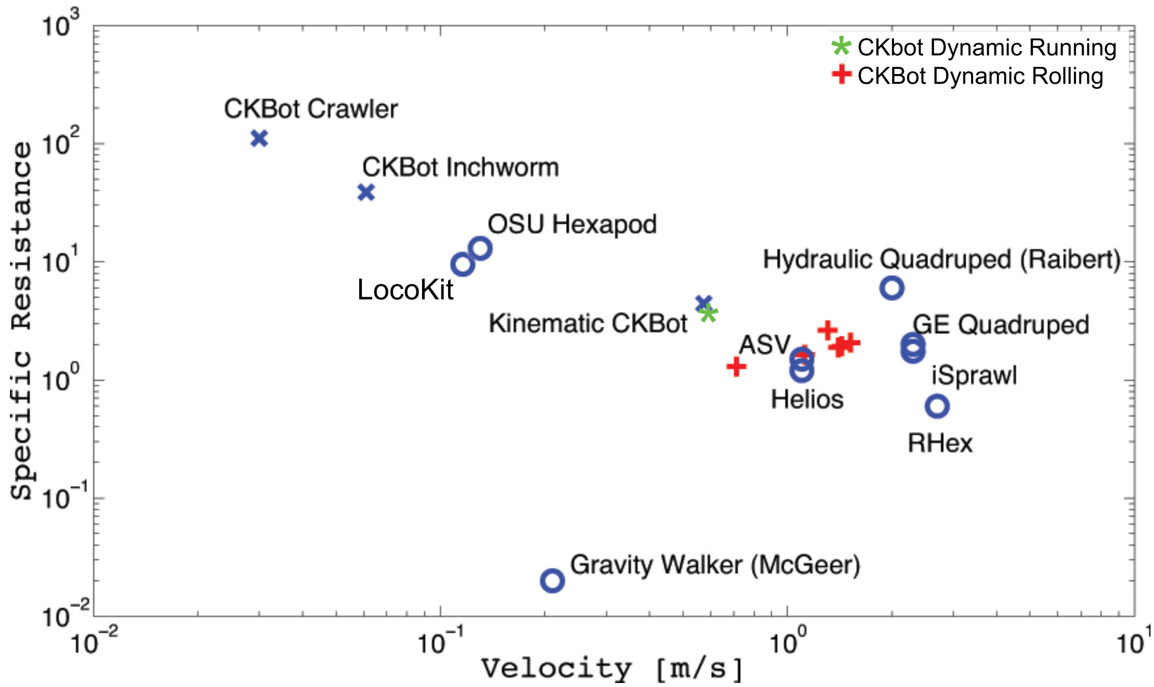


Figure 5.13: Specific resistance vs. velocity for several robotic systems. The data for other systems denoted in blue is included from [39] and [56]

5.7.2 Gait Optimization

We observed that soft legs give 16% higher speed with a 11% lower gait frequency. This difference cannot be accounted for by either frequency-length scaling, which predicts soft legs would be slower, or by estimating ground speed of the feet, which predicts soft legs should be much faster than observed in practice.

Frequency-length scaling

At its most elementary form, speed scales with frequency and leg length. We computed the distance from axis to toe under the assumption of a rigid leg, taking a horizontal displacement equal to leg length, and a height of half a module plus leg height, giving $182\,mm$ and $156\,mm$ for soft and stiff legs (resp.). Thus, the soft leg toe is 16% further away from its attachment point, and we would expect the 16% to appear only with equal gait frequency.

Horizontal sweeping

More precisely, one may model the ground speed of the toe relative to the body using the gait parameters. The product $\rho :=$ (stance duty cycle) $\times$ (period) $\times$ (sweep angle) $\times$ (leg length) should be proportional to the ground speed of the foot in the body frame. The ratio of ρ values is 1.45 in favor of soft legs – suggesting that with the soft legs, the robot should be moving 45% faster than the stiff legs, far above the observed 16% improvement.

Biomechanical approach

Biomechanists [60][1] have developed methods to compare animal gaits despite large variability in scale and body structure. A key observation is the presence of dynamic similarity between running in various organisms, which expresses itself in the existence of a common parameter, the "Froude Number" $\mathfrak{f}$, governing the transition between running and walking ([1] fig 1).

We argue that for our robots, the appropriate leg length to use is not the unloaded hip-height[60], because very substantial compression occurs even when standing still. Since our robot runs in an alternating tripod gait, we use as our vertical length l the equilibrium height of the body when standing on a single tripod of legs. At a weight of $15.7\,N$, this implies $l = 21\,mm$ and $\mathfrak{f} = 3.0$ with soft legs; $l = 35\,mm$, $\mathfrak{f} = 1.3$ with stiff legs. We conclude that with the soft legs, the robot achieves a running gait dynamically similar to animal running ([1] Figure 1), while with the stiff legs, the robot is (dynamically speaking) merely executing a rapid walk.

Our results suggest that in the biologically relevant parameter range of leg stiffness, the performance of dynamically running robots is difficult to anticipate from first principle arguments, or at least those arguments that rely on rigid body models and no-slip contact conditions. That the results are counter-intuitive in laboratory conditions on a flat and uniform floor, speaks to the baffling complexity that must appear as we attempt to design robots to run in uncontrolled environments. The combination of simplicity and flexibility provided by using a modular plaform affords us the opportunity to explore performance optimization through careful empirical study – at least until such a time when we have better theory to guide our designs.

5.7.3 Presence of LLS

Another major finding in this work is the discovery that our robot exhibits LLS dynamics when it was equipped with the set of soft legs. This is demonstrated by the ZMP plots in Figure 5.12 and Table 5.5, which show that the placement of the virtual toe is remarkably different between the soft and stiff legs. The ZMP point moves side to side in the gait when using soft legs while it stays beneath COM in the gait with stiff legs whose relative stiffness is closer to those found in animals. This shows that, in the case of our hexapod, a relative leg stiffness, which is more compliant than those found in animals, provides a fundamentally different mode of locomotion which exhibits LLS-like dynamics.

