LOCOMOTION WITH A UNIT-MODULAR RECONFIGURABLE ROBOT

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Mark Yim
December 1994

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Jean-Claude Latombe
(Principal Adviser)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Mark Cutkosky

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Oussama Khatib

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies

Abstract

A unit-modular robot is a robot that is composed of modules that are all identical. In this thesis we study the design and control of unit-modular dynamically reconfigurable robots. This is based upon the design and construction of a robot called Polypod. We further choose statically stable locomotion as the task domain to evaluate the design and control strategy. The result is the creation of a number of unique locomotion modes.

The exciting aspect about a modular robot like Polypod is that it does not only describe one robot, but also presents the building blocks from which many different types of robots can be formed. Dynamic reconfigurability adds a new dimension to the capabilities of the robot.

To gain insight into these capabilities in the domain of locomotion, we first build a general, functional taxonomy of locomotion modes. We show that Polypod is capable of generating all classes of statically stable locomotion, a feature unique to Polypod. Next, we propose methods to evaluate vehicles under different operating conditions such as different terrain conditions. We then evaluate and compare each mode of locomotion on Polypod within each class. This study leads to interesting insights into the general characteristics of the corresponding classes of locomotion.

Finally, since more modules are expected to increase robot capability, it is important to examine the limit to the number of modules that can be put together in a useful form. We answer this question by investigating the issues of structural stability, actuator strength, computation and control requirements.

Acknowledgments

First, I would like to thank Professor Jean-Claude Latombe for allowing me to take on this project and advising and supporting me throughout the course of my research. I was extremely fortunate to have someone who knew how to encourage creativity and inspire excellence in research.

I would also like to thank the other members of my reading committee Professors Mark Cutkosky and Oussama Khatib, and the members of my defense committee Professors Bernard Roth and Greg Kovacs. Professor Cutkosky was also my program advisor and gave insightful suggestions throughout my time at Stanford. I also thank Professor Khatib for his unwavering advocacy.

Many thanks must also go to Professor Ed Carryer and his Smart Product Design Lab class which in large part led to the creation of Polypod and thus this thesis. Likewise the student machine shop and its caretakers were invaluable in the prototyping and construction phase of Polypod. I am also grateful to Meggy Gotuaco, Rocky Kahn and Dan Arquilevich for their help in the design and construction of Polypod.

Special thanks go to my fellow students at Stanford: Wonyun Choi, Sanford Dickert, Bob Holmberg, Lydia Kavraki, Yotto Koga, Tsai-Yen Li, Sean Quinlan, Joaquin Salas, David Williams and everybody in the robotics lab.

Finally, I'd like to thank my family for their support and encouragement. Especially my wife Laura who not only helped to produce this work (first suggested reconfigurability to me) but produced our son Justin during the beginning of my thesis work and will have another baby soon after this thesis is completed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem statement

In the past decade, there has been some work on modularity in robotics [Wurst 1986][Tesar 1989][Cohen 1990] with the goal of making more versatile, easily adaptable manipulator arms. More recently there has been some work on adding reconfigurability to modular robotic systems. Again the goal was to make even more versatile autonomous robot systems.

In this thesis we wish to explore the versatility of reconfigurable modular robotic systems, the initial goal being to determine how versatile such a system can be. To make the problem manageable, the application domain was restricted to statically stable locomotion.

To study versatility in a chosen domain, we must examine that domain in general. While there has been work on specific means of statically stable locomotion, a generalized study of locomotion has had little attention.

The end result of this dissertation is the creation of a taxonomy of the different kinds locomotion possible, and the design, construction and analysis of a robot called Polypod that can achieve them.

1.2 Locomotion

Robot locomotion has been studied quite extensively, though each study was typically designed for one type of robot that locomotes in one particular way in one particular type of environment. For example, many three-wheeled synchro-drive (all wheels are driven synchronously and turned synchronously) robots such as those built by Nomadic Technologies, Real World Interface, and Denning Robotics have been used to study path planning and obstacle avoidance in an indoor setting. The CMU Ambler is a very different, large six-legged robot built to traverse Martian like terrain. While the three-wheeled robots can not traverse over a one cubic foot boulder, the Ambler cannot traverse through a standard 10 foot wide building hallway. Size is just one factor in determining the suitability of a form of locomotion for a given terrain. Different areas of terrain may pose different constraints to locomotion.

A reconfigurable robot would be able to reconfigure itself to use different modes of locomotion to adapt to different terrains. This adaptability can be very important. A Mars exploration mission is one example. If a robot sent to Mars is not well suited to the type of terrain in which it lands, the mission could be an expensive failure.

To illustrate the potential difficulties consider Dante II, an eight-legged robot that uses statically stable locomotion. Dante made news history in August 1994 by descending to the bottom of Mt. Spur, an active volcano in Alaska, something that had never been done. One of the missions secondary goals was as a preparation for a possible Mars or moon exploration mission. There are many interesting locomotive issues in this event. The terrain was extremely difficult including steep slopes, large boulders, falling boulders, soft soil, and harsh temperatures. As the robot returned from the bottom, it fell over onto its back helpless. This occured despite being teleoperated and having the aid of a support tether. This mission followed Dante I, another eight-legged robot which was to descend into Mt. Erebus, a volcano in the Antarctic. Dante I broke down after descending 21 feet past the rim in January 1993.

1.2.1 Related Work on Locomotion

Our study of locomotion comes in two parts, a generalized study of statically stable locomotion by the creation of a taxonomy of locomotion, and an analysis of some of the parameters that are used to evaluate the suitability of locomotion modes for different locomotion tasks.

There has been a great deal of study on *animal* locomotion. The text by J. Gray [Gray 1968] has served as a guide for many researchers on this subject. The main purpose here, as with other texts on locomotion, was to study each type of locomotion separately, and not to generalize. In these animal locomotion texts, the implicit classification is based on animal type, i.e. invertebrates, vertebrates, mammals etc. In our taxonomy we will use a more mechanistic view since we wish to include mechanisms as well as animals.

For vehicle locomotion an analogous book to Gray's text has been written by Bekkar [Bekkar 1969]. There has been a significant amount of work in studying wheeled and tracked locomotion on rough terrain as there are over 900 references in Bekkar's book. In this sense his book also serves as a survey of the state of the art in mechanical locomotion in 1969. Bekkar also briefly covers legged locomotion and screw locomotion although he makes little attempt at generalization.

1.3 Review of Modularity and Reconfigurability

Very often the terms *modular* and *reconfigurable* are meant to describe different things. Here we make our meaning clear.

Modularity: Modularity is defined as the characteristic of being constructed of a set of standardized components which usually can be interchanged. We wish to examine unit-modularity. Unit-modular describes a system which is composed of a single type of repeated component.

As implied in Figure 1.1, systems may have varying degrees of unit-modularity. Single type modular systems would be at one end of the scale (true unit-modular), systems with two types less unit-modular, etc.

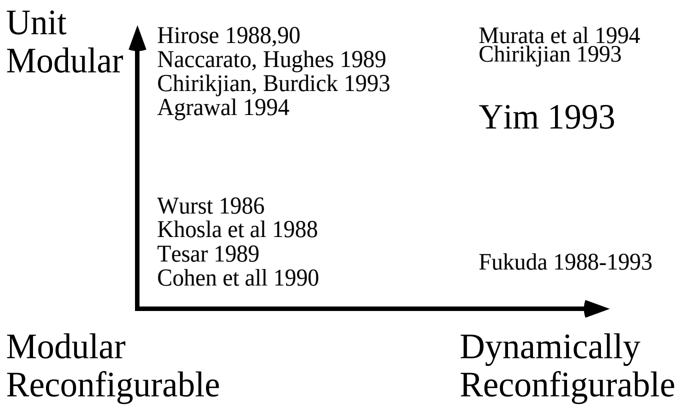


Figure 1.1: Relative modularity and reconfigurability

For an autonomous robot, a system with at most one of each type of module should contain all the components needed to be autonomous; this minimum set could be an autonomous robot in itself.

Reconfigurability: Reconfigurability is a nebulous term which has often been used to mean different things in robotics. The three most common definitions are as follows:

- the ability to attain the same end-effector positions in manipulators with grossly different joint positions, for example, elbow-up and elbow-down configurations in articulated arms
- the ability to rearrange a robot's physical components
- the ability of the robot to rearrange its own physical components.

We will be using the last definition. *Dynamically reconfigurable* in this sense means the robot may reconfigure itself "on-the-fly." Its opposite is *manually reconfigurable* which means another agent (human or robot) must reconfigure the robot.

As with modularity, we also show in Figure 1.1 varying degrees of dynamic reconfigurability. Fully autonomous reconfiguration at one end, and human assembled at the other.

1.3.1 Reconfigurable Modular Manipulators

Typical reconfigurable modular manipulators [Wurst 1986][Tesar 1989][Cohen 1990] consist of rigid link modules of varying lengths and actuator modules with varying degrees of freedom (DOF). Khosla et. al. has developed a modular manually reconfigurable manipulator arm called RMMS [Schmitz 1988]. An advantage over traditional manipulator arms is that as a user rearranges the modules for new tasks, the dynamic parameters of the links are automatically generated.

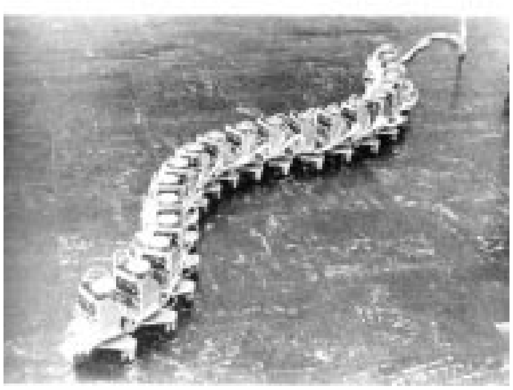
One example of a reconfigurable module in today's industrial manipulators is the quick-change end-effector. Quick0change end-effectors have been used in industrial manipulators for many years. The end-effector of most robots has at least one DOF like a gripper that opens and closes, or a screw driving mechanism that turns on or off. Quick-change end-effector systems consist of a set of docks which hold a number of different end-effectors, and a quick-change wrist mounted on the manipulator arm. The arm is then free to pick up an end-effector, use it, then drop it off at a dock and pick up another one.

In this sense the end-effector can be considered a modular link to the arm. It shares the characteristics of common connection mechanisms between modules and is dynamically reconfigurable. This is the most common use of a modular dynamically reconfigurable robot, although it is quite restricted.

1.3.2 Unit-Modular Systems

Several systems which are almost unit-modular have also been developed. The key element missing in all of these systems is that the are not autonomous; computation and power requirements are supplied off-board. Reconfigurability of these systems is not part of the design.

Active Chord Mechanism (ACM): Hirose was an early developer of many novel mobile robots, most notably his snake-like robots [Hirose 1986-1992]. These snake-like robots, as shown in Figure 1.2, are inherently modular systems. Hirose studied the locomotion of these robots attempting to mimic the locomotion of snakes.



from [Hirose 1990]

Figure 1.2: Active Chord Mechanism, ACM III snake-like robot

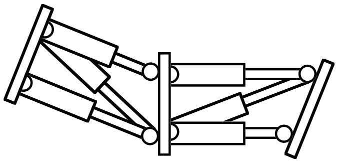
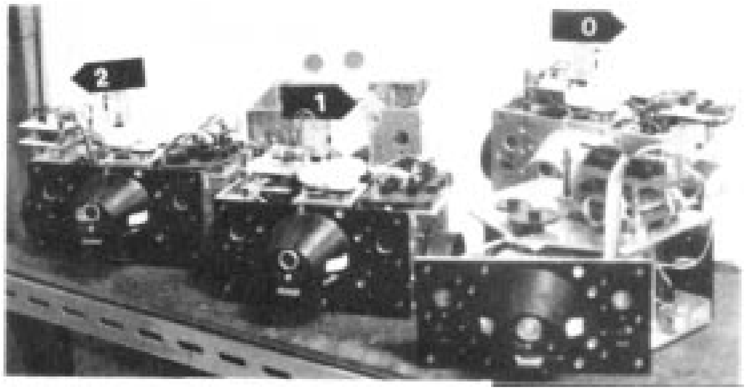


Figure 1.3: Variable Geometry Truss (VGT)

Variable Geometry Truss: The Variable Geometry Truss or VGT, is a truss structure with several of the struts having actuated lengths.

Each truss can be viewed as a module. VGT's were originally studied for applications to large space structures [Atluri 1988] in which typically they controlled the structural resonances. More recently, VGT's have been applied to manipulators as in Figure 1.3.

Chirikjian and Burdick used a VGT similar to the one pictured [Chirikjian 1993]. Initial work by Chirikjian and Burdick was on highly redundant manipulators which they call "Hyper-redundant." [Chirikjian 1991,1993a]. They pioneered an inverse kinematics method for hyper-redundant manipulators with some extensions to snake-like locomotion. This work is important to unit-modular robots as these robots tend to have a large number of DOF.



from [Fukuda 1989]

Figure 1.4: Mobile CEBOT

Medical Robots: Agrawal has designed and built a 3 DOF modular robot for analyzing the human spinal chord in [Agrawal 1994]. Each module can simulate the motions of a bone in the spinal chord. The range of motion is limited just as the human spine is limited. Reconfigurability was not a goal, as the robot only forms serial chains as do the Hirose's ACM robots.

1.3.3 Modular and Dynamically Reconfigurable Systems

Cellular Robots: By far the most prolific work in reconfigurable modular robots has been that of Fukuda et al. In biology, cells can be viewed as one module in highly complex organisms. Fukuda thus coined the term cellular robotics or CEBOT [Fukuda 1986] for his modular reconfigurable systems.

They studied the connection and docking of mobile cells in [Fukuda 1989] and proposed several methods for the organization of these cells including genetic algorithms [Fukuda 1992].

Fukuda and co-workers built and experimented with a separate cellular manipulator [Fukuda 1990] shown in Figure 1.5. These studies included finding optimal arrangements of cells for given manipulator tasks.

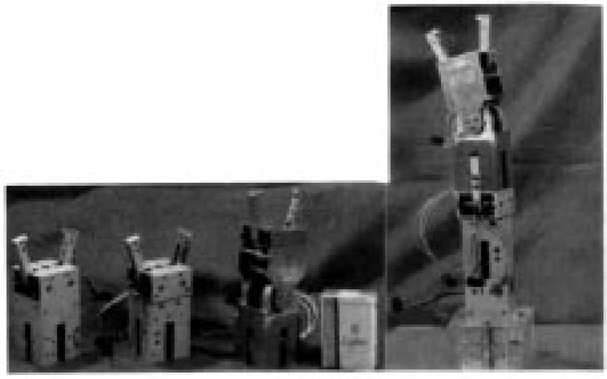
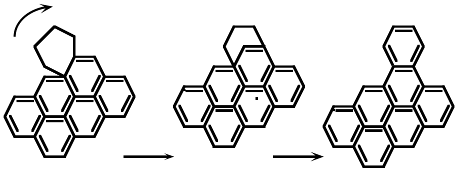


Figure 1.5: Manipulator CEBOT



based on [Chirikjian 1993]

Figure 1.6: Metamorphosing robot

Metamorphosing Robot: While the CEBOT's were not unit-modular, the following "metamorphosing" robot is. Soon after the development of Polypod but independently, Chirikjian proposed a dynamically reconfigurable unit-modular robot which he calls metamorphosing robots. Each module is a planar hexagonal shaped robot with three DOF as shown in Figure 1.6. Each side of the hexagon can connect to a side on another hexagon of the opposite polarity. Cells move by attaching and detaching on neighboring cells "rolling" over each other. Global motions are obtained by this rearrangement of cells.

This work differs from Polypod as it is restricted to configurations in 2D, and has yet to be implemented.

1.4 Characteristics of Reconfigurable Modular Robots

There are many advantages to having a system built up of unit-modules. These include:

- *Manufacturability:* Reducing the number of operations for individual parts simplifies manufacturing, making them easier and cheaper to build. Repeating modules reduces the number of operations for a mechanism of comparable complexity. Economies of scale come into play and making many modules becomes feasible.

- *Redundancy:* Unit-modularity usually implies highly redundant systems since many modules are available due to the ease of manufacture.

- *Repairability:* If a module fails, it is easy to replace the module since there are many of the same kind. Reconfigurability adds the characteristic that the system can be self-repairing.

- *Robustness:* Redundancy and repairability combine to add to the robustness of the system. Redundancy alone does not necessarily increase robustness as adding redundant components adds more components that can fail. There are two properties which mitigate against this. First, modules can each be made very simple which usually results in a higher robustness per module. Second, typically each module in a system has a limited effect on the overall performance; thus the failure of one module is not catastrophic. The result is a gradation of failure instead of a catastrophic one in non-redundant systems.

- *Ease of design:* Modularity has always been useful as a way of breaking down complex systems into simpler modules to help in both design and analysis (which are tightly coupled).

Another property of highly redundant modular robots is that each module usually has a relatively limited workspace. This is often a result of the small size of each module relative to the overall robot. For revolute joints, the range of motion is typically on the order of ±20 degrees and usually less than ±45 degrees (which is the range of the Polypod segments). These robots rely on the serial chaining of modules to attain much larger workspaces.

1.5 Reconfigurability

One possible measure of versatility for reconfigurable robots is the number of morphologically different shapes that the robot can assume. Chen et al [Chen et al 1993] examined the problem of enumerating the isomorphic shapes for modular robotic systems that can be arranged in a tree-like structure. In graph theory, Harary poses the question as "How many animals?" where an animal is made up of regular polygonal shaped cells [Harary 1967]. If the cells are labelled, Klarner (1965) found a lower bound on the number of different shapes possible, a_n, for n square shaped cells.

$$a_n > \frac{3.6^n}{8} \tag{1.1}$$

For non-labelled cells, a_n would be smaller. In any case a_n is typically exponential in n. Modern algebra and the study of symmetry groups (such as the Polya-Burnside enumeration method) is one way to approach this problem.