SLIP has proven useful in informing the mechanical design of impressive robots, such as RHex and iSprawl, and has aided the design of their controllers. LLS could be as useful as SLIP, but it has not been reported yet in any robot. The only other robot that has been shown to exhibit LLS-like dynamics is the horizontal runner developed by Shill et al [91]. This robot was specifically designed for LLS and employs ball bearings on the bottom of its body to glide over the surface. The absence of LLS-based robots may be partly due to the effects of scaling. Animals that have a laterally sprawled posture such as cockroaches and geckoes are typically small. As animals increase in size it becomes harder to support the larger loads. A factor that plausibly enabled LLS in our hexapod robot is its very sprawled posture, unlike such robots such as Rhex, WHegs, Sprawlita or iSprawl.

One particular advantage of imbueing a robot with LLS instead of SLIP it is likely to improve lateral stability in the horizontal plane. Furthermore our robot has the ability to exert forces in the lateral direction. In combination with LLS dynamics, future work might further exploit horizontal dynamics in such behaviors as turning maneuvers.

Chapter 6

Conclusion

In this thesis we have presented a methodology for building physical machines using Modular Robots. The appeal of this method is that it allows researchers to work quickly, building robots fast, typically in a few hours, unlike the more traditional design process, in which a robot is devised from scratch, which can take weeks, months, or even years. This is achieved by modules which function as actuation building blocks. They have a consistent electrical, mechanical and software interface. All version 1 modules have an added on-board controller. A consistent electrical interface consisting of a power and communication bus facilitates ease of wiring. A consistent software interface was created making it easier and faster to program behaviors. We have paid particular attention to the connection mechanism of our building blocks, which not only provides an easy interface between modules, but also connects passive pieces to create structural linkages between the modules. This feature makes it easy to extend and scale the morphology, slightly change parameters in the kinematics, or take them apart entirely and reuse them to build completely different robots.

We explored the modular building methodology in the realm of emergency response and product design, but have chosen to focus this thesis on dynamic locomotion. We believe our approach is of particular interest to this field, because it is at an experimental stage in which theory can not yet fully explain what happens in all the complex ground interactions. Theory alone can not sufficiently drive robot design. Simply being able to rapidly explore and compare different legged robots even though such designs might not be fully optimized for weight or torque is still of incredible value.

But we also went beyond merely developing this methodology and hardware. We used the hardware to find interesting dynamic locomotion properties. We came up with two novel locomotion modes. One is a rolling loop mode which had not been previously analyzed. The second is a legged configuration with a sprawled posture. This hexapod configuration is particularly interesting because legs have two degrees of freedom with only seven actuators, unlike RHex which has only one degree of freedom per leg, and six motors. Additionally, ours is the only robot with intermittent ground contact to report LLS like dynamics.

It is difficult to say concretely, what exactly, constitutes a Modular Robot. One might argue, that simply using hobby servos is itself a way to build robots with modules. A comparison of the time required to build using these two methods is also difficult. Experienced engineers typically re-use components from previously designed robots and put them together to form new ones. Again, it could be argued that this is also a form of modular design. In effect, deciding on the granularity of your building blocks is a trade off between what level of resolution and optimization you desire and how much time you want to spend designing and building. In our case we tried to choose as large a granularity as possible to optimize build time, trading off resolution and performance of the overall robot, while still being able to achieve radically different modes of dynamic locomotion.

Certain advantages of the modular approach are also difficult to measure and is more observed in the design process, for which it is hard to find any specific concrete metric. Under our methodology, not only can we quickly build machines, we can also rapidly change parameters of the design, like scalability and kinematics. In the case of the dynamic rolling loop, we could increase the size of the robot by increasing the number of modules in the loop. In the case of the hexapod configuration, we can easily add to the number of legs like a caterpillar, though that belongs to the realm of future work. Another difficult advantage to quantify ojbectively is that we are able to physically try out different kinematic chains very quickly by varying the number of modules and orientations in the body. For example, it took several different configurations of the hexapod before coming to the realization that two modules were required to drive the undulatory motion of the body in the horizontal plane. This effectively created a four-bar linkage constraint between contralateral legs in a tripod such that they don't slip with respect to each other–an example of how modules are a good way to try out morphologies, even to stumble upon a novel gait by *thinking with your hands*.

In looking forward to the future of dynamic locomotion we can reflect on aviation history. Figure 6.1 shows flight distances that were achieved by various inventors in the period between 1890 and 1909 as they attempted to invent the airplane, reported by Gary Bradshaw [12]. In blue is denoted what he calls the experimental period. Many attempts at flying were made, and we gained increased understanding of the dynamics of flight principles, but progress remained essentially flat. The Wright brothers however started in 1900 and the positive slope shown in green shows that they were able to make steady progress over the next three years. Their stategy had a heavy focus on iteratively prototyping sub-components and testing them empirically. Perhaps the field of dynamic legged locomotion is at a similar point in time. We have come very far in our fundamental understanding of control and mechanical design, but no machine has yet been turned into a product of practical use in the world. Many advances have been made recently in light of global metrics like running speed and efficiency that perchance signal that we are on the verge of developing real running machines of practical utility. One might argue that we are nearing the end of the experimental period with some of the fundamental principles already understood, thus calling for empirical exploration and prototyping following the example of the Wright brothers. We believe this document shows a methodology that is particularly effective in doing this and hope it will encourage others to adopt our methodology of rapidly exploring different morphologies using Modular Robots.

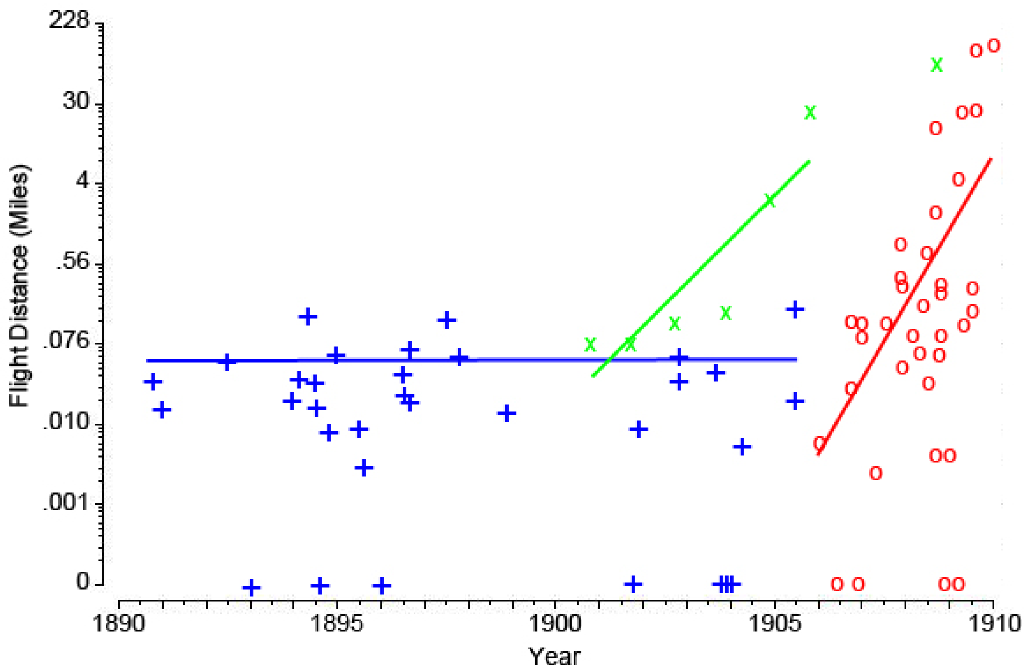


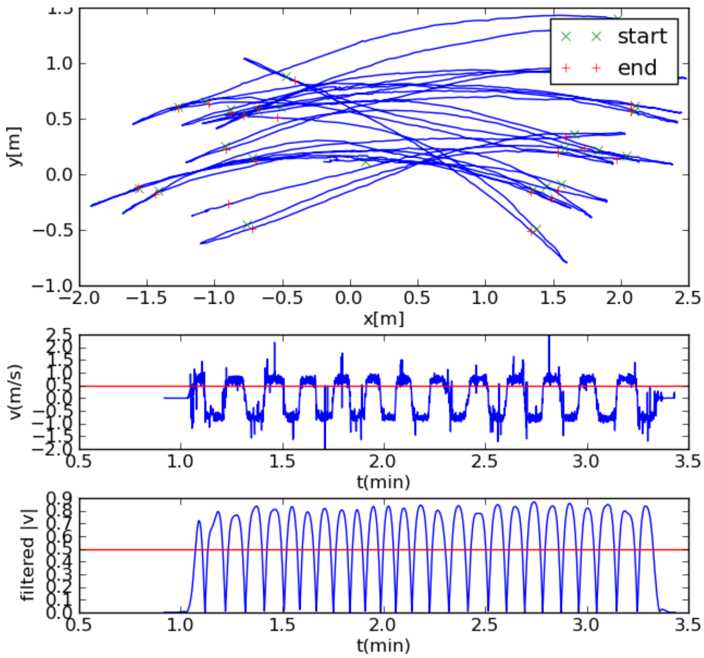
Figure 6.1: Longest flights made by various aircrafts in the period 1890 to 1909. Data taken from [12]

Appendix A

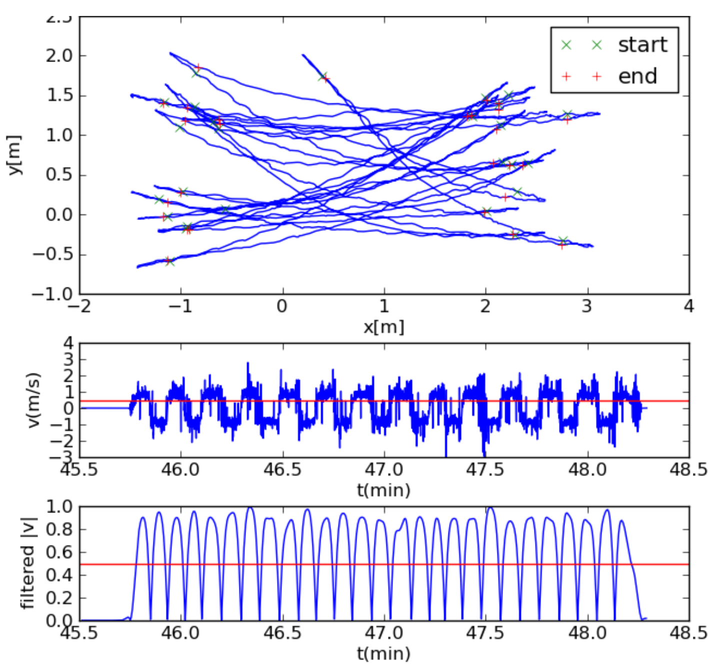
Appendix

	stiff forward	stiff backward	soft forward	stiff backward
mean duty	0.46	0.473	0.46	0.48
std duty	0.01	0.005	0.02	0.01

Table A.1: This table shows the mean and standard deviation of the duty cycle of a single tripod over a stride. It is derived from detected ground contacts of individual legs as shown in Figures A.4 and A.5. Since one step is half a stride, and duty cycle is near half it shows that there is hardly any aerial phase.