For unit-modular reconfigurable robots, several factors contribute to this number. They are: the number of connection ports per module, the number of ways that two connection ports can be attached, symmetries in the connection port and symmetries in the module.

If there is only one connection port, then only two modules may be attached. Two connection ports means that modules may be attached in a serial chain, or a single closed loop. At least three connection ports are needed to attain tree-like and arbitrary graph structures.

The size of a_n can be increased further if there is more than one way that two connectors can mate. Connectors can be made hermaphroditic, that is they contain both genders so that any connector may mate with any other connector (instead of just male to female). Symmetries in the connector also allow multiple ways that two connectors can mate. For example, a common house outlet and two prong plugs have one rotational symmetry allowing the plug to mate two ways, by flipping the plug 180 degrees. Symmetries in the structure of modules on the other hand, reduce the number of kinematically distinct configurations.

Each additional manner that two connectors can mate forces a redundancy in the communications link. For example, if both male and female portions are included on a connection mechanism then circuitry for communications through both female and male portions must be included. This follows for the rotational symmetry as well.

The study of reconfigurability will not be discussed further in this thesis. Readers are referred to [Chen 1993][Harary 1967] and [Fukuda 1992].

1.6 Overview

Chapters 2, 4 and 5 present the design of Polypod and a simple control method which allows Polypod to implement many different modes of locomotion. Before the control and implementation of Polypod locomotion, however, we present a taxonomy of locomotion in Chapter 3. This taxonomy is entirely general and can be applied to all statically stable vehicles as well as Polypod and forms the structure for the presentation of Polypod locomotion in Chapter 5.

To evaluate and compare different classes of locomotion one must look at the environment for which the locomotion is intended. The main evaluation characteristic of an area of terrain is traversability – which areas of the terrain can the robot traverse over. We present a taxonomy of terrain features and corresponding vehicle parameters that can rate a vehicle's ability to deal with that feature in Chapter 6. Finally we evaluate the many different modes of Polypod and the classes of locomotion that they represent and give an example environment in which Polypod must reconfigure to different gaits in order to follow a given path.

Chapter 7 is the last chapter before the concluding chapter, and explores the prospects for scaling up the number of modules. We investigate the limitations on the number of modules that can be connected together in terms of structure, actuation, computation, and control.

Chapter 2

Polypod Design

2.1 Introduction

A dissertation on a unit-modular and reconfigurable robot only has value if such a robot can be built. Moreover, discussing theories and algorithms for locomotion is not so important if those theories and algorithms cannot be implemented on a real robot. This chapter presents the design and implementation of Polypod. The rest of this thesis studies locomotion and the capabilities of modular robots, with this design as a particular case.

Polypod is the name of the reconfigurable modular robot system designed and implemented by the author. It consists of two modules so, by definition, it is not exactly unit-modular, however virtually all the benefits are the same. Also, since most of the functionality of Polypod is in one of the two modules, we will treat Polypod as if it were unit-modular. Eleven modules have been built at the time of this writing.

2.2 Philosophy of Design

The goal of the Polypod project is to build a highly versatile robot that can accomplish a large range of tasks. The optimal (or good) design of a robot depends on the range of tasks it is to perform. Polypod is not an attempt at the end-all be-all solution to robotic design as we focus particularly on locomotion. Designing a robot for the task of locomotion differs from the typical manipulator design in the following ways:

- Low precision: In most types of terrain, the foot placement or the configuration of the robot does not need to be precise.
- Low stiffness: Low stiffness is allowed as a result of low precision.
- Low maximum velocity: Statically stable locomotion does not require high speed.
- On the other hand, the possibility of combining several kinematic chains, including closed loops is significant.

Unlike the modular reconfigurable manipulator systems previously described, a unit-modular robot must have all major components on one module. That is link structure, actuator, interconnection mechanism, computation and power must all be on every module. This is the first requirement of the design, full functionality. In the case of Polypod we separate out the power requirement. We have one module called a **segment** which contains all components except a power supply and a second module called a **node** which holds the power supply.

Any robot or electro-mechanical system can be broken down into three parts: the electrical hardware, the mechanical hardware, and the software. Software is easily changeable and great improvements can be achieved with better software; however, the best that can be achieved by software is always limited by the hardware. Thus, the first design philosophy of Polypod was to build hardware that was the least limiting in the types of tasks that it could achieve.

Two items that partially define the suitability of a robot to specific tasks are the degrees of freedom of the robot and the size of the robot. The size has two conflicting roles:

1. the robot must have a large enough workspace to reach all points necessary,
2. the robot must be small enough to be contained within the boundaries of the environment and easily avoid obstacles.

Since Polypod is modular, to achieve large workspaces we may add as many modules as needed (assuming we have a very many of them). However, to be useful in constrained environments, we need these modules to be as small as possible. Considerable effort was put into making the modules small while still cost effective. The result is modules that are less than 10 cubic inches (2.5in. on a side) that cost less than $100.00 each. For mass produced modules, the cost and possibly size could be reduced dramatically.

The design goals for the Polypod modules are summarized below:

1. Full functionality in one module.

2. Minimum size.

3. Manufacturability (to make many modules feasible and cost effective)

4. Minimum cost (limited available funds)

5. Although stiffness of the structure, and strength and speed of the actuator is not critical for statically stable locomotion, these were also desirable.

Is it possible to be too small? While a small size is one of the overriding goals, we are limited by the available tools and parts. For example, there exist very small motors and transmissions, on the sub-centimeter scale. However, their cost is prohibitive. Micro-structures and actuators (on the micrometer scale) using silicon wafer technologies are an interesting possibility for the future. But at present, integrating them into a fully functional module is not feasible.

The two types of modules of Polypod, segments and nodes, are shown in Figure 2.1. The next two sections describe the segment and the node, respectively. They are followed by sections on the interconnection system and manufacturability.

2.3 Segment

The segment is a two-DOF module with two motors actuating the two DOF, position and force sensors, a micro-computer, and two connection ports.

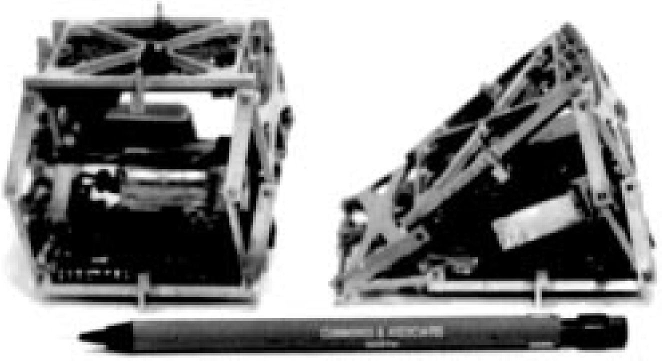


Figure 2.1: Photo of a segment and a node

2.3.1 Structure

The first question in designing the structure is how many DOF should be put into a module and what type (rotational or translational). A larger number of DOF per module allows more functionality with a given number of modules, however it complicates the design of each module. Since we assume that many modules are easily obtained, we instead choose to minimize the number of DOF per module. If more DOF are required, more modules will be used.

Prismatic joints in combination with revolute joints have been shown to have a large reachable workspace in a cluttered environment [Korolav 1990]. We therefore choose to have both a prismatic and a revolute joint in each module.

The prismatic DOF poses a problem since single-stage prismatic joints require a large amount of space violating our second design goal. In modules like the variable geometry trusses described in the previous chapter, the ratio of each prismatic joint between fully compressed and fully expanded is 1:2 at best, often closer to 2:3. Multi-stage prismatic joints like those used in the Odetics mobile robot "Robin", are typically expensive and complicated to build, violating our third and fourth design goals.

So a parallel structure is chosen, using four-bar linkages as prismatic actuators. In this way, the theoretical ratio is infinite. (For Polypod we achieve a 1:4 ratio for

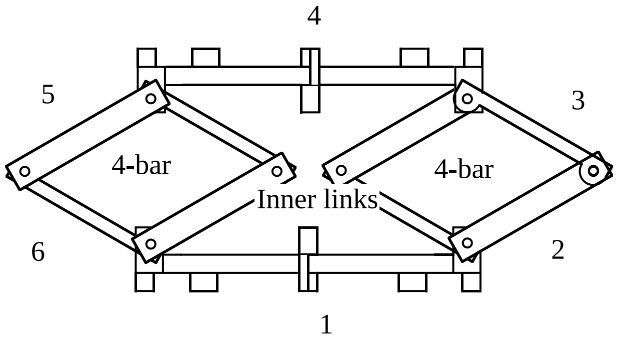


Figure 2.2: Segment is a 10-bar linkage

each joint.)

Finally, the resulting structure of the segment is a ten-bar linkage combining two four-bar linkages and two connection plates as shown in Figure 2.2. The two pairs of inner links of the two four-bar linkages are kinematically redundant leaving a six-bar linkage. Normally a six-bar linkage has three DOF, but we eliminate one DOF by the addition of a "sliding bar" as shown in Figure 2.3. This bar constrains two joints of each four-bar to remain co-linear. The bar is called sliding because the joints of the inner links must slide on the bar. The bar is composed of two telescoping tubes so that as the segment moves, the bar will not protrude outside the boundary of the ten-bar linkage. Thus all surfaces of the segment may be used as manipulation surfaces, as in the whole-arm manipulation concept of Salisbury [Salisbury 1988].

Why is this third DOF eliminated rather than actuated? Actuating it would greatly complicate the system since its addition disrupts the symmetry of the first two DOF. The penalty of adding a third actuator, both in terms of size and cost, outweighs the benefits in versatility especially since another module may be added

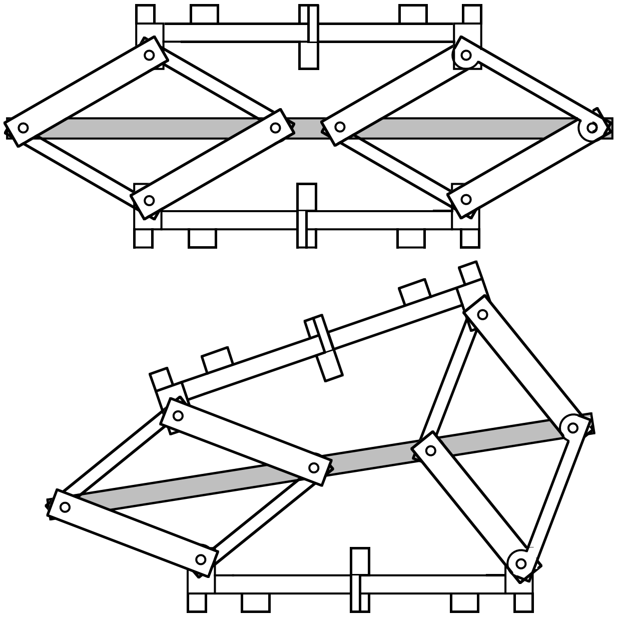


Figure 2.3: Sliding bar shown shaded

serially which can emulate the existence of the third actuator.

This parallel structure has the added benefit of being stiffer than a corresponding serial structure, and is rotationally symmetric about the center.

2.3.2 Kinematics

Kinematically, we may treat the ten-bar linkage with the sliding bar constraint as a serial configuration of two prismatic joints linked with a revolute joint with the prismatic joints constrained to have the same length, as in Figure 2.4. In the figure we attach a frame to the center of the base connection plate and call it $\{B\}$ (using Craig's [Craig 1986] notation) and a frame to the center of the end connection plate and call it $\{E\}$. We will refer to the two constrained prismatic joints together as the prismatic DOF whose generalized coordinate is specified by D, the sum of the lengths of both joints. The revolute joint is referred to as the revolute DOF with generalized coordinate θ, the angle between $\{B_x\}$ and $\{E_x\}$. These parameters relate $\{B\}$ to

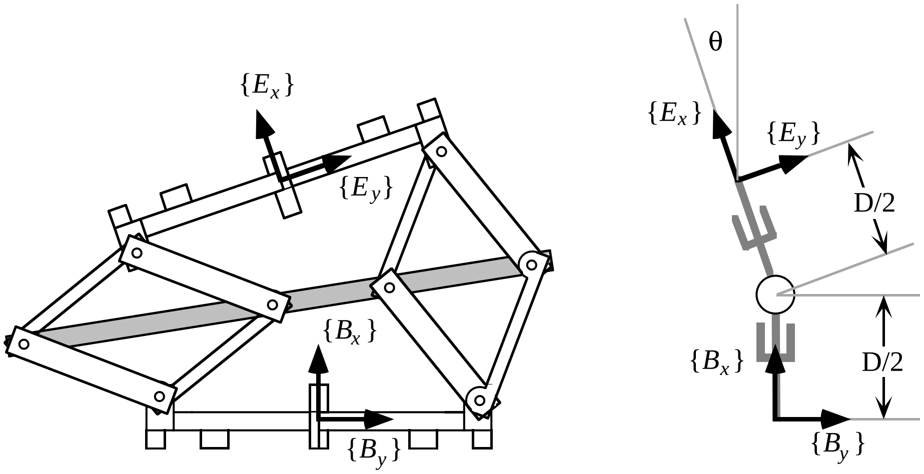


Figure 2.4: Segment kinematics

$\{E\}$ by the following equation:

$$E = {}^{E}_{B}TB, \tag{2.1}$$

where ${}^{E}_{B}T$ is the transformation matrix below:

$${}^{E}_{B}T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & \frac{D}{2} + \frac{D}{2}\cos(\theta) \\ \sin(\theta) & \cos(\theta) & 0 & \frac{D}{2}\sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.2}$$

Table 2.1 shows the Denavit-Hartenberg parameters as interpreted in [Craig 1986] for one segment and two segments attached perpendicularly. For each segment there are three links $\{i\}$ corresponding to the two prismatic and one revolute joints. Note that the two prismatic joints have the same variable D. We will refer to the space of all 2×1 vectors formed by $\{\theta, D\}^T$ as the **RPjoint space**.

One Segment

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	D	0
2	90	0	0	θ
3	90	0	D	0

Two Segments Mounted Perpendicularly

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	D_1	0
2	90	0	0	θ_1
3	90	0	D_1	0
4	0	0	D_2	90
5	90	0	0	θ_2
6	90	0	D_2	0

Table 2.1: Denavit-Hartenberg parameters for one and two segments.

2.3.3 Actuator

The type of actuator chosen to drive the four-bar linkage is decided by two factors, size and cost. Pneumatic and hydraulic actuators are out of the question if the robot is to be autonomous, since carrying around a large compressor would not be feasible. Furthermore, there is too much loss in valves and actuators to use pressurized cartridges. In addition, the connectors required for reconfigurability would be costly and large.

Shape memory alloy (SMA) actuators are an interesting new actuator. However, control and range of motion make them difficult to use. They also require precise temperature control which is difficult in uncontrolled environments. Piezo-electric motors are another promising new actuator technology, but cost and availability precluded its use.

Electrical motors are by far the most prevalent in the size range of interest. And of these, DC brush motors are the smallest and cheapest. So we choose these actuators.

The two actuators are mirror images of each other as shown in Figure 2.5. The small motors drive a tapped pulley on a lead screw via a toothed belt. The size and

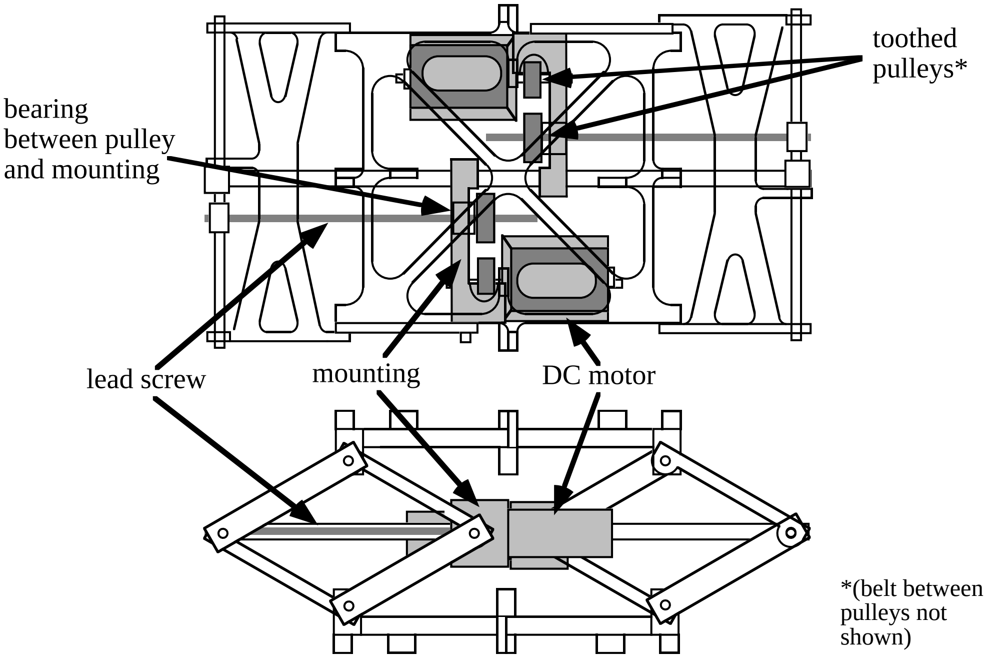


Figure 2.5: Motor and transmission schematic

construction of this motor and transmission were the optimum available for the space and cost allowed (the motors cost 60 cents each). This transmission is not back-driveable and has some backlash and non-linearity in the belt. This implies that this system would be very difficult to model for high-performance control. However, high-performance control is not a design goal of Polypod.