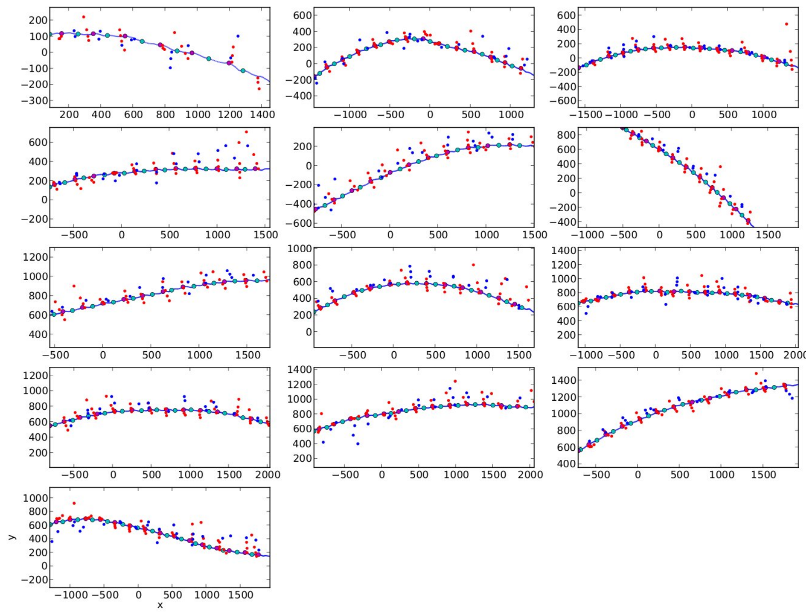


(a) Robot with Soft Legs

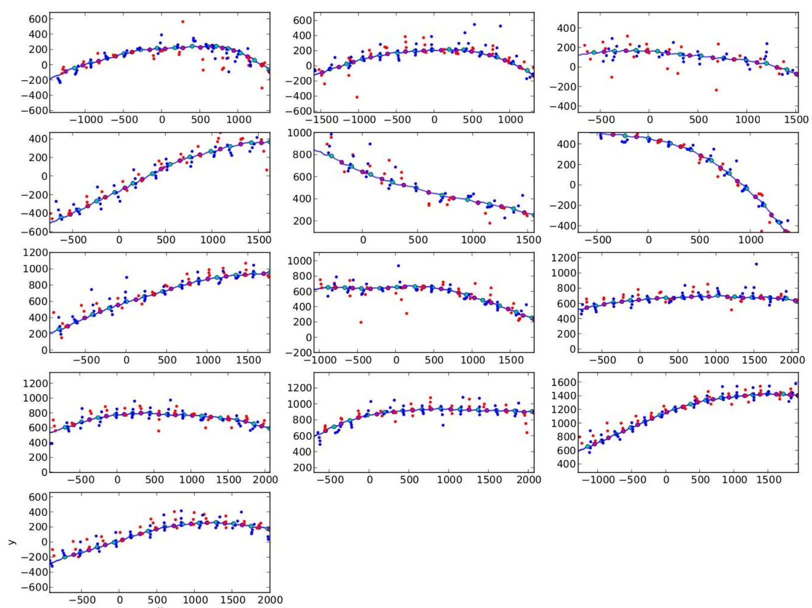


(b) Robot with Stiff Legs

Figure A.1: Segmenting trials. Data is shown for both the stiff and soft legs. The top plots show a top-down view of the robot as it runs back and forth in the world frame. The green x denotes the start of a trial. The red + denotes the end of a trial. Velocity is shown in the middle plots. To detect the start and the end of a trial, we filter the magnitude of the velocity and threshold at 0.5 m/s as shown in the bottom plots.

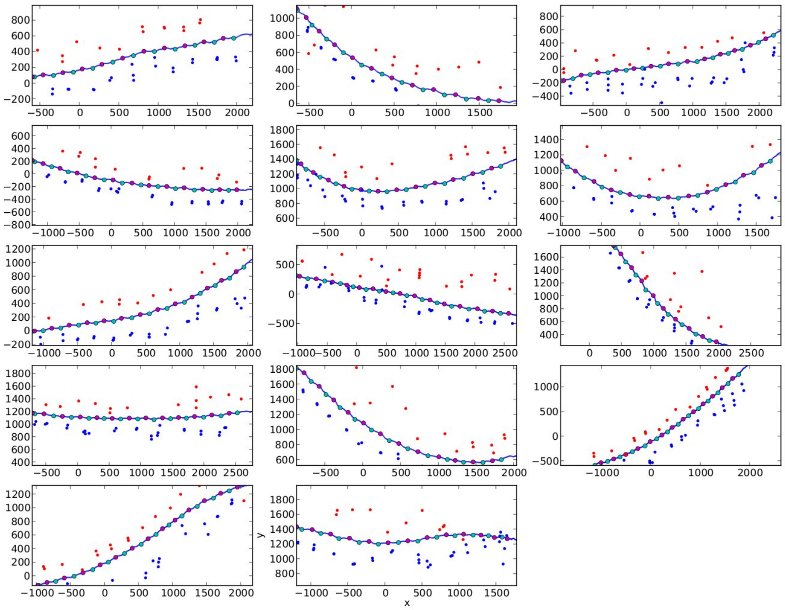


(a) Stiff Forward

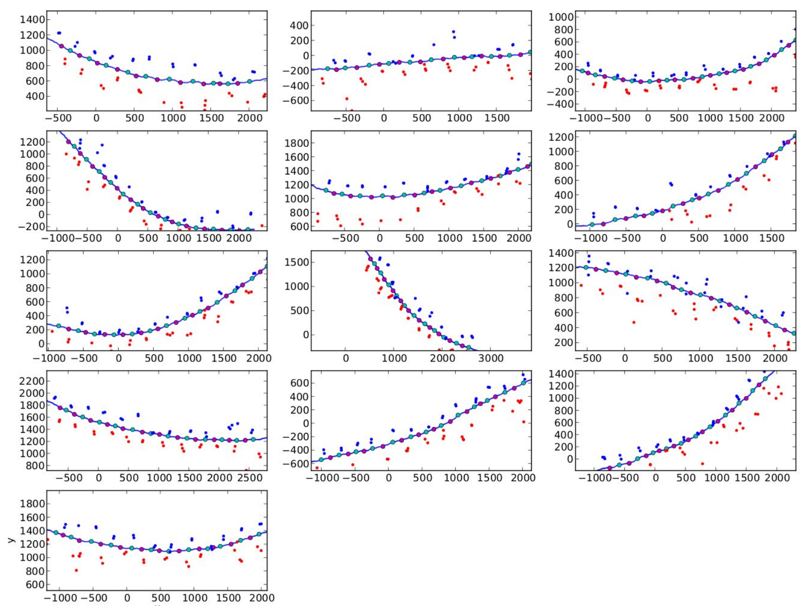


(b) Stiff Backward

Figure A.2: Data of COM and ZMP is shown for stiff legs in the world frame. The COM motion of the robot is shown with a blue line. ZMP are shown in red and blue dots. Corresponding location of COM is shown in cyan and magenta. In the forward gait, the robot is moving from left to right. In the backward gait, the robot is moving from right to left.

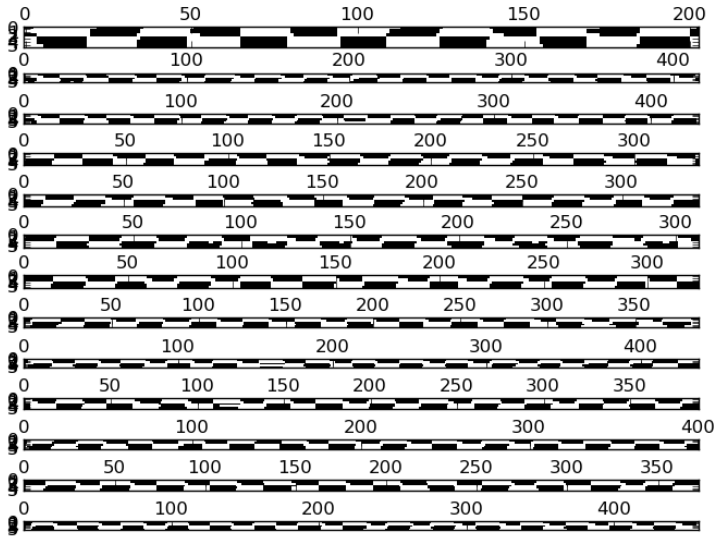


(a) Soft Forward

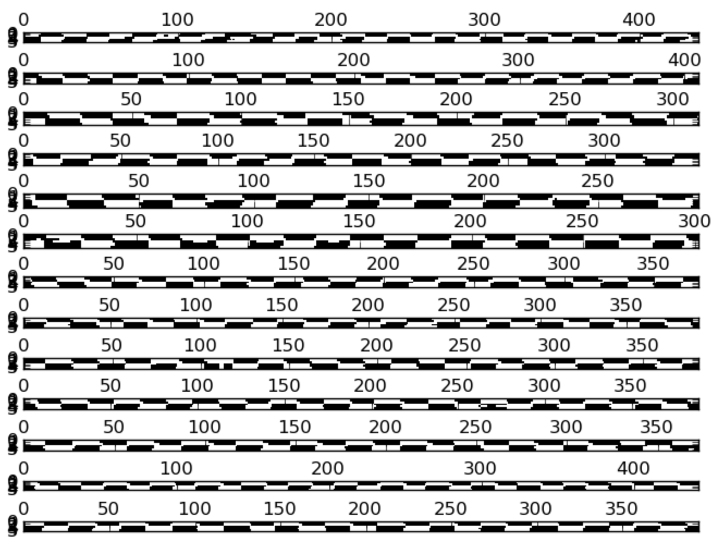


(b) Soft Backward

Figure A.3: Data of COM and ZMP is shown for stiff legs in the world frame. The COM motion of the robot is shown with a blue line. ZMP are shown in red and blue dots. Corresponding location of COM is shown in cyan and magenta. In the forward gait, the robot is moving from left to right. In the backward gait, the robot is moving from right to left.

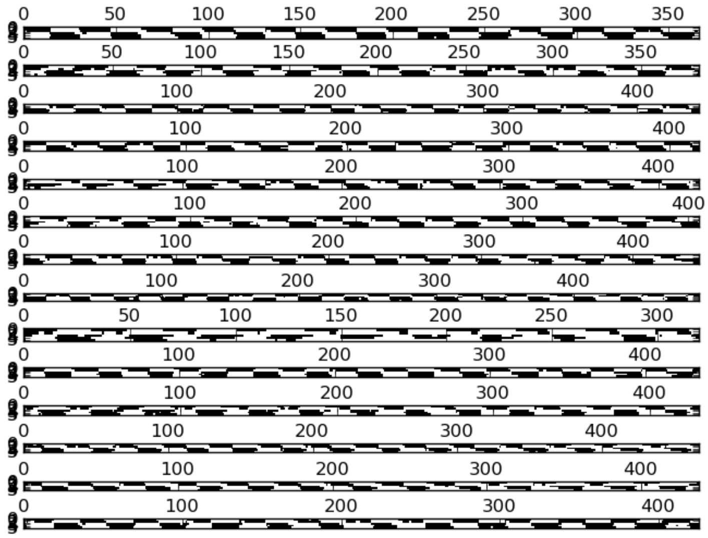


(a) Stiff Forward

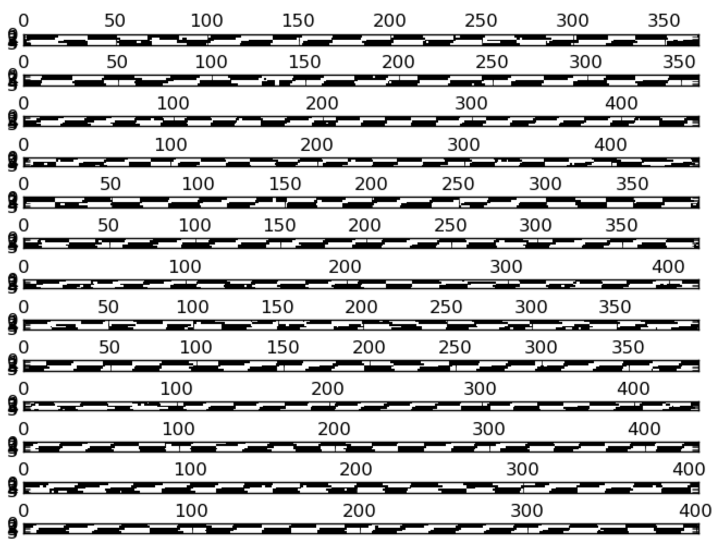


(b) Stiff Backward

Figure A.4: Ground contacts versus time in frames for stiff legs . Black denotes touchdown; white denotes aerial phase. The top three rows denote the left tripod. The bottom three rows denote the right tripod



(a) Soft Forward



(b) Soft Backward

Figure A.5: Ground contacts versus time in frames for soft legs. Black denotes touchdown; white denotes aerial phase. The top three rows denote the left tripod. The bottom three rows denote the right tripod

Bibliography

[1] R Alexander and AS Jayes. A dynamic similarity hypothesis for the gaits of quadrupedal mammals. *Journal of zoology*, 201(1):135–152, 1983.

[2] R. Altendorfer, N. Moore, H. Komsuoglu, M. Buehler, Jr. Brown H. B, D. McMordie, U. Saranli, R. Full, and D. E. Koditschek. Rhex: A biologically inspired hexapod runner. *Autonomous Robots*, 11(3):207–213, 2001.

[3] R Altendorfer, N Moore, H Komsuolu, M Buehler, H B Brown, D McMordie, U Saranli, R J Full, and D E Koditschek. Rhex: A biologically inspired hexapod runner. *Autonomous Robots*, 11(3):207–213, 2001.

[4] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional, 2004.