The forces from the linear motion are applied perpendicular to the nominal load forces as shown in Figure 2.6, so a singularity in the Jacobian matrix mapping the joint space to Cartesian space exists when the segment is fully extended.

At the singularity, mechanical advantage goes to infinity. The mechanical advantage for the segment as a function of actuator position is given in the following equation:

$$M_A = \tan(\arcsin(\frac{X}{L})) \tag{2.3}$$

where M_A is the multiplier of the actuator force for the force seen at the load, X and L are the measurements of the distance between load points and the length of the actuator links, respectively, as shown in Figure 2.6.

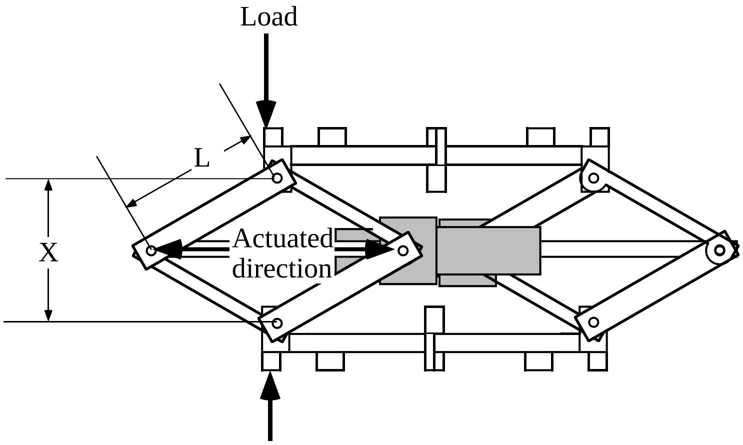


Figure 2.6: Mechanical advantage

Not being back-driveable is another advantage in disguise. This can also be interpreted as being self-locking. That is, the mechanism will not move unless it is commanded to move. We can take advantage of the singularity and the self-locking aspects of the actuation to move very heavy objects with two columns of segments. One column supports a heavy weight while self-locked, while the other set moves such that at least one segment is near the singularity when it starts to lift the object a small distance. Once a joint limit is reached, that column stops (and is self-locked) and the two columns switch roles. This procedure repeats until the object is moved to its goal position.

Hirose showed that self-locking mechanisms are also very important for efficiency in statically stable walking machines [Hirose 1984]. These machines use swinging motions, where some of the motion is used to support weight. Self-locking actuators that are gravitationally decoupled are more efficient than most back-driveable actuators, since they can lock during the weight support phases without expending energy.

The motors used in Polypod are cheap, square, open DC motors about 0.3 cubic in. There was the option of using high quality $60.00 gear motors that fit the form factor or using these $0.60 motors. The negative side of using the cheap motors is that they tend to be weaker, and they generate a large amount of EMI/RF interference which can reset the logic and wreak havoc with the sensors.

The low torque problem was solved by driving the motors at a higher voltage level than the manufacturer specified. A difficulty with overdriving the voltage is overheating. This would occur when the motors are at full duty cycle and stalled for a sustained period of time, which can be avoided by proper software control.

Overdriving the motors unfortunately exacerbates the noise problem as well. Solving the noise problem took a great amount of effort. However, proper shielding and power isolation solved it. The reader is referred to the text by Ott, [Ott 1988] as a great help in solving electrical noise problems.

2.3.4 Sensing

There are two types of sensors built into each segment, two position sensors and two three-function IR emitter-detector pairs. Each of these sensors is not particularly sensitive or high-performance, though on a robot with many segments, the sensing capabilities should increase with the increased numbers of sensors.

Position: There are two position sensors to sense the position of the two DOF of the segment. Two small surface-mount-device potentiometers measure the angles of each four-bar linkage. This is done by mounting the base of the potentiometer (pot) to one link and the wiper to the other link, with the rotational axis of the pot coincident with the hinge of the two links. The output of the pot is fed into an analog-to-digital converter built into an onboard single chip micro-computer. This configuration for position sensing is more susceptible to electrical noise (especially from noisy motors) than encoders, however the smallest encoder commercially available was over two orders of magnitude larger in volume and weight.

Three-Function IR: Infra-red emitter-detector pair diodes are mounted on the printed circuit board (PCB) mounted on each connection plate. These diodes provide three functions: proximity sensor, force sensor, and local communication medium.

When the connecting plate is not connected to another plate (on the distal end of an arm for example), the emitter-detector pair faces outward and can be used as a reflective proximity sensor. Since there are two pairs of such sensors, two range values can be read to get a sense of orientation as well as distance. A typical situation would be one where the connecting plate on the bottom of a segment is acting as a foot. The proximity sensor would then give a rough estimate of the distance to the ground, as well as the angle of the foot with respect to the ground. Since the range measurements are based on the intensity of the reflection of the sensed object, the accuracy is susceptible to the color and specularity of the object as well as ambient light.

The two connecting plates connecting two segments have the emitters on one plate facing the detectors of the other plate and vice versa. Since this system is enclosed, the range as a function of intensity is more reliable and so can be amplified a great deal. Resolutions of 0.00001 inches were obtained. This distance is used to calculate the force when given the elasticity of the latching mechanism holding the two plates together.

This force sensor will have hysteresis in the output since it is made up of two moving parts with friction between them. This degrades the accuracy of the force sensor however, in most cases, the sensor is used to sense only the general direction and rough size of forces. Precise force measurements are not needed.

The third function of the IR sensors is as a local communications medium between two adjacent segments. Since emitters on one side are facing detectors on another, a natural communications medium is established.

2.3.5 Computer and Electronics

There are four functions that the electronics must provide for each segment. Any unit-modular system which shares communications and power needs the following four subsystems:

1. computer and communications drivers

2. connection system (communications and power bus)

3. actuator electronics (motor drivers etc.)

4. sensor electronics (PCB mounted sensors, analog signal conditioning, etc.)

The electronics are implemented on two PCB's that are mounted into the connection plates. One board is mainly for the computer and communications drivers, the other board is mainly for driving the actuators. Both boards have identical connection systems and sensing.

Computer and Communications: Each segment has a Motorola MC68HC11E2. This 68HC11 series micro-computer is a popular 8-bit processor with on board memory (512 bytes RAM), non-volatile memory (2 kilobytes EEPROM), A/D (8 channel 8-bit), and digital I/O all on one chip running at 2MHz. This micro-computer was chosen since it contains all the necessary functions in a small 44 pin PLCC package (a footprint of 1 in^2). Cost was also an issue in using this particular CPU as Motorola generously donated over 100 compatible devices to this project.

The 68HC11's have a built-in high speed (1 megabit/sec) serial peripheral interface (SPI) which provides a simple high-level protocol for multiprocessor communication. To help increase the noise immunity, RS485 compatible drivers (a high-speed multi-drop protocol) were added to interface between processors.

The network is in a single master architecture. Typically a computer on a node module serves as the master computer, although this does not have to be the case. Any computer on any module may be the master. This is done by setting appropriate jumpers on the PCB's. The computer on the node is presently a 68HC11 as on the segments, however upgrading this computer to a MC68F333 a 32bit single chip micro-computer is possible as there is more physical space available inside the node than in the segment.

2.4 Node

Nodes are rigid cube-shaped modules made up of six connection ports, one on each face of the cube. One reason for choosing the shape of a cube is that a cube is the only known regular solid that can be close packed in three-dimensional space.

The node is pictured in Figure 2.1. Nodes hold two functions. One is to contain the power source for the segments. The other is to allow non-serial configurations. Virtually any shape can be approximated with enough segments and nodes.

The power source chosen was gel-cell batteries for the amount of peak output (Amps) and the rechargeable nature of the batteries. It turned out that poor efficiency of the motors and transmission system required external batteries as well as those supplied in the nodes for many of the initial demonstrations.

2.5 Interconnect System

The connection plates as shown in Figure 2.7 have two functions: to attach one module connected to its neighbor, and to electrically connect a power and communication bus. These two functions are needed on any modular reconfigurable system that is autonomous and has a separate power supply module.

Two plates connect by having the positive X-axes (labelled $\{E_x\}$ in Figure 2.7) of each plate point toward each other. As they approach, chamfers on the plate guide the plates together, plastic chamfers on the electrical connectors guide mating connectors together, and finally a spring loaded latch mechanism latches the two together.

The connection plates are symmetric four times about the center along the X-axes and are hermaphroditic, so there is no need for a male form and a female form. The plate symmetry allows four ways that two plates can be connected. However, since the segments are also symmetric twice about the X-axis, there are only two distinct ways that two segments may be attached to each other: one where the segments move in the same plane as in Figure 2.8, and one where they move in perpendicular planes as in Figure 2.9. According to the frames attached to the links in Figure 2.4, Figure 2.8 has all Z-axes parallel, and Figure 2.9 has the Z-axes of adjacent connecting plates

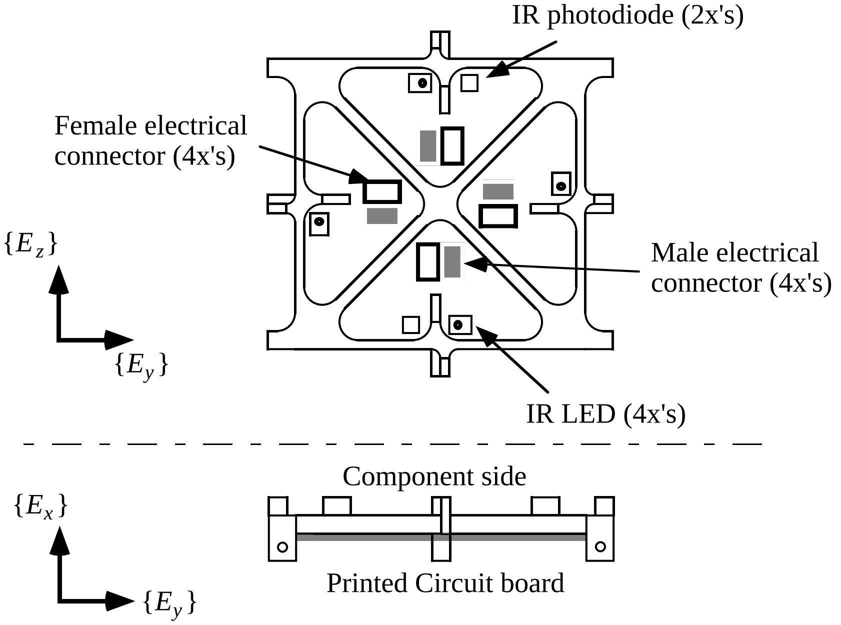


Figure 2.7: Schematic of connection plate

perpendicular to each other.

Nodes are symmetric four times about the X-axis of any connector, so there is only one morphologically distinct way that a node may be connected to another plate.

For dynamic reconfiguring, care must be taken with the power bus. If two robots with two separate power supplies join while both are in different power states, noise spikes may result, or if the power supplies are both regulated, the power regulation can go unstable. For Polypod, unregulated batteries and power noise shielding is used.

Electrical connectors are often the failing point of electro-mechanical systems, especially in those where disconnecting and reconnecting are automatic. The power and communications bus are nominally eight times redundant, both in view of this potential problem and to allow the four-way symmetry that exists in the connection plate.

The electrical bus consists of ten lines, four power lines (one pair for motor power, one pair for logic and sensing power), and six lines for a high-speed synchronous bus

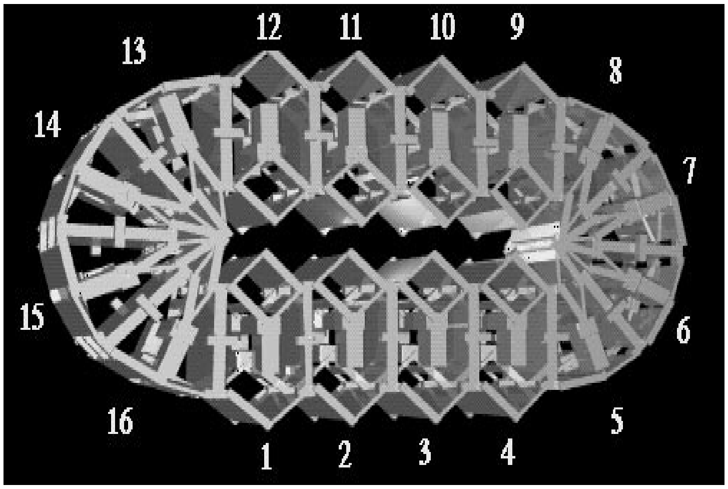


Figure 2.8: Rolling-Track locomotion, planar motion

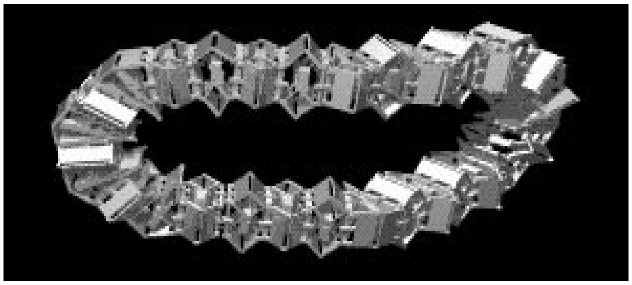


Figure 2.9: Turning Rolling-Track locomotion, out of plane motion

(one pair for a clock, one pair for master in, one pair for master out).

The latching mechanism holds two segments together. It is a spring-loaded hook design similar to Fukuda's CEBOT latch mechanism. When two segments are brought together, the mechanism latches them. The unlatching actuator is a shape memory alloy (SMA) wire actuator which simply pulls the hooks back. SMA actuators are notoriously difficult to control in an analog fashion, however in a simple on/off design as here, SMA actuators work well. The latching mechanism has been designed, but has not yet been fully integrated into Polypod.

2.6 Design For Manufacturability

One advantage of unit-modular systems over standard modular systems is the ease of manufacturing due to repeated parts. The design of all the structures within each module as well, was done with this in mind.

The segment is made up of sixteen machined parts, the node is made of eighteen. Of these thirty-four machined parts only five are unique. To achieve this small number, symmetries in the structure were exploited. For example, the node is a cube with six identical faces, one part can be repeated six times. Each piece was designed to be machined on a three-axis CNC machine. Multiple copies of each piece could be made with one or two simple fixturings.

As an evaluation of the manufacturability, note that eight segments and three nodes were designed, machined, assembled and debugged by one full-time student (with occasional help from independent study students) in the course of his PhD work on a minimal budget. In the final form, one segment or node could be built per man-day.

The design of the segments as a ten-bar linkage is also easily modified to be made out of sheet metal or by an injection molding process to make very low cost large scale manufacturing feasible.

Chapter 3

A Taxonomy of Locomotion

The exciting thing about a modular robot like Polypod is that it does not describe one robot, but presents the building blocks from which many different types of robots can be formed. Furthermore, dynamic reconfigurability adds another dimension to the capabilities of the robot. The result is that it becomes difficult to find the limits of what it can do. Consequently, we cannot explore statically stable locomotion of modular reconfigurable robots per se, but must first study statically stable locomotion in general.

3.1 Introduction

This chapter presents a taxonomy of statically stable locomotion. After presenting the taxonomy, an analysis of the different classes of locomotion is presented giving insight into the relative advantages of one class over another.

The basis for classification in a taxonomy depends on the intended use. A taxonomy is most useful in terms of the insights and generalities it gives by the specific classification. For our purposes, it should help in comparing locomotion gaits, as well as facilitate the development of new gaits. In this sense a functional taxonomy is most practical as we are not classifying vehicles, but the methods of locomotion.

The most typical classification of land locomotion divides locomotion into four areas: wheeled, tracked, legged, and all others. For our purposes, this is unsatisfactory for several reasons. First, the last area is a catchall, and would include such dissimilar means of locomotion as snake-like sidewinding, concertina, screw locomotion, etc. Second, there are too many instances of ambiguity. For example, a child cartwheeling may be considered legged locomotion since the child has legs. Would a spoked wheel with no rim or partial rims also be considered legged locomotion? Tracked locomotion is defined as traveling on endless belts. Is a belt around a tire then tracked locomotion? What about a slightly flat tire?

If we want to examine new and novel ways of locomotion and look at locomotion in general, the above method is not adequate. Clearly the above method could be applied to vehicles or mobile robots simply by asking does it have wheels, legs, tracks, but it does not necessarily tell us anything about how the vehicle locomotes.

3.1.1 Definitions

The definition of locomotion is the act or power of moving from place to place. Our interest lies in sustainable locomotion, not single events such as falling off a table. We can divide sustainable locomotion into three components:

1. Moving some incremental straight-line distance

2. Turning and/or translating in multiple directions

3. Path planning and navigation.

In this thesis we will concentrate only on the first two items.

Gait Locomotion: For the first two components of locomotion listed above, a pattern of motion is usually repeated for each increment.

Definition 3.1 *A gait is defined as one cycle of a pattern of motion that is used to achieve locomotion.*

One characteristic of a gait in a homogeneous environment is that at the beginning and end of the gait the vehicle is in identical configurations with some net rotation and/or net translation. It is certainly possible that a sustainable motion may not be cyclical in nature, for example, a craft that uses rockets for levitation and propulsion or a legged animal that moves its legs in random motions. As in these examples it seems that either an effort must be made to avoid cyclical behavior or the locomotion is not statically stable.

Simple and Compound Gaits: Before presenting the taxonomy of locomotion, there is a need to understand the concept of simple and compound locomotion gaits. Classifying locomotion gaits is different from many subjects in systematics in that locomotion gaits can be combined with other gaits in specific fashions. For example as wheeled locomotion is one type of locomotion and bipedal walking clearly is another, the two can be combined as with a person wearing roller skates. And further, if one can imagine gluing the backs of ants to the wheels of the skates (super strong ants that would not be crushed and could support the weight of a person) the ants could walk - as the wheels roll - as the person walks, and we would have yet another completely different type of locomotion. And if the ants had roller skates... this process could go on ad infinitum. This may not be realistic, however it illustrates one aspect of combined gaits.