[5] S. Bhattacharya and S. K. Agrawal. Spherical rolling robot: a design and motion planning studies. *IEEE Transactions on Robotics and Automation*, 16:835–839, 2000.

[6] R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(11-12):1217–27, 1989.

[7] R Blickhan. The spring mass model for running and hopping. *J Biomech*, 22(11-12):1217–1227, 1989.

[8] R Blickhan and R J Full. Similarity in multilegged locomotion - bouncing like a monopode. *J Comp Physiol , A*, 173(5):509–517, 1993.

[9] R. Blickhan and R. J. Full. Similarity in multilegged locomotion: Bouncing like a monopode. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 173(5):509–517, 1993.

[10] R. Blickhan and R. J. Full. Similarity in multilegged locomotion: Bounding like a monopod. *Journal of Comparative Physiology*, 173(5):509–517, 1993.

[11] Walter Boyne. *Dawn Over Kitty Hawk: The Novel of the Wright Brothers*. Forge Books, 2003.

[12] Gary Bradshaw. Wright brothers history: The tale of the airplane. http://www.wright-house.com/wright-brothers/taleplane.html, 2012.

[13] Tim Brown. Creativity and paly. `http://www.ted.com/talks/tim_brown_on_creativity_and_play.html`, November 2008.

[14] Martin Buehler. Dynamic locomotion with one, four and six-legged robots. *Journal of the Robotics Society of Japan*, 20(3), 2002.

[15] Z. Butler, K. Kotay, D. Rus, and K. Tomita. Generic decentralized control for a class of self-reconfigurable robots". In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'02)*, pages 809–815, Washington, D.C., 2002.

[16] A. Castano, W. M. Shen, and P. Will. Conro: Towards deployable robots with inter-robot metamorphic capabilities. *Autonomous Robots Journal*, 8(3):309–324, July 2000.

[17] G. A. Cavagna, N. C. Heglund, and Taylor C. R. Mechanical work in terrestrial locomotion: Two basic mechanisms for minimizing energy expenditure. *American Journal of Physiology*, 233, 1977.

[18] 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*, 21(10), 2002.

[19] J G Cham, J K Karpick, and M R Cutkosky. Stride period adaptation of a biomimetic running hexapod. *Int J Rob Res*, 23(2):141–153, 2004.

[20] G. S. Chirikjian. Kinematics of a metamorphic robotic system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1452–1457, 1996.

[21] G.S. Chirikjian and J. W. Burdick. A modal approach to hyper-redundant manipulator kinematics. *IEEE Transactions on Robotics and Automation*, 10(3):343–354, June 1994.

[22] Octavio Gonzlez Claire T. Farley. Leg stiffness and stride frequency in human running. *Journal of biomechanics*, 29(2):181–186, 1996.

[23] Mike Cohn. *Agile Estimating and Planning*. Prentice Hall, 2005.

[24] Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

[25] Steven H. Collins, Andy Ruina, Russ Tedrake, and Martjin Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, 2005.

[26] Steven H. Collins, Martijn Wisse, and Andy Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *International Journal of Robotics Research*, 20(7):607–615, 2001.

[27] M H Dickinson, C T Farley, R J Full, M A R Koehl, R Kram, and S Lehman. How animals move: an integrative view. *Science*, 288:100–106, 2000.

[28] Digi. Need wireless? choose xbee. `http://www.digi.com/xbee/`, June 2012.

[29] James Dyson. *Against the Odds:An Autobiography.* Texere, 2003.

[30] C. T. Farley, J. Glasheen, and T. A. McMahon. Running springs - speed and animal size. *Journal of Experimental Biology*, 185:71–86, 1993.

[31] M. Farsi, K. Ratcliff, and M. Barbosa. An overview of controller area network. *Computing and Control Engineering Journal*, 1999.

[32] Daniel P. Ferris, Micky Louie, and Claire T. Farley. Running in the real world: Adjusting leg stiffness for different surfaces. *Proceedings: Biological Sciences*, 265(1400):989–994, 1998.

[33] James Dyson Foundation. Build it, break it. `http://media.dyson.com/downloads/JDF/Poster_9_Build_It_Break_It.pdf`, March 2012.

[34] T. Fukuda and Y. Kawauchi. Cellular robotic system (cebot) as one of the realization of self-organizing intelligent universal manipulator. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 662 –667 vol.1, May 1990.

[35] R. J. Full, R. Blickhan, and L. H. Ting. Leg design in hexapedal runners. *Journal of Experimental Biology*, 158(UL):369–390, 1991.

[36] R. J. Full and D. E. Koditschek. Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202(23):3325–3332, 1999.

[37] Keving Galloway. *Passive Variable Compliance for Dynamic Legged Robots.* PhD thesis, University of Pennsylvania, 2010.

[38] D. Gomez-Ibanez, E. Stump, B. Grocholsky, V. Kumar, and C.J. Taylor. The robotics bus: A local communications bus for robots. In *Proc. of the Society of Photo-Optical Instrumentation Engineers*, 2004.

[39] Pedro Gregorio, Mojtaba Ahmadi, and Martin Buehler. Design, control and energetics of an electrically actuated legged robot. *IEEE Transactions on Systems, Man and Cybernetics*, 27(4):626–634, August 1997.

[40] A. Halme, T. Schonberg, and Y. Wang. Motion control of a spherical mobile robot. In *Proceedings of 4th IEEE International Workshop on Advanced Motion Control - AMC '96 - MIE*, pages 259–264, Mie, Japan, 1996.

[41] Masato Hirose and Kenichi Ogawa. Honda humanoid robots development. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):11–19, 2007.

[42] S. Hirose and A. Morishima. Design and control of a mobile robot with an articulated body. *The International Journal of Robotics Research*, 9(2):99–114, April 1990.

[43] Philip Holmes, Robert J. Full, Dan Koditschek, and John Guckenheimer. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Rev.*, 48(2):207–304, February 2006.

[44] James Hopkins, Spranklin Brent, and Gupta Stayandra. A survey of snake-inspired robot designs. *Bioinspiration and Biomimetics*, 4(2), 2009.

[45] Peter Jake. *Visions of a Flying Machine*. Smithsonian Books, 1997.

[46] A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, and K. Tomita. Automatic locomotion design and experiments for a modular robotic system. *IEEE/ASME Transactions on Mechatronics*, 10(3):314–325, June 2005.

[47] A. Kamimura, H. Kurokawa, E. Yoshida, K. Tomita, S. Kokaji, and S. Murata. Distributed adaptive locomotion by a modular robotic system, m-tran ii. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2370–2377, 2004.

[48] Akiya Kamimura, Satoshi Murata, Eiichi Yoshida, Haruhisa Kurokawa, Kohji Tomita, and Shigeru Kokaji. Self-reconfigurable modular robot - experiments on reconfiguration and locomotion. In *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS200)*, pages 606–612, Maui, Hawaii, October 2001.

[49] D. C. Kar. Design of statically stable walking robot: A review. *Journal of Robotic Systems*, 20(11):671–686, 2003.

[50] Tom Kelly, Jonathan Littman, and Tom Peters. *The Art of Innovation: Lessons in Creativity from IDEO, America's Leading Design Firm*. Crown Business, 2001.

[51] Sangbae Kim. isprawl: Design and tuning for high-speed autonomous open-loop running. *International Journal of Robotic Research*, 25(9):903–912, September 2006.

[52] H. Kimura, S. Akiyama, and K. Sakurama. Realization of dynamic walking and running of the quadruped using neural oscillator. *Autonomous Robots*, 7(3):247,258, 1999.

[53] E. Klavins, H. Komsuoglu, R. J. Full, and D. E Koditschek. *The Role of Reflexes Versus Central Pattern Generators in Dynamical Legged Locomotion*, chapter Neurotechnology for Biomimetic Robots, pages 351–382. MIT Press, 2002.

[54] A. D. Kuo, J. M. Donelan, and A. Ruina. Energetic consequences of walking like an inverted pendulum: Step-to-step transitions. *Exercise and Sport Science Review*, 33(2), 2005.

[55] Haruhisa Kurokawa, Kohji Tomita, Akiya Kamimura, Shigeru Kokaji, Takashi Hasuo, and Satoshi Murata. Self-reconfigurable modular robot m-tran: distributed control and communication. In *Proceedings of the 1st international*

conference on Robot communication and coordination, RoboComm '07, pages 21:1–21:7, Piscataway, NJ, USA, 2007. IEEE Press.

[56] J C Larsen. Locomotion through morphosis: development of the modular robotic toolkit - locokit. Master's thesis, South Denmark University, 2011.

[57] Woo Ho Lee and Arthur Sanderson. Dynamic rolling of modular robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, April 2000.

[58] T. Matsuda and S. Murata. Stiffness distribution control - locomotion of closed link robot with mechanical softness. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1491–1498, Orlando, Florida, May 2006.

[59] Tad McGeer. Passive dynamic walking. *Int. J. of Robotics Research*, 9(2):62–82, April 1990.

[60] T A McMahon and G C Cheng. The mechanics of running - how does stiffness couple with speed. *J Biomech*, 23(Suppl. 1):65–78, 1990.

[61] Microchip. Microchip pic18f2585/2680/4585/4680 data sheet. http://ww1.microchip.com/downloads/en/DeviceDoc/39625c.pdf, 2012.

[62] R. Mukherjee, M. A. Minor, and J. T. Pukrushpan. Simple motion planning strategies for spherobot: a spherical mobile robot. In *Proceedings of 1999 Conference on Decision and Control*, volume 3, pages 2132–2137, Phoenix, AZ, 1999.

[63] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji. M-tran: Self-reconfigurable modular robotic system. *IEEE/ASME Transactions on Mechatronics*, 7(4):431–441, December 2002.

[64] Y Or and E Rimon. Analytic characterization of a class of three-contact frictional equilibrium postures in three-dimensional gravitational environments. *IJRR*, 29(1):3–22, 2010.

[65] E. Østergaard. *Distributed Control of the ATRON Self-Reconfigurable Robot*. University of Southern Denmark, Odense, 2004.

[66] Xerox Parc. Modular robotics. http://www2.parc.com/spl/projects/modrobots/, 2012.

[67] Michael Park, Sachin Chitta, Alex Teichman, and Mark Yim. Automatic configuration recognition methods in modular robots. *The International Journal of Robotics Research*, 27(3-4):403–421, 2008.

[68] C. C. Phipps and M. A. Minor. Introducing the hex-a-ball, a hybrid locomotion terrain adaptive walking and rolling robot. In *CLAWAR*, volume 3, London, UK, 2005.

[69] Open Prosthetics Project. The open prosthetics project: An initiative of shared design alliance. `http://openprosthetics.org`, March 2012.

[70] Hayes Solos Raffle, Amanda J. Parkes, and Hiroshi Ishii. Topobo: a constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 647–654, New York, NY, USA, 2004. ACM.

[71] H.M. Raibert. *Legged Robots that Balance*. The MIT Press, Cambridge, MA, 1985.

[72] Marc Raibert. Legged robots. *Communications of the ACM*, 29(6), 1986.

[73] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, and the Big-Dog Team. Bigdog, the rough-terrain quadruped robot. In *Proceedings of the*

17th World Congress The International Federation of of Automatic Control, Seoul, Korea, July 2010.

[74] Marc H. Raibert. *Legged robots that balance.* MIT Press series in artificial intelligence. MIT Press, Cambridge, Mass., 1986.

[75] Marc H. Raibert, H. Benjamin Brown, and Michael Chepponis. Experiments in balance with a 3d one-legged hopping machine. *The International Journal of Robotics Research*, 3(2):75–92, 1984.

[76] S Revzen and J M Guckenheimer. Estimating the phase of synchronized oscillators. *Phys Rev E*, 78(5):051907, 2008.

[77] S Revzen, J Sastra, N Eckenstein, and Yim. Ckbot platform for the icra 2010 planetary challenge. In *Workshop "Modular Robots: The State of the Art", Proceedings of IEEE ICRA conference*, pages 11–12. IEEE, 2010.

[78] S. Revzen, J. Sastra, N. Eckenstien, and M. Yim. Ckbot platform for the icra 2010 planetary challenge. In *Proceedings of the IEEE 2010 International Conference on Robotics and Automation Workshop. Modular Robots: The State of the Art*, 2010.

[79] Pololu Robotics and Electronics. Mini maestro 12-channel usb servo controller. http://www.pololu.com/catalog/product/1352, 2012.

[80] Robotis. Robotis website. `http:robotis.com`.