Definition 3.2 *Simple gaits are those which cannot be broken down into separate gaits.*

Definition 3.3 *Compound gaits are gaits which are combinations of simple gaits.*

3.1.2 Turning Gaits

In most applications the robot will be expected to be able to turn. The definition of a turn for existing vehicles such as cars or legged animals is clear as there is a rigid body on which the wheels or legs attach; turning is measured by the rotation of this rigid body. In general however, a central rigid body does not necessarily exist, as in a snake for example. We define a turning gait to be one pattern of motion that starts and ends with the robot in the same global shape, though rotated.

There are two significant forms of rotating, differential translation and sequential rotation. Differential translating occurs when two or more generalized feet translate at different velocities along lines that are not co-linear.

Sequential rotation turning applies to articulated vehicles like a snake or a train. It is characterized by portions of the vehicle rotating in sequence as the vehicle moves forward such that each portion rotates over the same point on the terrain. For example, as each car in train moves over a curved section track, the car rotates. This type of locomotion is particularly useful for moving a large robot through a cluttered environment. In addition, each portion of rotation may use differential translation.

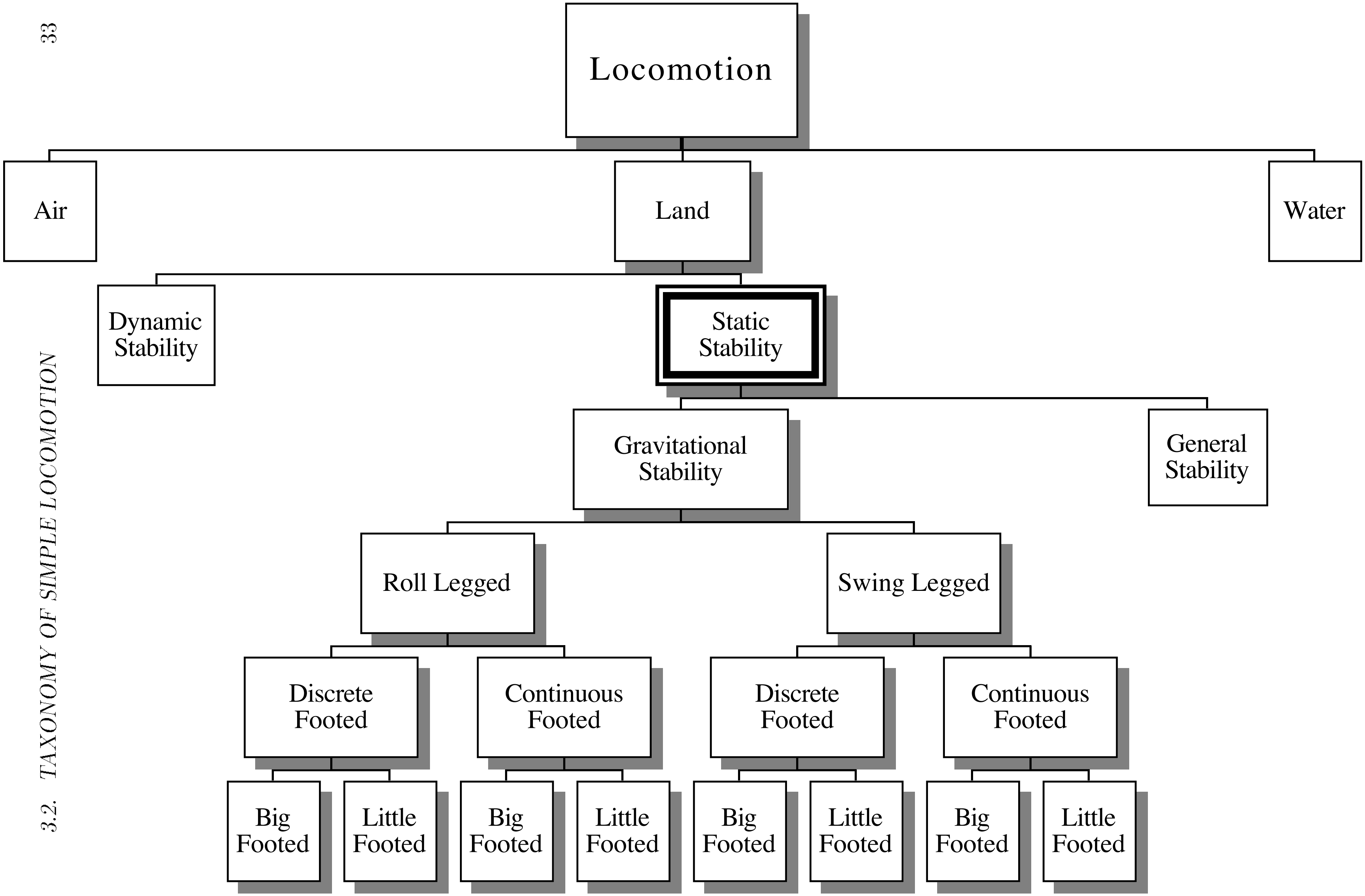
Since straight line gaits may be used to implement turning (as in differential translation), this taxonomy of locomotion will concentrate straight line locomotion only.

3.2 Taxonomy of Simple Locomotion

3.2.1 First Level: Air, Water or Land

The chart in Figure 3.1 shows the full classification scheme for simple locomotion. We are interested in the middle of the chart, statically stable locomotion. It is classified as a land type of locomotion. We will touch lightly on the other types of locomotion just to define statically stable locomotion within the global picture.

The first level of locomotion is divided into air, water and land locomotion. Or perhaps a better nomenclature is gas, liquid, and solid. This level of locomotion is defined by the type of propulsion that is used. Gas or air locomotion pushes gases to achieve locomotion, liquid locomotion pushes liquids and solid locomotion pushes solids. For example a propeller boat is half in air, half in water, yet it pushes water to get propulsion and to steer and so is classified as liquid or water locomotion. On the other hand a sail-boat which relies on air for propulsion would be air locomotion. Another example is a legged underwater robot, it travels underwater, yet pushes the ground for propulsion so is classified as solid or land locomotion. The distinction between air and water is made here more for historical reasons rather than theoretical


Locomotion
Air
Land
Water
Dynamic Stability
Static Stability
Gravitational Stability
General Stability
Roll Legged
Swing Legged
Discrete Footed
Continuous Footed
Discrete Footed
Continuous Footed
Big Footed
Little Footed
Big Footed
Little Footed
Big Footed
Little Footed
Big Footed
Little Footed

as they are both essentially fluids, (one compressible and one incompressible).

When we say land or ground or terrain, it is implied that the ground or land is a solid terrain feature that the land locomotor uses for propulsion or support, whether it is flat ground, the floor of a building or branches in a tree.

3.2.2 Second Level: Land Locomotion

A requirement for being sustainable is that the locomotion must be stable. There are two classes of stability with regard to land locomotion, dynamic stability and static stability.

Often in the literature, statically stable locomotion also implies quasi-static motion. We make a distinction here between statically stable locomotion and quasi-static motion. Quasi-static motion is defined to be motions that are slow enough that inertial, Coriolis and centrifugal effects are negligible. We will define statically stable locomotion as the following:

Definition 3.4 *Statically stable locomotion is any locomotion that is statically stable at any instant, and that does not depend on inertial, Coriolis or centrifugal forces to maintain motion.*

So it is possible to have locomotion that is statically stable yet not quasi-static. An example would be a passenger car moving on a road. Inertial forces are significant, however if those forces disappeared, the car could still move from place to place. The only constraint is that the dynamic effects must not disrupt the stability of locomotion. One characteristic of static stability is that the locomoting body may have zero velocity and remain stable. An analysis of the stability of locomotion is presented in the locomotion evaluation chapter.

Dynamically stable locomotion includes all other forms of sustainable land locomotion.

Gravitational Stability

The most common form of land locomotion is that of travelling on a flat surface in the presence of gravity. A less typical form of land locomotion is seen when climbing a pole, or inside a chimney by pushing on the walls. In these latter cases, horizontal forces are used to generated vertical frictional forces for stability.

The key feature distinguishing these two forms of locomotion is that one form relies solely on the vertical force components of surfaces for support and stability, and the other makes use of horizontal forces. We call the first form, **gravitational static stability** and the second, **general static stability**.

Gravitational static stability exists when the vertical projection of the center of gravity (CG) lies within the convex hull of all ground contact points. For the remainder of this thesis we will explore only gravitational static stability.

3.2.3 Third Level: Taxonomy of Statically Stable Locomotion

Requirements For Statically Stable Locomotion: To achieve statically stable locomotion in general, one has to repeatedly do four things in any order:

1. remove ground contact points from the rear of the robot,
2. place ground contact points in front of the robot,
3. shift weight forward,
4. maintain static equilibrium throughout all motions.

A statically stable gait defines a cyclical pattern that achieves these steps.

One thing in common with all statically stable locomotion (and dynamically stable land locomotion) is that the vehicle must interact with the ground. These interactions are through the use of *generalized feet.*

Definition 3.5 *A* generalized foot or G-foot (G-feet plural) *is defined to be one contiguous set of points of a locomoting body that comes into contact with the ground.*

Note that this definition does not require the points to be in contact with the ground at the same time. So, the outer surface of a wheel would be *one* G-foot as the surface is contiguous and all points touch the ground at some point in time.

There is another form of foot that should be mentioned at this point, that of *time contact feet.*

Definition 3.6 *A* time contact foot or T-foot (T-feet plural) *is defined to be one contiguous set of points of a locomoting body that would come into contact with flat ground at the same time.*

A T-footprint *is the region of contact that would be made by the T-foot.*

Note that in this case a wheel is made up of an infinite number of T-feet. A T-foot is always a subset of the points in a G-foot. The body of a vehicle is defined as all portions of the vehicle that are not G-feet.

The next three levels of classification are not necessarily hierarchical. That is, the three levels can be considered orthogonal to each other and not a subclass of one another. The levels are roll or swing-legged, big or little-footed, and discrete or continuous-footed. We will first define each level and then look at examples in each of the eight possible combinations. As in our definition of a gait (Definition 3.1), we will assume ideal conditions

(R)oll or (S)wing-Legged [R—S]: Just below the gravitationally statically stable class in Figure 3.1 there are two fundamentally different modes of locomotion, rolling and swinging (moving back and forth). This classification is based on the cyclical nature of a gait as defined in Definition 3.1.

Definition 3.7 *Any gait that results in a G-foot going through a net rotation about any axis at the end of the gait is classified as a* roll-legged *gait. All other gaits are classified as* swing-legged.

The term swinging is used since a non-rotational gait must have some form of swinging motion.

(D)iscrete or (C)ontinuous-Footed [D—C]: The next level for each family concerns the continuity of ground contact points.

Definition 3.8 *Gaits which have G-feet which never break contact with the ground is classified as* continuous-footed. *All other gaits are classified as* discrete-footed.

If a vehicle uses a statically stable gait which has *one* G-foot, it must be continuous since that one G-foot must remain in contact with the ground to be statically stable. All simple continuous gaits have one G-foot.

(B)ig or (L)ittle-Footed [B—L]: The last level deals with the relation between G-feet and stability.

Definition 3.9 Little-footed *gaits is defined as those gaits for which one ground contact point per G-foot is a sufficient condition to maintain stability. If two or more ground contact points of a single G-foot are required for static stability, the gait is classified as* big-footed.

Big-footed gaits can have as few as one foot on the ground and still maintain static stability. Little-footed gaits must have at least three G-feet. Big and little footed is somewhat a misnomer since it is possible to have an arbitrarily small footprint that is still big-footed. For example, a sphere is statically stable and therefore big-footed since it has only one G-foot, but its footprint is not much more than a point.

3.2.4 Examples

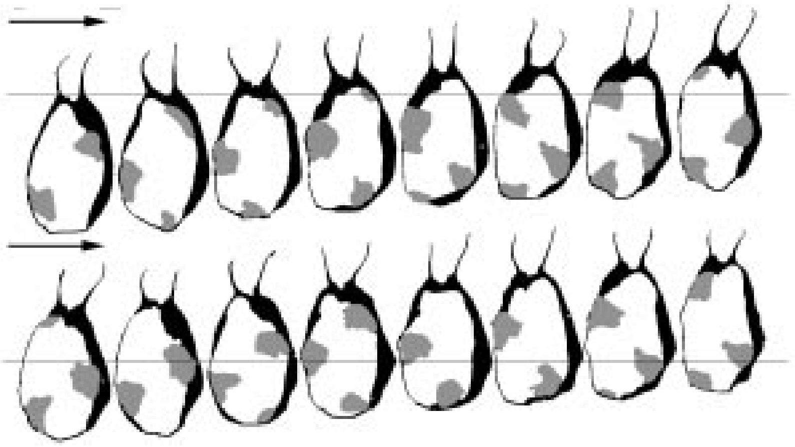
A simple statically stable gait is classified by three letters, **R** or **S**, **D** or **C**, and **B** or **L** for roll- or swing-legged, discrete- or continuous-footed, and big- or little-footed respectively. Table 3.1 is a listing of many animal gaits and mechanical gaits sorted according to their category.

Swing-Discrete-Little: In the animal world this is the most prevalent form of statically stable land locomotion. **SDL** gaits have a minimum of four G-feet, because of the need to have at least three ground contact points surrounding the center of gravity and one foot free to allow locomotion. In nature it is common to see six or more legs as on insects.

Example Statically Stable Vehicle/Gaits

Simple Gaits			
	Gait Class	Figure	Reference
4 Wheel Passenger Car	**RCL**		
Tricycle	**RCL**		
Screw Locomotion	**RCL**		Bekkar69
Treaded Tank	**RCB**		
Polypod Rolling-Track	**RCB**	5.1	Yim93
CMU Ambler	**RDL**	3.3	Simmons91
Polypod Cartwheel	**RDB**		Yim93
Polypod Slinky	**RDB**	5.2	Yim93
Sidewinding Snake	**SCB**		Burdick93
Loping Snail	**SCB**	3.2	Gray68
Earthworm	**SCB**		Gray68
Polypod Earthworm	**SCB**	5.7	
Panto-graph-legged	**SDL**		Shigley63
Dante,Dante II	**SDL**		
OSU ASV	**SDL**		
Odex I, Robin	**SDL**		Byrd90
Attila,Ghengis	**SDL**		Maes90
cockroach	**SDL**		Gray68
Stalking Cat	**SDL**		
Polypod caterpillar	**SDB**	5.8	Yim94
Human on balance beam	**SDB**		

Table 3.1: Examples of some simple locomotion classifications



based on [Gray 1968]

Figure 3.2: Loping Snail locomotion

A typical example of an **SDL** gait is a cockroach walking. At relatively fast speeds it uses the front limb and hind limb on one side of the body with the middle limb on the opposite side to form a triangle of support. The other three legs form another triangle that is placed after the first triangle has propelled the body forward. Alternating triangles for equal periods of time create a simple gait that many insects use [Gray 1968]. At slower speeds it may move one leg at a time.

In robotics, the simple cockroach alternating triangle is used quite often although it is usually called a tripod gait [Waldron 1990]. The majority of multi-legged robot "walkers" use an **SDL** gait. The most notable exception is the CMU Ambler which uses a **RDL** gait.

Swing-Discrete-Big: An example of an **SDB** gait can be seen when a human attempts to walk slowly on a balance beam. This gait is seen more often in robot toys that try to mimic human walking than in nature. Statically stable gaits are easier to synthesize and big feet tend to help stabilize dynamic walking gaits, so many biped robots are close to using **SDB** motions.

Swing-Continuous-Big: The **SCB** gait is best represented by the loping snail. The view of the loping snail's footprint as it travels is shown in Figure 3.2. There is one continuous foot that changes its shape to locomote. The dark patches in the figure are areas which do not touch the ground. These dark areas move forward in the direction of motion at twice the speed that the snail moves. Earthworms are another example, however they are actually built up of segments. In this sense the earthworm could be considered **SDB**, but it is classified as **SCB** as most of the properties of continuous gaits apply to the earthworm.

Swing-Continuous gaits are rare in mechanisms as usually they require near infinite DOF. The Polypod earthworm locomotion is one example of **SCB** as shown in Chapter 5. Its classification as continuous is valid if one considers Polypod earthworm locomotion through very soft soil. The entire lower surface of Polypod would be composed of one G-foot.

Swing-Continuous-Little: **SCL** gaits are rare in both nature and mechanisms as the swing-continuous gaits in nature tend to be big-footed. An imaginary example of an **SCL** gait would be a modified side-winding snake gait. Normally, a side-winding snake touches the ground in several long tracts in such a way that it is classified as big-footed. If it only touched the ground at the beginning and end of each tract (lifting its body in the middle), the gait would be **SCL**.

Roll-Continuous-Little: The archetypical **RCL** gait is represented by a vehicle with three or more wheels. It has continuous, point contact with the ground, and rolls as it translates. Unlike other little-footed gaits, **RCL** gaits only need three G-feet to maintain static stability as a continuous G-foot can translate without losing contact with the ground.

In the animal world, rolling gaits are not common. Although certain desert spiders have been known to curl up into a ball and roll down hills, this locomotion is not controlled. Very recently Full has presented an analysis of stomatopod locomotion [Full 1993]. This is a small crustacean which lives near the shore of oceans and is occasionally thrown onto the sand. It has legs which are too short to walk, so it propels itself by somersaulting, curling its body and rolling on the ground.

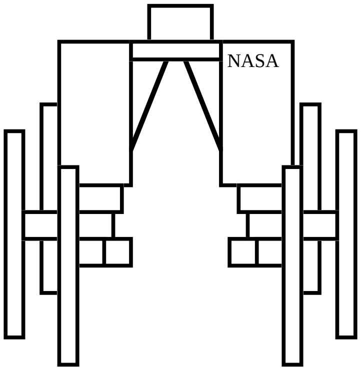


Figure 3.3: CMU Ambler

Roll-Continuous-Big: Most vehicles with tracks use an **RCB** gait such as a tank or bulldozer. A half-track truck has two tracks in the rear and two wheels in the front. If the center of gravity lies between the two tracks, then it uses big-footed locomotion. If the center of gravity lies ahead of the two tracks, then the tracks are not big-footed and the vehicle uses **RCL** locomotion.

Roll-Discrete-Little: **RDL** locomotion is best represented by a wagon with rimless wagon-wheels. Here the wheels rotate to achieve locomotion, however they contact the ground in a small set of discrete contacts. At least three rimless wheels are needed although each wheel is made up of many G-feet (the end of each spoke is one G-foot).

The CMU Ambler is an interesting case and is shown in Figure 3.3. While at first appearance this robot seems to be an **SDL** robot with six legs, it uses an unique circulating gait. The trailing legs recover through the middle of the body to become leading legs. They then rotate outside the body while in contact with the ground. Thus they rotate about a vertical axis. This rotation places the robot in the rolling category while the foot contacts place it in the discrete.

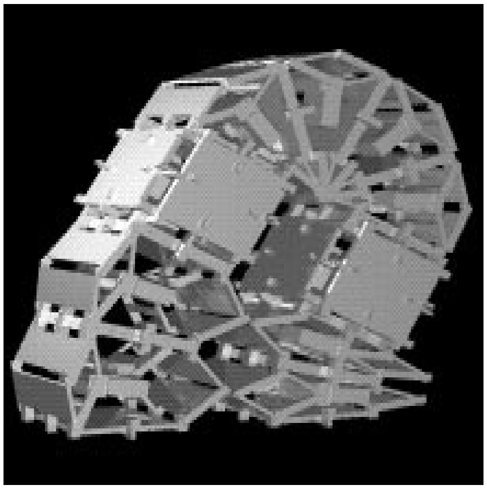


Figure 3.4: Slinky locomotion step 1

Roll-Discrete-Big: The Polypod Slinky locomotion is an **RDB** gait that is named after the "Slinky" toy. The toy is made up of a long coil of wire that traverses down stairs end over end. Both use the same form of locomotion composed of two steps, placing a generalized foot in front then shifting a majority of weight onto that foot. These two steps are shown in Figures 3.4 and 3.5. Since the body continually rotates +180 degrees with each step, the locomotion is classified as rolling. The center of gravity is kept within one footprint, until it switches to the next G-foot, so it is classified as discrete big-footed.

3.3 Taxonomy of Combined Locomotion

There is an infinite number of possible combinations of locomotion gaits as illustrated in the ant example in Section 3.1.1. In this section we will present a framework for classifying compound gaits.

There are three fundamental ways that a simple gait may be combined with another gait: Articulation, Hierarchical, and Morphological. These three types are illustrated by a horse and buggy, a roller-skater, and Rolling-Slinky Polypod gait (Section 5.3.3) respectively. Each of these examples combine a discrete and a continuous footed gait resulting in completely different morphologies.

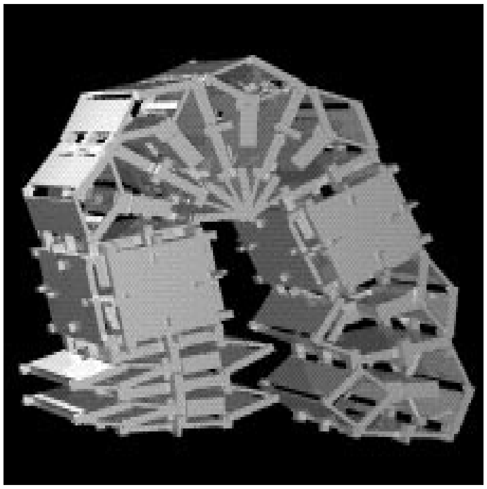


Figure 3.5: Slinky locomotion step 2

Note that in each case the simple gaits are still recognizable and in some sense separate. If this were not the case, the resulting locomotion gait would not be a combination gait but another simple gait by Definition 3.2. In the case of the horse and buggy, the two gaits are used on two physically separate bodies that are linked together. With the roller-skater, the two gaits are spectrally separate. Specifically, the frequency of the wheel gait is much higher than the frequency of the legged gait. In the Rolling-Slinky the two gaits are dimensionally separate. The rolling-track locomotes along one dimension, the Slinky along another.

3.3.1 Articulation

Definition 3.10 *Combination by articulation is defined to be the uniting by means of a joint, bodies which could separately execute a gait.*

When used in relation to vehicles, articulation usually implies a chain of bodies, like a train. With respect to mobile robots, articulated implies actuated joints linking bodies in a chain. Robots of this type have gained popularity partially because of the work of Hirose [Hirose 1985b][Hirose 1990].

For combined gaits, articulation implies adding components to a vehicle that may or may not use the same form of locomotion. In addition, the articulation need not

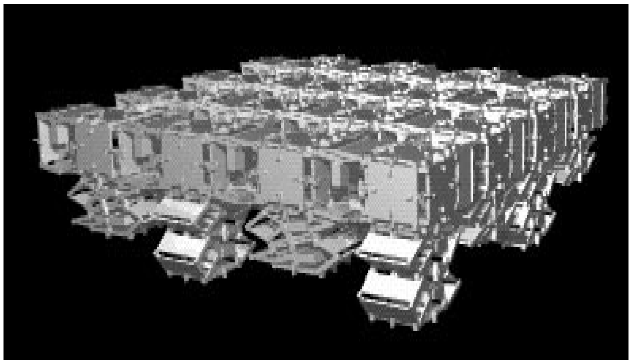


Figure 3.6: Example of non-serial articulation

be in a chain, but can be a two-dimensional array as in Figure 3.6.

One characteristic of this form of combination is that it lends itself easily to modularity. For example. an arbitrary number of cars may be added on to a train, and various numbers of horses may be added to a horse drawn carriage. Each horse or car could be considered a module.

Another characteristic is that *any* two gaits may be combined in this fashion. The proof can be seen by observing that any two vehicles can be linked together by a non-constraining 6 DOF link without effecting locomotion.

There is some overlap between big- vs little-footed gaits and articulated combinations. Indeed, any big-footed gait can be transformed into a corresponding little-footed gait by connecting three big-footed vehicles of the same type together such that the CG of the combined vehicle lies inside the triangle formed by the three.

The type of articulation serves as a further classification. Active or passive joints serve as one level of classification. The type of joint, e.g. ball joint, prismatic joint, etc., serves as the next.

3.3.2 Hierarchical Combination

Definition 3.11 *Hierarchical Combination is defined as combining two gaits, one called the superior and one the inferior. The inferior gait is added to the superior by modifying the superior's G-feet such that they no longer make contact with the ground and are supported by the inferior gait.*

The ant roller-skating example in Section 3.1.1 is an example of four successive hierarchical combinations. In practice it is rare to see two levels much less four levels.

Typically the inferior gait is implemented on a smaller scale than the superior. The inferior gaits also typically run at higher frequencies. Hierarchical combinations like articulated combinations can be applied to any two gaits given the appropriate mechanical structures.

A further classification of the hierarchical combination could be done by using all possible combinations of simple locomotion modes. For the statically stable case this would consist of 64 categories. Referring again to Section 3.1.1 the first two levels would result in an SDB-RCL classification. The gaits are ordered left to right by height in the hierarchy. The final result would be SDB-RCL-SDL-RCL

3.3.3 Morphological Combination

Definition 3.12 *Morphological combination is defined as the merging of gaits such that each gait can act in a different direction. Each gait can still make contact with the ground when locomoting (i.e. is not hierarchical) and cannot be separated by removing only a joint (i.e. is not articulated).*

The simplest example of this category is that of a sphere. A sphere can achieve the same rolling and twisting motions that a wheel makes as well as rolling instantaneously in any direction. Therefore it can be considered as the morphological combination of an infinite number of wheels.

Unlike the previous two categories, any two gaits cannot necessarily be combined in this fashion.

As in hierarchical combinations, morphological combinations can be further classified by the types of simple gaits that are being combined. In the statically stable case the sphere would be RCB-RCB-∞. The ordering of gaits is unimportant.

3.3.4 Compound Examples

Table 3.2 shows several examples of compound gaits.

Example Statically Stable Vehicle/Gaits

Compound Gaits				
	Gait Components	Compound Class	Figure	Reference
Truck & trailer	**RCL-RCL**	articulated		
Elastic Frame SLRV	**RCL-RCL**	articulated		Bekkar69
Polypod earthworm	**RCB-RCB...**	articulated		Yim93
Robby, Go-For	**RCL-RCL**	articulated		Wilcox92
RAMI	**RCB-RCB**	articulated		Iagolnitzer92
ACM III, IV, KRI	**RCL-RCL...**	articulated		Hirose90
Airoll	**RCL-RCL**	hierarchical	C.1	Bekkar69
Lockheed-Forsythe	**RCB-RCL**	hierarchical	C.2	Bekkar69
Moonwalk	**RCB-SDL**	hierarchical	5.12	Yim94
Rolling-While-Carrying	**RCB-SDL**	hierarchical	5.12	Yim94
Roller Skating	**SDB-RCL**	hierarchical		
Wheels with legs	**RCL-SDL**	hierarchical		Ohmichi83
Tracked legs	**SDL-RCB**	hierarchical		Kholer77
Tracked legs	**SDL-RCB**	hierarchical		Maeda85
Rolling Sphere	**RCB-RCB...**	morphological		
Rolling Slinky	**RCB-RDB...**	morphological		

Table 3.2: Examples of some compound locomotion classifications

3.4 Analysis

When deciding which gait would be appropriate for a given situation, it would be useful to know the characteristics of each type of classification. This is especially important to a reconfigurable robot that can choose which gait and configuration to use dynamically. A general analysis of each classification is done below. A more indepth analysis is done for Polypod in Chapter 6.

3.4.1 Simple Gaits

Roll vs. Swing-Legged: All rolling gaits have a rotational inertia that swinging ones do not. This can add stability to gaits which are not quasi-static. McGeer presents an analysis equating a swinging gait with a rolling gait in [McGeer 1990]. However, he lumps the mass of the wheel to a point thereby eliminating the rotational inertia.

The rolling action in most rolling discrete cases is used in the recovery portion of a footstep. In most cases, robots do not have to use as many foot steps as swinging discrete robots with a comparable number of legs. This should lead to increased efficiency.

Rolling systems tend to be less complex to build. The axis of motion about which rolling occurs, usually requires only a simple rotational actuator. No complicated linkages or long-stroke linear actuators (which tend to be more complex than rotational ones) are needed.

Roll-legged gaits are also usually more efficient than swing-legged locomotion. For the case of Polypod the roll-legged gaits are in general more efficient. In fact for 56 segments and 16 nodes, *every* roll-legged gait is more efficient than every swing-legged gait for all gaits implemented on Polypod.

Discrete vs. Continuous Footed: The discrete/continuous classification is not a hard division between two classes but rather an axis with gradations of discreteness or continuity. In one sense nothing is physically continuous, as at some level interactions can be viewed as contacts being made and broken (sometimes microscopic). A tracked vehicle is not continuous but is made up of discrete track segments.

A general advantage of continuous gaits over discrete gaits is that continuous gaits can achieve smoother motions over flat hard terrains. The result of this is that the continuous gaits usually have higher top speeds. This can be seen in most cases in Table 3.1 if the vehicles are scaled by their relative sizes.

As described earlier, one of the characteristics of continuous gaits is that a continuous vehicle's G-foot never need to leaves the ground. This is an advantage in planning the motions because there is no need to plan the collision free motions of the feet and legs between ground contact points.

Creators of walking robots often argue that wheeled vehicles are not as mobile on rugged environments as walking vehicles since walkers need only discrete contact points [Simmons 1991] [Raibert 1990]. While this seems obvious at first, it is not always the case. If we were to examine a vehicle that had wheels which had the same radius as the length of a leg on a walking vehicle, they would arguably be able to traverse the same types of terrain. An analysis of the effects of terrain on each of these classes is given in Chapter 6.

Big vs. Little Footed: For little footed gaits, a valid support is any point on the ground that can sustain the weight of the robot in contact with any point of a G-foot without sliding. Planning for static stability is then left to finding points on the ground that are within the range of motion of each G-foot to form a support polygon.

For big-footed gaits (which usually, though not always) have large area footprints, planning for static stability is done by finding several points on the ground that would be *inside* a footprint to form a support polygon.

The actual size of the footprint also effects the vehicles performance on soft terrain. In general, the larger the footprint, the better the performance in terms of speed, efficiency, and mobility [Bekkar 1969].

An interesting thing to note here is that most small insects have pointed feet. This may be attributable to the proportionately large frictional forces due to scaling effects as described in Chapter 7. A pointed foot in contact with the ground can be represented mechanically as a ball and socket joint having 3 DOF, while a larger foot would be represented as a rigid attachment. Experiments with Polypod have shown that rigid attachments with the ground make **SDB** locomotion more difficult.

3.4.2 Compound Gaits

Articulated: Single chain articulated gaits have several desirable features. They present a small front profile relative to their lengths, so they are well suited to travel in highly constrained areas. The profile allows them to fit between obstacles, and the length allows them to have a large reach to cross large obstacles. In Chapter refterrevalchap it is shown that indeed the top three gaits in each category for obstacle traversal is an articulated gait.

Hirose et al claim that the payload of articulated robots are higher than others in [Hirose 1990]. They scale the payload by the front profile of the vehicle rather than the overall weight or size, so this may not be suitable for all applications. For Polypod an array of articulated gaits results in the largest payload and carrying surface area for a given number of modules.

Hierarchical: Hierarchical combinations are an interesting way of generating new gaits, as given the right size of structure, any two gaits can be combined hierarchically. While articulated gaits are commonly found in nature and in mechanisms, hierarchical combinations are relatively rare. Some of the more interesting gaits in Chapter 5 are the result of hierarchical combinations.

While the effect of combining gaits by articulation does not change the efficiency or speed of the gait, hierarchical combinations typically do both. This is because when gaits composing the hierarchical combination move at the same time, the motions are additive.

Morphological: Morphological combinations can result in interesting gaits that have never been seen before such as the Rolling-Slinky gait. However, generating these gaits is not a straight forward matter as with the previous two compound classifications. The main usefulness of morphological combinations is the ability to move in different directions. The advantages over other gaits is not clear.

Overall, the fact that gaits can be combined is one important contribution of this taxonomy, it presents one way of generating new an interesting gaits.

Chapter 4

Polypod Locomotion Control

4.1 Introduction

This chapter presents a control method that will be used in the following chapter to implement many different modes of locomotion. Although this control is designed with locomotion in mind, it can still be used in many other applications. The key element in this method is simplicity.

To achieve any task with a robot built up of many modules, control becomes the first difficult problem. How does one control the many DOF? For robot arms which have more DOF than the workspace, the problem is called redundancy resolution. Typically researchers have dealt with less than ten DOF and have struggled to find ways to take advantage of the extra DOF such as obstacle avoidance and optimal dynamic performance for manipulators. With reconfigurable modular robots we may have hundreds or thousands of DOF.

An ideal control method for a reconfigurable modular system should be scalable to an infinite number of DOF. One of the advantages of having a reconfigurable modular robot is that in many instances, the robot can be scaled to fit the task, i.e. multiplying or dividing the number of modules while maintaining the same approximate shape. If the robot is to be scaled, the control method must also scale.

To this end we present a very simple scalable control method with one assumption: motions are quasi-static so scaling dynamic effects is not an issue.

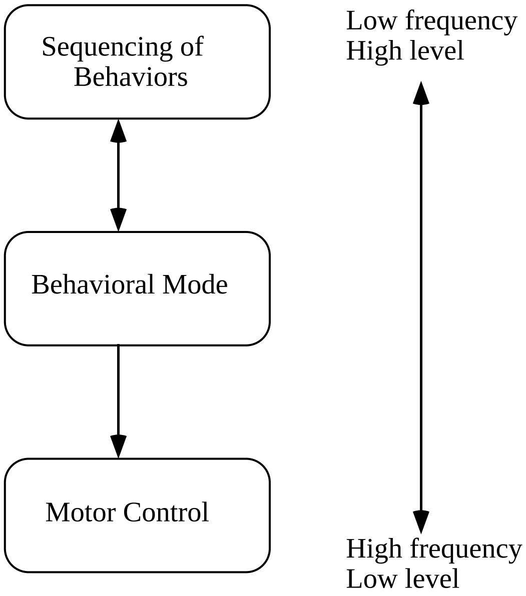


Figure 4.1: Hierarchical control

Overview: The control architecture has three hierarchical levels as shown in Figure 4.1. The lowest level runs the fastest, at about 200Hz in the current implementation of Polypod, the median level at 20Hz, and the highest is variable averaging about 0.2Hz.

The lowest level consists of a standard proportional-integral-derivative (PID) feedback control on each of the motors (but any other low level control could be substituted for performance). The median level consists of "behavioral modes" for each DOF[1]. This is the heart of the control method and sends desired position commands to the lower level. The highest level called the master control, decides which behavioral mode should run and when it should run.

There are two master control schemes presented. The first is the minimal synchronous master control scheme which makes no assumptions about the abilities of the low level control. The second is called the masterless control scheme which assumes that the low level control follows position commands exactly. The first method

[1] prismatic DOF or revolute DOF as defined in Section 2.3.2.

is used to implement some of the simple locomotion gaits. The second method is used to implement compound and turning gaits.