[81] B. Salemi, M. Moll, and W.-M. Shen. Superbot: A deployable, multi-functional, and modular self-reconfigurable robotic system. In *P IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.

[82] U. Saranli and D.E. Koditschek. Template based control of hexapedal running. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 1, pages 1374–1379, Sept. 2003.

[83] J Sastra, W G Bernal-Heredia, J Clark, and M Yim. A biologically-inspired dynamic legged locomotion with a modular reconfigurable robot. In *ASME Dynamic Systems and Control Conference*, October 2008.

[84] J Schmitt, M Garcia, R C Razo, P Holmes, and R J Full. Dynamics and stability of legged locomotion in the horizontal plane: a test case using insects. *Biol Cybern*, 86(5):343–353, 2002.

[85] J. Schmitt, M. Garcia, R.C. Razo, P. Holmes, and R.J. Full. Dynamics and stability of legged locomotion in the horizontal plane: a test case using insects. *Biological Cybernetics*, 86, 2002.

[86] J. Schmitt and P. Holmes. Dynamics and stability of legged locomotion in the horizontal plane – ii. application. *Biological Cybernetics*, 83, 2000.

[87] J Schmitt and P Holmes. Mechanical models for insect locomotion: stability and parameter studies. *Phys D: Nonlinear Phenom*, 156(1-2):139–168, 2001.

[88] Standford D School. Method prototype to test. `http://dschool.stanford.edu/wp-content/themes/dschool/method-cards/prototype-to-test.pdf`, June 2012.

[89] Andre Seyfarth, Hartmut Geyer, and Hugh Herr. Swing-leg retraction: a simple control model for stable running. *J Exp Biol*, 206(15):2547–2555, 2003.

[90] Wei-Min Shen, Maks Krivokon, Harris Chiu, Jacob Everist, Michael Rubenstein, and Jagadesh Venkatesh. Multimode locomotion via superbot robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2552–2557, Orlando, Florida, May 2006.

[91] J. Shill, B. Miller, J. Schmitt, and J. Clark. Design of a dynamically stable horizontal plane runner. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automations, Anchorage, Alaska, USA*, 2010.

[92] B. Shores and M. Minor. Design, kinematic analysis, and quasi-steady control of a morphic rolling disk biped climbing robot. In *IEEE Int'l Conf. on Rob. and Autom*, volume 3, pages 2732–37, Barcelona, Spain, 2005.

[93] Y. Sugiyama, A. Shiotsu, M. Yamanaka, and S. Hirai. Circular/spherical robots for crawling and jumping. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, April 2005.

[94] U.Saranli, M. Buehler, and D. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 7(7):616–631, July 2001.

[95] M. Vukobratoic, B. Borovac, and D. Surdilovic. Zero-moment point – proper interpretation. *International Journal of Humanoid Robotics*, 1, 2004.

[96] J D Weingarten, G A D Lopes, M Buehler, R E Groff, and D E Koditschek. Automated gait adaptation for legged robots. In *Robotics and Automation, 2004. Proceedings. IEEE International Conference on*, volume 3, pages 2153–2158, 2004.

[97] Joel D. Weingarten, Gabriel D. Lopes, Martin Buehler, Richard E. Groff, and Daniel E. Koditschek. Automated gait adaptation for legged robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 2004.

[98] Cornell Wright, Aaron Johnson, Aaron Peck, Zachary Mccord, Allison Naaktgeboren, Philip Gianfortoni, Manuel Gonzalez-rivero, Ross Hatton, and Howie Choset. Design of a modular snake robot. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2008.

[99] Tom Wujec. The marshmallow challenge. `http://marshmallowchallenge.com/TED_Talk_files/TED2010_Tom_Wujec_Marshmallow_Challenge_Web_Version.pdf`, June 2010.

[100] M. Yim. *Locomotion With a Unit-Modular Reconfigurable Robot*. Stanford University, Palo Alto, CA, 1994.

[101] M. Yim. *Locomotion With A Unit-Modular Reconfigurable Robot*. PhD thesis, Stanford University, Palo Alto, CA, December 1994.

[102] M. Yim. Planetary contingency [education]. *Robotics Automation Magazine, IEEE*, 15(4):14–16, dec. 2008.

[103] M. Yim, D. G. Duff, and K. D. Roufas. Polybot: a modular reconfigurable robot. In *P IEEE International Conference on Robotics and Automation (ICRA)*, pages 514–520, San Francisco, CA, April 2000.

[104] M. Yim, C. Eldershaw, Y. Zhang, and D. G. Duff. Limbless conforming gaits with modular robots. In *Proc. of the International Symposium on Experimental Robotics*, Singapore, 2004.

[105] M. Yim, R. Hinden, C. Conley, C. K. Wang, K. D. Roufas, and C. Eldershaw. Open loop climbing with modular robots. In *Video Proc. of the IEEE Intl. Conf. on Robotics and Automation(ICRA)*, Orlando, FL, 2006.

[106] M. Yim, S. B. Homans, and K. D. Roufas. Climbing with snake-like robots. In *IFAC Workshop on Mobile Robot Technology*, Jejudo, Korea, MAY 2001.

[107] M. Yim, Wei-Min Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G.S. Chirikjian. Modular self-reconfigurable robot systems [grand challenges of robotics]. *Robotics Automation Magazine, IEEE*, 14(1):43 –52, march 2007.

[108] M. Yim, P. White, M. Park, and J. Sastra. *Encyclopedia of Complexity and System Science*. Springer, New York, 2009.

[109] Mark Yim, Jimmy Sastra, and Simon Kim. The robot etudes. In *2011 ICRA Workshop on Robots and Art*, 2011.

[110] Y. Zhang, C. Eldershaw, M. Yim, K. D. Roufas, and D. G. Duff. A platform for studying locomotion systems: Modular self-reconfigurable robots. In *Proc. of NIST workshop on Performance Metrics for Intelligent Systems (PerMIS)*, pages 893–899, Gaithersburg, MD, August 2002.

[111] V. Zykov, A. Chan, and H. Lipson. Molecubes: An open-source modular robotics kit. In *International Conference on Intelligent Robots and Systems (IROS) Workshop on Self-Reconfigurable Robots*, 2007.

[112] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson. Self-reproducing machines. *Nature*, 435:163–164, 2005.