4.2 Low Level Motor Control

The position is sensed through potentiometers mounted on the inner and outer links and is fed to the computer via on chip A/D's (analog to digital converters). The computer reads this value and computes the level of the motor voltage determined by the following control law (a standard PID control):

$$\dot{\varepsilon} = e \tag{4.1}$$

$$V_m = -K_V \dot{e} - K_P e - K_I \varepsilon \tag{4.2}$$

where V_m is the motor voltage, e is the difference from the sensed position from the desired position, and ε is the integral of the tracking error $e(t)$. The values of the constants K_V, K_P, K_I are empirically chosen such that the control is stable for all static loads and that acceptable speed performance (quasi-static) is attained with no added load.

The system plant could be modeled as a second order system however the non-linearity of the linkages and the large amount of friction and viscous damping in the transmission make this very complex. Since we are not concerned with high-performance, the PID control suffices.

The low level control receives desired position commands from the behavioral control level in the RPjoint space. The low level converts these commands into the actuator space commands and executes them.

4.3 Behavioral Modes

We define a behavioral mode as a form of control applied to a DOF on a segment. Behavioral modes have a sign, that is, they may be positive or negative. Thus, each mode corresponds to two control laws. The control consists of just two basic

SUMMARY OF BEHAVIORAL MODES

	prismatic		revolute	
mode	$+$	$-$	$+$	$-$
ends	$D = 2.0$ $\theta = 0$	$D = .5$ $\theta = 0$	$D = 1$ $\theta = 45$	$D = 1$ $\theta =$ -45
springs	$D =$ -F/k	$D =$ F/k	$\theta =$-T/k	$\theta =$ T/k

D and θ are shown in Figure 2.4. F and T are the force and torque sensed, k is the spring constant.

Table 4.1: Steady state positions of a segment

behavioral modes: the *ends* mode, and the *springs* mode. A third mode is included for completeness, the *no* mode where nothing is done.

4.3.1 Ends Mode

A DOF in the *ends* mode moves at a constant speed in either the positive or negative direction until it reaches the joint limit, then it stops, signals the master computer and then does nothing. The speed at which the DOF moves is slow enough that we may ignore inertial effects. To achieve this motion, the behavior mode level of the control simply increases the desired position of the DOF at a constant rate as the commands are sent to the low level.

The control law for *ends* mode can be expressed as

$$\dot{x} = -\text{sgn}(\text{mode})C \tag{4.3}$$

where $\dot{x}$ is the velocity of the DOF in RPjoint space, C is a constant speed, and sgn(mode) is the sign of the mode.

A $+$*ends* mode applied to the prismatic DOF extends the segment, $-$*ends* compresses the segment. In Figure 2.4, $D = 2.0$ in. when extended, and $D = 0.5$ in. when compressed. A $+$*ends* mode applied to the revolute DOF moves θ to $+45$ degrees, $-$*ends* moves θ to -45 degrees. These are summarized in Table 4.1.

4.3.2 Springs Mode

In the springs mode, the DOF behaves basically like a spring. An easy way to simulate a spring would be for the actuators to apply a force based on the sensed displacement. Since the transmission in the actuator is not back-driveable, the controller instead moves to a position based on the force sensed. We are assuming motions to be quasi-static, so large damping factors are applied at the lowest level.

A DOF in the *springs* mode is applied the following control law:

$$x = -\frac{F}{\mathrm{sgn(mode)}} \tag{4.4}$$

where F is the force sensed, x is the desired position to which the low level controller will servo, k is a constant, and sgn(mode) is the sign of the behavioral mode. The value of k, unless otherwise specified, is not critical since the *springs* mode is often used to relieve internal stresses. For specific tasks, k must be chosen reasonably as will be shown for the Slinky gait in the next chapter.

For a $+$*springs* mode, the DOF will move as a spring, that is, a constant compression force will move the spring to a steady state position somewhat smaller than the initial position, as Hookes law predicts. For a $-$*springs* mode the spring constant can be considered as negative. For example, if a linear spring with a negative constant has a constant force applied to compress it, the spring will expand to a steady state position *larger* than the initial position (opposite from a spring). When the DOF reaches a joint limit, the master CPU is notified.

When characterizing a spring, two parameters are required, the spring constant k, and the natural length of the spring l. This is set depending on the sign of the mode and whether it applies to a prismatic or revolute joint. For a prismatic DOF with $+$*springs*, l = 2.0 in. which is when the segment is fully extended. For $-$*springs* , l = 0.5 inches when the segment is fully compressed. For a revolute DOF l=0 degrees for both $+$*springs* and $-$*springs* , the segment is straight ($\theta = 0$). These modes are summarized in Table 4.1.

Prismatic and Revolute Coupling: It should be noted that *ends* mode cannot exist on both the revolute and prismatic DOF at the same time. This is because the joint limits are coupled. When the prismatic DOF is at a joint limit, the revolute DOF is at $\theta = 0$ and has zero range of motion. When the revolute DOF is at a joint limit the prismatic DOF is at $x = 1$in. and has zero range of motion. When a conflict between DOF occurs due to the joint limit constraints, one DOF will have priority over the other according to what mode it is in. *Ends* modes have the highest priority, *springs* modes the next, and lastly no mode has the lowest. In the event of both DOF having the same mode, the prismatic has priority over the revolute (choice of priority is arbitrary, but should be defined in order to design gaits robustly).

4.4 Master Control Level

The highest level of control decides which behavioral mode runs and when it is run. Each task is composed of a sequence of behavioral modes running on a set of DOF and can be represented by a table. We call this the **gait control table**. Each element in the table corresponds to a behavioral mode for a DOF at a step in the gait.

4.4.1 Minimal Synchronous Master Control

Each mode runs until it reaches a joint limit, or the master computer tells the segment that the DOF should change modes. When a DOF reaches a joint limit, it signals the master that it has done so. A set of joint limit signals comprises a trigger condition. When a specified trigger condition occurs (a set of DOF on specified segments have reached joint limits) the master computer signals all the segments to use the next set of behavioral modes.

If we look at Table 4.2, each row corresponds to the behavioral modes for all segments in the configuration during the first step. Each column corresponds to the sequence of behaviors for one segment. The segments start out with the modes in step 0, and stay in those modes until the segments listed at the end of the row reach a joint limit. Then the segments change to the modes listed in the next step. When the last step is reached, the cycle starts over again at step 0. In the table, $\rightarrow$ corresponds to $+$*springs* for revolute DOF and *no* mode for prismatic DOF, and $\Leftarrow$ corresponds to

	Segment number																
step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	trigger
0	→	→	→	→	⇐	⇐	⇐	⇐	→	→	→	→	⇐	⇐	⇐	⇐	5,13
1	⇐	→	→	→	→	⇐	⇐	⇐	⇐	→	→	→	→	⇐	⇐	⇐	6,14
2	⇐	⇐	→	→	→	→	⇐	⇐	⇐	⇐	→	→	→	→	⇐	⇐	7,15
3	⇐	⇐	⇐	→	→	→	→	⇐	⇐	⇐	⇐	→	→	→	→	⇐	8,16
4	⇐	⇐	⇐	⇐	→	→	→	→	⇐	⇐	⇐	⇐	→	→	→	→	9,1
5	→	⇐	⇐	⇐	⇐	→	→	→	→	⇐	⇐	⇐	⇐	→	→	→	10,2
6	→	→	⇐	⇐	⇐	⇐	→	→	→	→	⇐	⇐	⇐	⇐	→	→	11,3
7	→	→	→	⇐	⇐	⇐	⇐	→	→	→	→	⇐	⇐	⇐	⇐	→	12,4
8	→	→	→	→	⇐	⇐	⇐	⇐	→	→	→	→	⇐	⇐	⇐	⇐	13,5

Table 4.2: Gait control table for Rolling-Track locomotion

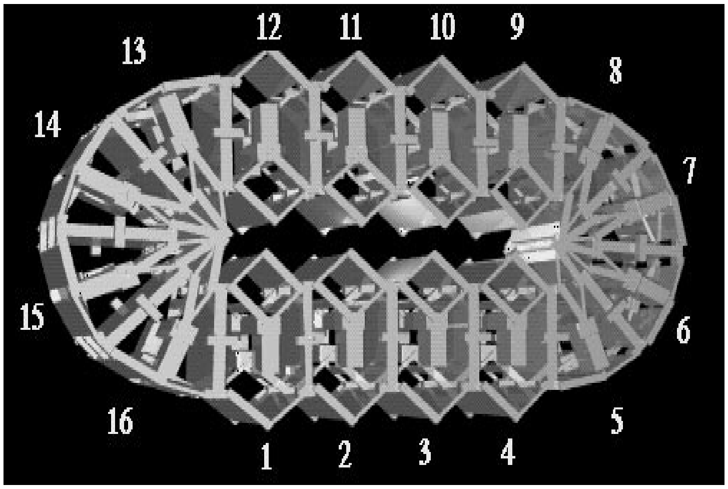


Figure 4.2: Rolling-Track locomotion

		prismatic DOF				
	Mode	*no*	*+springs*	*-springs*	*+ends*	*-ends*
Revolute	*no*		↑	↓	⇑	⇓
D	*+springs*	→	→↑	→↓	→⇑	→⇓
O	*-springs*	←	←↑	←↓	←⇑	←⇓
F	*+ends*	⇒	⇒↑	⇒↓	⇒⇑	⇒⇓
	-ends	⇐	⇐↑	⇐↓	⇐⇑	⇐⇓

Table 4.3: Short-hand for behavior modes combinations in a segment

-*ends*, *no* respectively. The symbols are short hand for the 21 possible combinations of modes listed in Table 4.3. This particular table shows the mode sequence for the Rolling-Track simple locomotion gait which is shown in Figure 4.2.

This method can be used to simulate all of the simple gaits in the Polypod locomotion chapter.

4.4.2 Masterless Control

Typically, a mobile robot will be given a trajectory to follow, that is a path and a time value for each position on the path. Since the minimal synchronous master control has no concept of time, the architecture must be augmented to accomplish trajectory following. Specifically we must add a new form of continuous synchronization.

The minimal synchronous master control had a minimal form of synchronization (thus the name). The master synchronized motions by starting each mode at the beginning of a new step. Within one step each module could move at any arbitrary speed.

Synchronization The simplest form of continuous synchronization would be to have one clock to which all segments refer. Since we want to limit the communications between segments, we assume all internal clocks are synchronized. We then add a time parameter to the architecture by specifying a length of time for each behavioral mode. The control becomes a sequence of modes and corresponding times that each mode should run before completion. For the *ends* modes this time indirectly specifies

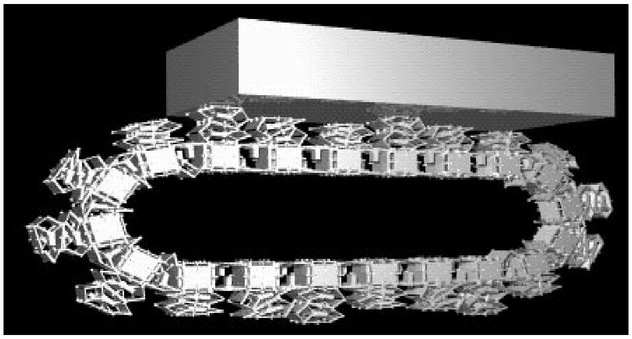


Figure 4.3: Rolling-Track / Caterpillar hierarchical combination

the speed at which the DOF should move. For the *springs* modes, this means the length of time the DOF should act as a spring.

With the synchronized internal clocks and the assumption that no other unexpected events occur, the master computer is no longer needed. Each segment can keep its own list of behaviors and times for a task.

Two different time parameterizations can be seen graphically in Tables 4.5 and 4.4. These are examples of two gaits using one robot configuration combining the Rolling-Track and a gait called Caterpillar hierarchically. In the first table, the Caterpillar gait manipulates an object while the robot translates with the Rolling-Track. In the second table, the Caterpillar gait moves quickly in the opposite direction that the Rolling-Track moves, with a net motion backwards. This gait is called "The Moonwalk."

In the tables, the vertical space represents the time associated with each mode. Each row is a step, the horizontal listing of ⇒'s and ⇑'s represents modes for corresponding segments. An ⇒ represents revolute *ends* mode with no mode on the prismatic, and ⇑ represents prismatic *ends mode* with no mode on the revolute. All modes on the Rolling-Track segments are positive.

The feet of the caterpillar segments go from a free position ⇓⇓ to a forward contact position ⇐⇒ and sweep back to a rear contact position ⇒⇐. The feet on top of the Rolling-Track in Table 4.4 move in the opposite direction to pass the object toward the back of the track.

In The Moonwalk, the vertical spacing between each step of the Rolling Track

Caterpillar Segments				Rolling-Track Segments
Top feet		Bottom feet		
⇐ ⇒	⇓ ⇓	⇒ ⇐	⇓ ⇓	⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒
	⇒ ⇐		⇐ ⇒	
⇓ ⇓	⇐ ⇒	⇓ ⇓	⇒ ⇐	⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒
⇒ ⇐		⇐ ⇒		
⇐ ⇒	⇓ ⇓	⇒ ⇐	⇓ ⇓	⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒
	⇒ ⇐		⇐ ⇒	
⇓ ⇓	⇐ ⇒	⇓ ⇓	⇒ ⇐	⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒
⇒ ⇐		⇐ ⇒		
⇐ ⇒	⇓ ⇓	⇒ ⇐	⇓ ⇓	⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑
	⇒ ⇐		⇐ ⇒	
⇓ ⇓	⇐ ⇒	⇓ ⇓	⇒ ⇐	⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑
⇒ ⇐		⇐ ⇒		
⇐ ⇒	⇓ ⇓	⇒ ⇐	⇓ ⇓	⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑
	⇒ ⇐		⇐ ⇒	
⇓ ⇓	⇐ ⇒	⇓ ⇓	⇒ ⇐	⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑
⇒ ⇐		⇐ ⇒		
⇐ ⇒	⇓ ⇓	⇒ ⇐	⇓ ⇓	⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑

Table 4.4: Rolling-While-Carrying an object behavioral modes and timing

Caterpillar Segments		Rolling-Track Segments
odd feet	even feet	
⇐ ⇒	⇓ ⇓	⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒
	⇒ ⇐	
⇓ ⇓	⇐ ⇒	
⇒ ⇐		
⇐ ⇒	⇓ ⇓	
	⇒ ⇐	⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒
⇓ ⇓	⇐ ⇒	
⇒ ⇐		
⇐ ⇒	⇓ ⇓	
	⇒ ⇐	
⇓ ⇓	⇐ ⇒	⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒
⇒ ⇐		
⇐ ⇒	⇓ ⇓	
	⇒ ⇐	
⇓ ⇓	⇐ ⇒	
⇒ ⇐		⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒⇒⇒⇒⇑⇑⇑⇑⇑⇑⇒
⇐ ⇒	⇓ ⇓	etc.

Table 4.5: The Moonwalk behavioral modes and timing

segments is much larger than the Caterpillar segments. This results in the Rolling-Track gait moving slower than the Caterpillar gait and allows the robot to achieve a net motion in the opposite direction of the Rolling-Track. Note that the Rolling-While-Carrying gait has a relatively smaller vertical distance between Rolling-Track segments.

One result of not needing a master is that there is no need for communication between segments. For scaling purposes, this architecture is ideal as the communications limitation is removed. The consequence of this though, is a loss of robustness since the segments run "open-loop." If a master is available, it can be used to monitor the progress of a task to regain some fault tolerance. Or it can work on higher level problems, such as determining optimal configurations for given tasks, etc.

4.4.3 Generating the Gait Control Table

All of the gait control tables in this dissertation were created manually using five steps.

1. The number of modules and arrangement of them were selected.
2. A sequence of behaviours for each DOF is chosen and entered in a gait control table.
3. The resulting gait is viewed in simulation.
4. The gait control is modified.
5. Steps 3 and 4 are repeated until the desired gait is viewed.

The simplicity of this method is seen by the time it takes to develop seemingly complex gaits. Typically the above sequence would take one or two hours to develop gaits as complex as the Moonwalk with the aid of a real-time simulator. The Moonwalk involves over 120 DOF.

Another interesting approach would be to generate the gait control table automatically. Karl Sims has explored "artificial evolution" [Sims 1991]. One of his examples includes the automatic generation of modular "animals" or cellular automata that can swim and others that locomote on flat ground. This was done by simulating evolution using genetic algorithms [Spencer 1994]. The manual technique above could be modified by having step 2 be a random selection, and step 5 use a genetic algorithm technique.

Chapter 5

Polypod Locomotion

5.1 Introduction

As this chapter will show, Polypod can perform each of the four classes of big-footed simple locomotion gaits shown in Figure 3.1. By the articulation property shown in Section 3.10 any big-footed gait can be transformed into a little-footed gait by combining several vehicles. Thus Polypod is capable of all eight classes of simple statically stable locomotion.

Still, some of the more interesting gaits are a result of combining gaits. This chapter also includes examples of each type of compound gait. In addition, it is often desirable for the vehicles to be able to turn. We present two forms of turning which are implemented or simulated on Polypod.

In total there are eleven statically stable locomotion gaits that are shown with Polypod. Eight of these gaits are called the Rolling-Track, Slinky, Earthworm, Caterpillar, Cater-Cater, Slinky-Slinky, the Moonwalk, and Spider gait. Five additional gaits are presented which are variations of the Slinky (3-Segment Slinky, Cartwheel), Caterpillar (Turning Caterpillar), Rolling-Track (Turning-Loop) and the Moonwalk (Rolling-While-Carrying) gaits. All of the gaits except the Spider gait use the control methods previously described.

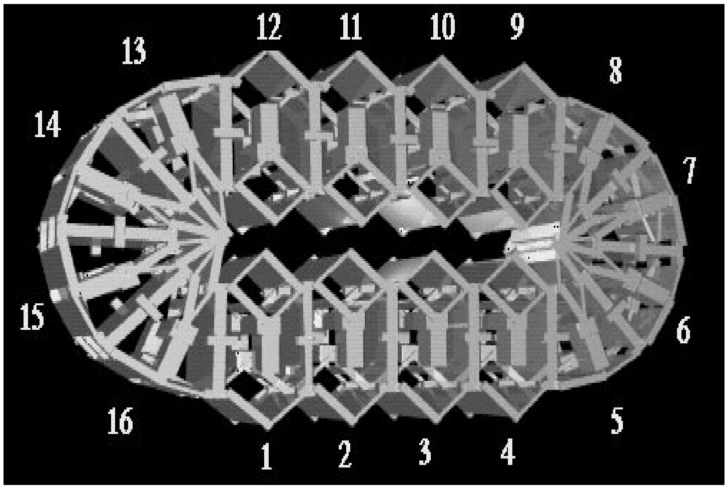


Figure 5.1: Rolling-Track locomotion

5.2 Simple Straight Line Gaits

The following are the simple straight line methods of moving some incremental distance forward. Each of the gaits in this section are big-footed gaits, with the exception of the Spider gait.

5.2.1 Rolling Continuous

Rolling-Track: Figure 5.1 shows the rolling-track gait with 16 segments. The modules form a loop so the robot can roll as a track on a tracked vehicle. This loop configuration can be considered as four sections of segments: four forming a line, four forming a half circle, four forming another line, and four forming another half circle.

Each DOF is in one of the behavioral modes depending on its location in the loop according to Table 5.1. The gait-control table is shown in Table 4.2 which explicitly shows the behavioral modes for this gait at each step.

To achieve locomotion using the minimal synchronous master control, the master computer notifies the segments at the end of each straight section (number 4 and 12 in figure 5.1) to change modes to *+ends* on the revolute DOF, and segments at the end of each curve (8 and 16) to change modes to *+springs* on the revolute DOF.

When both segments changing to *ends* mode (4,12) have finished moving to the

	straight	curve
prismatic DOF	+*springs*	+*springs*
revolute DOF	+*springs*	+*ends*

Table 5.1: Behavioral modes for Rolling-Track locomotion

joint limits, the trigger condition is satisfied and the robot will be again in a configuration equivalent Figure 5.1. The master then notifies the next set of segments (1,5,9 and 13) to change modes. This shifting of modes can be seen in Figure 4.2 by the diagonal pattern of symbols.

The *springs* mode on the straight section of the robot serves two purposes. It alleviates the high internal forces often generated when trying to control closed chains with imperfect position control, and it allows the robot to conform to non-flat terrain for better stability and traction.

The rolling nature of the gait clearly places it in the rolling class of locomotion. We classify it as continuous since the sides of the segments are used, and for soft terrain, the entire outside surface of the loop may contact the ground at some point with out interrupting locomotion. Thus the robot has one G-foot which never leaves contact with the ground. This places the robot in the continuous and big-footed category, (**RCB**).

This rather simplistic gait can easily be scaled to robots with more modules. Nodes or segments may be inserted anywhere in the loop as long as the segments in the straight sections will not reach their joint limits as they move in *springs* mode to compensate for the irregularities.

5.2.2 Rolling Discrete

There are three rolling discrete gaits presented here, each is a version of the basic Slinky gait described below.

Slinky: The Slinky mode of locomotion is shown in Figures 5.2 and 5.3. In this mode there is a single chain with both ends acting as legs and can thus be considered

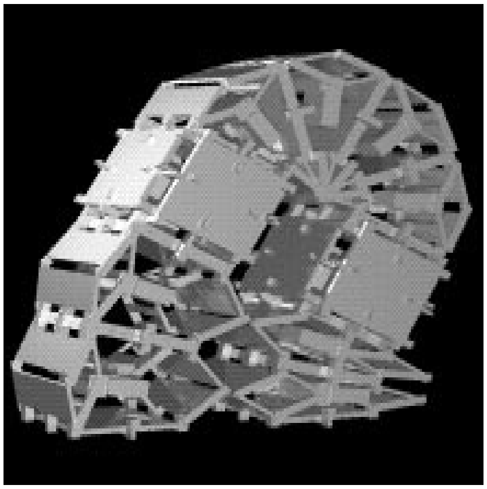


Figure 5.2: Slinky locomotion step 1

	Segment number								
step	1	2	3	4	5	6	7	8	trigger
0	→↑	→↑	⇒	⇒	⇒	⇒	→↓	→↓	7,8
1	←	←	⇐	⇐	⇐	⇐	→↓	→↓	3,4,5,6
2	→↓	→↓	⇐	⇐	⇐	⇐	←	→↑	1,2
3	→↓	→↓	⇒	⇒	⇒	⇒	←	←	3,4,5,6

Table 5.2: Gait control table for Slinky locomotion

a biped walker. There are two steps to this gait. The first one shifts the weight from the rear foot to the front foot which is typical of a biped walker. Instead of swinging legs back and forth, the second step brings the swing foot over top to be placed in front of the robot. Static stability is achieved by assuring the vertical projection of the center of mass is within the width of the foot. Thus this gait is classified as **RDB**.

There are three sections to the robot each with a behavioral mode. The supporting foot section is segments 1 and 2 in Figures 5.2 and 5.3. The next four segments 3, 4, 5 and 6 make up the middle curve. The free foot is composed of the last two segments, numbers 7 and 8. The behavioral modes for these sections in each step of the gait are shown in Table 5.2.

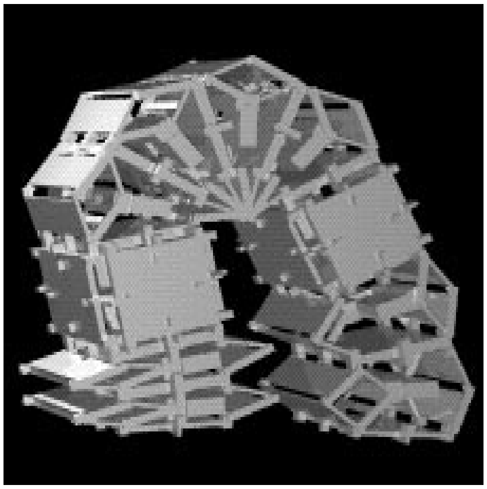


Figure 5.3: Slinky locomotion step 2

The supporting foot with $-$*springs* on the revolute DOF keeps the vertical projection of the center of mass within the width of the foot. How the $-$*springs* mode does this is shown in Appendix B. The free foot pushes off from the ground with a $+$*ends* mode on the prismatic DOF while a $+$*springs* mode on the revolute DOF keeps it flat to the ground.

While the Rolling-Track is easily scaled to large numbers of modules, this one is not. More segments may be added to the length, but as this mode relies on balancing the robot on its end, the longer the robot becomes, the less stable it will be in the middle of its gait.

Four-arm Cartwheel: The Four-Arm Cartwheel uses the same strategy as the Slinky gait. It is pictured in Figures 5.4 and 5.5 where it moves to the left by rolling end over end.

As in the Slinky gait, there are essentially two types of steps, shifting the weight and placing a new foot. When a new foot is placed, the previous two feet are still on the ground, as in Figure 5.4 however, most of the weight is on the middle foot. When the robot rotates one step, as in Figure 5.5 the upper leg extends to shift the weight onto the newer ground contact leg and to prepare for contacting the ground as this upper leg is the next leg to make contact. The gait-control table is shown in

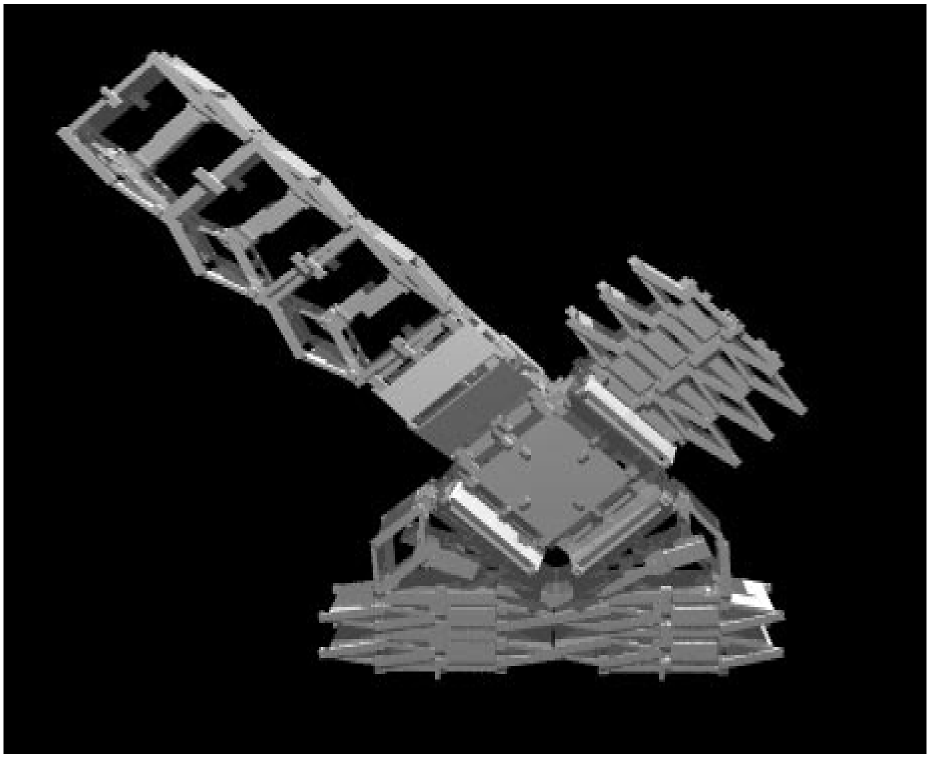


Figure 5.4: Weight shifting step for the four-arm cartwheel gait

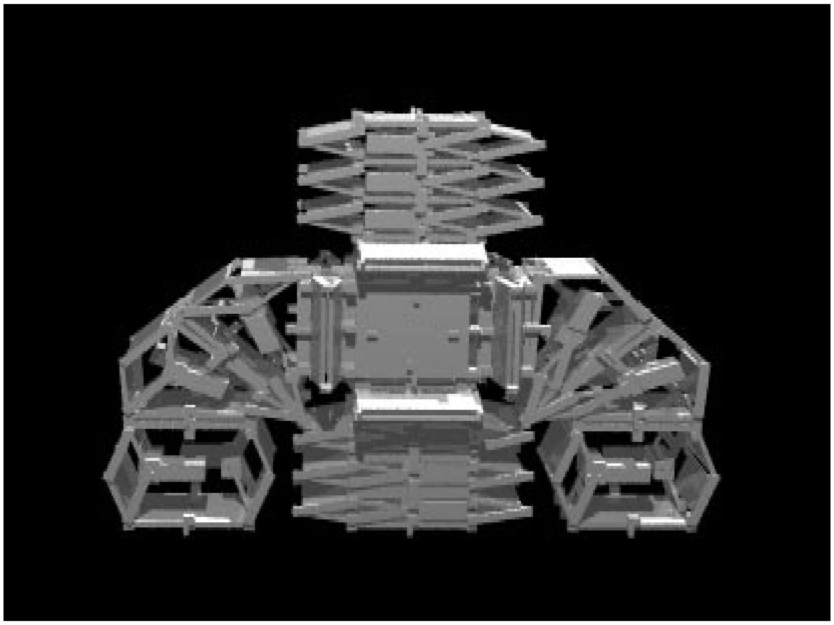


Figure 5.5: New leg placement step for the four-arm cartwheel gait

step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	trigger
0	←⇓	⇓	⇓	⇓	→⇑	⇐	⇐	⇓	⇓	⇓	⇓	⇓	→↑	⇒	⇒	⇓	all ¬(13)
1	→↑	⇓	⇐	⇓	⇓	⇓	⇓	⇓	⇑	⇑	⇑	⇑	→↑	⇓	⇒	⇓	all ¬(1,13)
2	→⇑	⇐	⇐	⇓	⇓	⇓	⇓	⇓	→↑	⇒	⇒	⇓	←⇓	⇓	⇓	⇓	all ¬(9)
3	⇓	⇓	⇓	⇓	⇑	⇑	⇑	⇑	→↑	⇓	⇒	⇓	→↑	⇓	⇐	⇓	all ¬(9,13)
4	⇓	⇓	⇓	⇓	→↑	⇒	⇒	⇓	←⇓	⇓	⇓	⇓	→⇑	⇐	⇐	⇓	all ¬(5)
5	⇑	⇑	⇑	⇑	→↑	⇓	⇒	⇓	→↑	⇓	⇐	⇓	⇓	⇓	⇓	⇓	all ¬(5,9)
6	→↑	⇒	⇒	⇓	←⇓	⇓	⇓	⇓	→⇑	⇐	⇐	⇓	⇓	⇓	⇓	⇓	all ¬(1)
7	→↑	⇓	⇒	⇓	→↑	⇓	⇐	⇓	⇓	⇓	⇓	⇓	⇑	⇑	⇑	⇑	all ¬(1,5)

Table 5.3: Gait control table for four-arm Cartwheel locomotion

Table 5.3.

In all cases one leg is intended to support the weight of the robot. This weight bearing leg has the segment in contact with the ground with $-$*springs* mode on the revolute DOF to balance the robot just as in the Slinky gait. The other legs in contact with the ground have $+$*springs* modes so that they can add stability to the robot.

Three-Segment-Slinky: The three-segment-Slinky was the first locomotion gait that was implemented with the physical modules. It is similar to the Slinky mode of locomotion except that it is uses only three segments. The main significance of this gait is that this is the minimum number of segments needed to achieve locomotion.

This motion is the only gait that is fixed in the number of modules and relies on some non-static statically stable motion. Note also that no node is mounted, nor can any node be added and still maintain locomotion. Thus power for this mode must be supplied off-board.

Figure 5.6 shows the sequence of motions. After the third step the robot uses the momentum to carry it back to the initial state (the last picture in the sequence). Every other motion is still quasi-static.

Each segment has *ends* mode on the revolute DOF. The prismatic DOF is in *no* mode. Table 5.4 lists the modes for each revolute DOF in Figure 5.6.

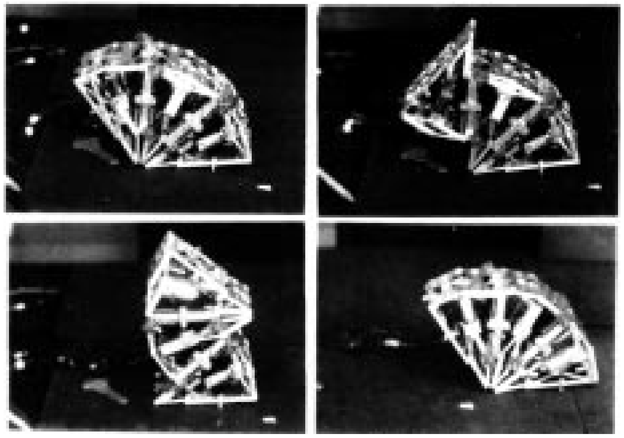


Figure 5.6: Three-segment-slinky locomotion

step	seg 1	seg2	seg 3	trigger
0	$\Leftarrow$	$\Rightarrow$	$\Rightarrow$	1
1	$\Leftarrow$	$\Leftarrow$	$\Rightarrow$	2
2	$\Leftarrow$	$\Leftarrow$	$\Leftarrow$	3
3	$\Leftarrow$	$\Leftarrow$	$\Rightarrow$	3
4	$\Leftarrow$	$\Rightarrow$	$\Rightarrow$	2
5	$\Rightarrow$	$\Rightarrow$	$\Rightarrow$	1

Table 5.4: Gait control table for three-segment-slinky locomotion

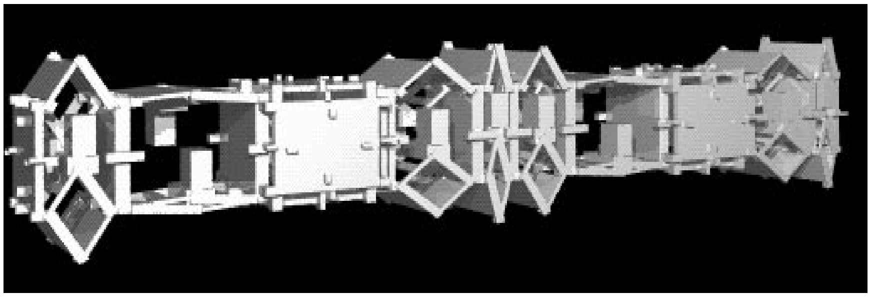


Figure 5.7: Earthworming locomotion

step	1	2	3	4	5	6	7	8	trigger
0	⇓	⇑	⇑	⇓	⇓	⇑	⇑	⇓.	all
1	⇑	⇑	⇓	⇓	⇑	⇑	⇓.	⇓	all
2	⇑	⇓	⇓	⇑	⇑	⇓.	⇓	⇑	all
3	⇓	⇓	⇑	⇑	⇓.	⇓	⇑	⇑	all

Table 5.5: Gait control table for earthworm locomotion

5.2.3 Swinging Continuous

Earthworming: Figure 5.7 shows the sequence for four segments taking advantage of the way the sides of the segments expand while the segment compresses. The sequential compressing of segments results in a traveling wave earthworm-like[1] motion similar to some described in [Chirikjian 1992]

Each segment has *ends* mode on the prismatic DOF. The revolute DOF may be in either +*springs* or *no* mode. Table 5.5 lists the modes for each prismatic DOF of Figure 5.7.

The trigger condition for each step is to have all prismatic DOFs reach their joint limit. Note that forward motion only occurs during the first and second step.

The "wave" of the traveling wave can be considered the pattern of arrows moving diagonally across Table 5.5 (a '.' has been introduced in Table 5.5 to emphasize this wave). This repeated pattern is made up of four segments. The control of each

[1] Previous papers by the author have referred to earthworm gait as the caterpillar gait, and the current caterpillar gait as the Polypod gait. The names have been changed as the previous ones were misnomers and caused some confusion.

segment is then dependent on its position in the chain modulo four.

Note also that between each successive step only half of the segments change mode, and of those that do, half compress and half extend at the same time at the same speed so each four segment group is constantly moving in the null space of the Jacobian. The Jacobian in this case is the time-varying linear transformation from the RPjoint space to Cartesian space. Since the robot is moving in its null space the contacts with ground do not move, and nodes may be interspersed throughout the robot without affecting the motion. Other segments may be attached to these nodes to act as arms or tactile sensors.

With a minimum of seven segments, the robot is guaranteed to have the vertical projection of the center of gravity falling between the two extreme contact points. With four segments, the robot is not completely statically stable. However locomotion can still occur with some portions of the robot sliding on the ground.

By having the revolute DOF in *+springs* mode, the robot may follow non-flat terrain. The weight of the robot will cause it to conform to the shape of the terrain.

This motion is ideal for traveling in a round pipe of diameter 7.18 cm (2.83") to 14.37 cm (5.66"). The expanding sides of the segment may be used to grip the inside walls by substituting *−springs* mode for the *−ends* mode on all the prismatic DOFs in Table 5.5. Turning would be achieved by using *+springs* mode on the revolute DOFs as for terrain following, and could even follow curves out of the plane by mounting alternate segments perpendicular to the plane.

Just as in the Rolling-Track gait, this gait contacts the ground on the side of segments and so like the Rolling-Track, we will consider this gait to be continuous. The one G-foot does not have a net rotation so this gait is classified as **SCB**.

5.2.4 Swinging Discrete

Caterpillar locomotion: Figure 5.8 shows a robot configuration with six G-feet. Caterpillar locomotion may have arbitrarily many sets of modules acting as feet and resembles one half of a walking caterpillar [2]. Each leg/foot group is composed of 1

[2] Previous papers by the author have referred to caterpillar gait as the Polypod gait.

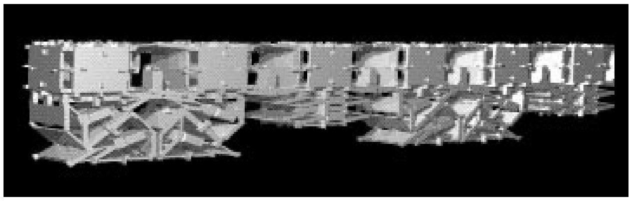


Figure 5.8: Caterpillar locomotion with six feet

step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	trigger
0	⇓	⇓	⇑	⇒	⇐	⇑	⇐	⇒	⇑	⇓	⇓	⇑	⇒	⇐	⇑	⇐	⇒	⇑	all
1	⇒	⇐	⇑	⇐	⇒	⇑	⇓	⇓	⇑	⇒	⇐	⇑	⇐	⇒	⇑	⇓	⇓	⇑	all
2	⇐	⇒	⇑	⇓	⇓	⇑	⇒	⇐	⇑	⇐	⇒	⇑	⇓	⇓	⇑	⇒	⇐	⇑	all

Table 5.6: Gait control table for six-footed Caterpillar locomotion

node, and two segments.

There are three steps to the cyclic motion of a foot:

1. Moving a rear G-foot to a clear position.

2. Moving the G-foot from the clear position to a forward ground contact position

3. Moving the G-foot from the forward position back to the rear ground contact position.

The two segments underneath a node run correspondingly in $+\mathrm{r}ends/-\mathrm{r}ends$, $-\mathrm{p}ends/-\mathrm{p}ends$, and $-\mathrm{r}ends/+\mathrm{r}ends$ in that sequence.

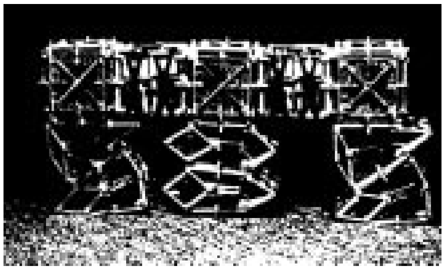


Figure 5.9: Caterpillar locomotion with three feet

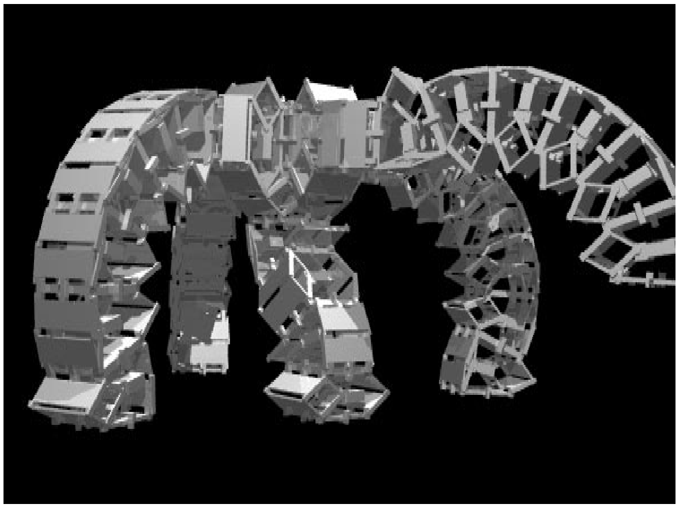


Figure 5.10: Spider Gait configuration

By staggering the steps we can assure static stability is maintained. There are many step patterns that can be used. For a three step gait, one foot at each position, six feet is the minimum number to assure stability with two feet in contact with the ground. The gait-control table for is shown in Table 5.6. For step patterns which have a different ratio of speed for each leg, the masterless architecture can be used.

A modified form of this gait was implemented using eight segments and three nodes and can be seen in Figure 5.9 and a video [Yim Video 1994].

5.2.5 Swinging Discrete Little-footed

The spider gait is unlike any of the previous gaits in that is uses a different type of control. The configuration in Figure 5.10 shows a five long-legged robot. The placement of each foot needs to be planned, and the motion of each leg needs to be planned explicitly. On flat terrain the concerns are that the robot is statically stable at all times, the legs do not collide with each other or themselves, and also that motion occurs in the desired direction. On rough terrain, the robot must also plan where it places each foot and the collision free motion of the foot to that point.

The tools to achieve the Spider gait consist of an inverse kinematics method, a motion strategy for each leg using this inverse kinematics method, and gait planner to plan the sequence of leg motions and foot placements.

The configuration in Figure 5.10 consists of a node from which five legs emanate. The four legs around the perimeter consist of 8 segments all of which act in a vertical plane except one segment near the node. This segment is used to swing the leg from side to side. The others are used to place the distal end of the leg some height and distance away from the node. In this configuration the Jacobian of each leg has rank four at all times. That is, the distal end of the leg cannot be moved in any arbitrary fashion, most notably the foot cannot instantaneously rotate about a vertical axis. The result is that as the robot translates, each foot must twist (and thus slide) on the ground.

The consequence of this sliding is discussed later in this chapter. To avoid this twisting, a different leg configuration may be chosen depending on the inverse kinematics method chosen. Three inverse kinematics methods are shown in Appendix A. One is for the planar 3 DOF case, one for the full 3D 6 DOF case using sets of 3 segments, one for the full 3D 6DOF case using sets of 4 segments. Each method maps the series of segments into arcs of circles.

For the case of Figure 5.10 the planar inverse kinematics method is used for the 6 segments on the distal end. This method maps the segments onto two arcs of circles. For example the two arcs on the raised leg consists of one segment arcing up, and the next 5 arcing down.

5.3 Compound Gaits

These next set of gaits are classified as compound gaits. We present three forms of articulated compound gaits, two hierarchical compound gaits, and two morphological gaits. The two hierarchical gaits presented also classify as “exotic” gaits (defined to be interesting gaits, but not necessarily efficient or useful).

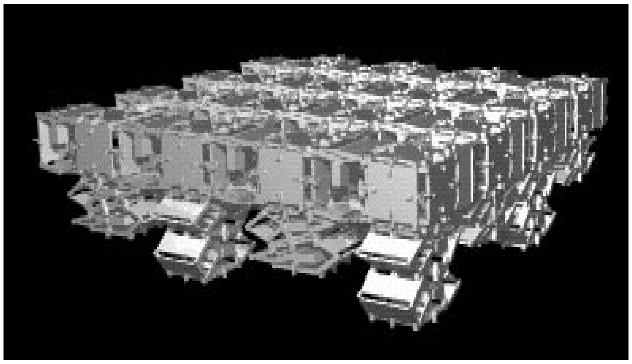


Figure 5.11: Cater-Cater locomotion in 2D

5.3.1 Articulated

The first pair of gaits, Earthworm and Caterpillar, are in the traditional sense of articulated robots, that is a single chain. The next gait, Cater-cater extends this by articulating several articulated chains along another Cartesian direction.

Earthworm, Caterpillar: Two gaits presented in the simple gaits section can also classify as articulated gaits. These are the Earthworm and Caterpillar gaits. In their simple gait form, they would consist of the minimal number of modules to achieve locomotion. In their compound form they are scalable to arbitrarily many modules.

Strictly speaking the Rolling-Track gait is not an articulated compound gait since it cannot be separated into simple gaits by removing joints. However many of the properties of articulated gaits apply to this gait. One way in which the Rolling-Track differs from articulated gaits is in turning as discussed in the turning gait section of this chapter.

Cater-cater: Figure 5.11 shows a 4x4 array of nodes and segments. In this array, the plane of action of every other foot is perpendicular to each other. That is every other foot alternates between acting in a forward/backward motion and right/left motion. By raising all the feet acting in a right/left motion, it can move forward or backward, and vice-versa. This action is shown in [Yim video 1994].

A clear use for this mode would be in transporting objects as it has a large flat

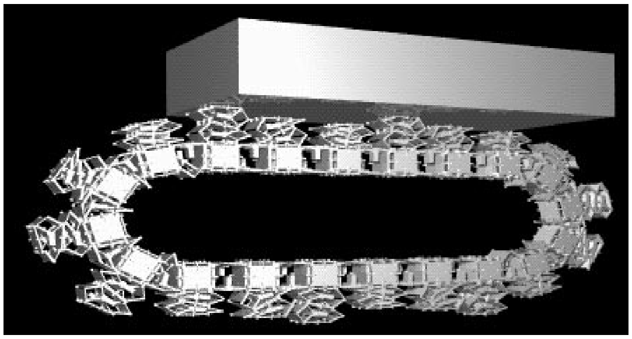


Figure 5.12: Rolling-While-Carrying

surface to carry things. The more feet it has, the larger and heavier the object can be.

5.3.2 Hierarchical

Here we present two exotic gaits, "The Moonwalk," and "Rolling-While-Carrying." Both of these gaits use a combined caterpillar gait and Rolling-Track gait as in Figure 5.12. The same type of combined gait control is used but the ratio of gait speeds (caterpillar:rolling) is different, 3:2 and 1:2 respectively.

The Moonwalk: The Moonwalk is a dance step made popular by Michael Jackson where there is an appearance of the dancer walking forward while in actuality he is moving backwards. This is achieved with Polypod by having the robot roll forward in the Rolling-Track mode thus appearing to move forward, while the caterpillar feet move more quickly in the opposite direction with the net effect that the robot moves backwards. The timing of this mode was shown previously in Table 4.5.

Rolling-While-Carrying: The interesting point about this motion is that the object cannot be held in a single grasp, otherwise it will fall to the ground and be trampled. Instead, the object must be passed from grasp to grasp continuously as the robot rolls.

As stated earlier, Robot manipulation and locomotion can be considered equivalent. With gravitationally stable locomotion, as presented here, force closure is

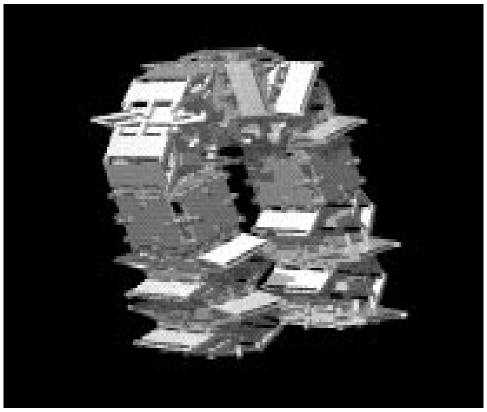


Figure 5.13: Slinky-Slinky gait

required for stability. These locomotion modes can be used for manipulating large objects through force closure as well.

In order to keep the object supported at a constant position above the robot, the object must be moved at a rate equal and opposite to the rate of motion that the rolling achieves. A similar motion to the Moonwalk can be used. If the caterpillar feet move at the same speed but opposite direction as the Rolling-Track, the robot will have no net motion and the object will be continuously supported above the robot. By having the feet on the lower half of the robot move at a different rate, the robot may have forward motion. The timing of this mode was shown previously in Table 4.4.

This motion along with the Moonwalk is simulated and presented in [Yim video 1994].

5.3.3 Morphological

Two morphological gaits are included here, the Slinky-Slinky gait and the Rolling-Slinky. Both gaits use two simple gaits that locomote in perpendicular directions to each other.

Slinky-slinky: The Slinky-Slinky gait consists of alternating segments mounted perpendicular to each other as in Figure 5.13.

The robot is treated as the combination of two simple gaits, both of them slinky gaits, however each moves perpendicular to the other. They are combined by alternating segments as if the two robots were shuffled together like a deck of cards. Both simple gaits cannot move at the same time. Thus each step is in one of two perpendicular directions. It may be possible for all segments to move at the same time to achieve motion along a 45 degree diagonal, however any other combination is not possible without losing stability given our simple behavioral control strategy. The robot changes direction by forming a straight line pointing up. This is the one configuration shared by both simple gaits.

Rolling-Slinky: The Rolling-Slinky gait is the combination of the Rolling-Track and the Slinky gait. The global form of the robot is that of the Rolling-track with alternating segments as in the Turning-Loop except that the robot takes the shape of rounded rectangle rather than a flattened oval.

Slinky locomotion occurs with the segments on each vertical portion of the robot that move perpendicular to the rolling-track locomotion. Both vertical portions must move at the same speeds to guarantee stability. If *+springs* modes are running on the revolute DOF of the segments between the two vertical segments, then some discrepancy is allowed between the motions of each vertical section. This would also allow concurrent rolling and Slinky locomotion.

5.4 Turning Gaits

Before presenting the turning gaits, we must mention the effects of sliding on locomotion.

While controlled sliding may have a usefulness in dynamic locomotion such as ice skating, it is generally something to avoid in statically stable locomotion. Robot induced sliding occurs when several ground contact positions are moved relative to each other. Nominally each ground contact should remain fixed unless the G-foot breaks contact. Sliding causes wasted energy and motor stress and can be severe in the case of locomotion on carpet or rough terrain.

In order to maintain a no slip constraint, all modules between two ground contact positions must maintain the same relative position. In other words, if we consider the modules which form a serial chain from one ground contact position to the next adjacent ground contact position as a robot arm, and assume the ground contact position to be rigid contact, then as the full robot locomotes, this robot arm portion can only move in the null space of its Jacobian.

For example, with the straight-line earthworm and caterpillar gait, this condition is met. The vertical, horizontal and revolute DOF within the plane of action are decoupled and all motions within steps can be straight lines in the RPjoint space. We can use *ends* modes since the motions are constant speed within one step, and are thus straight line motions in the RPjoint space. For motions in the null space which are not straight lines, (i.e. *ends* mode can not be used) +*springs* modes may be used assuming that static friction forces can support the forces needed to move the springs. To remain in the null space, each DOF must also move at the same speed, i.e. be synchronized. *springs* mode in the appropriate places may also accommodate imperfect synchronization.

5.4.1 Sequential Rotation Turning

Sequential rotation turns require forward motion in order to achieve rotation. The rotations presented here are applied to the serially articulated locomotion modes. The motions are achieved by linking simple gaits with segments whose plane of action is parallel to the ground. We call these segments **lateral segments**, as opposed to those segments moving perpendicular to the ground which we call **sagittal segments**.

In each locomotion mode, the robot turns by sequentially bending and unbending the appropriate segment at the appropriate time as the robot translates. To illustrate, as a robot chain translates over a specific point at which we desire the robot to turn, each lateral segment bends as it moves over it, and unbends when the *next* lateral segment starts to bend as it moves over the point.

Earthworm and caterpillar Turning: The earthworm and caterpillar morphologies are both single lines, and so are equivalent in terms of turning. Turning is

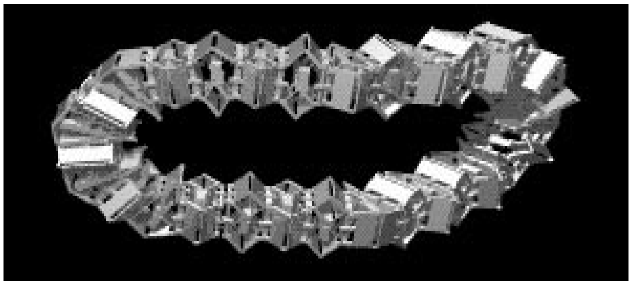


Figure 5.14: Turning Rolling-Track locomotion

achieved as described above with the following modification to handle the sliding.

When a bend is present between two ground contact positions, the motion needed to move in the null space is not a straight line as described in the previous section. So, we let the lateral segments run in a +r*springs* mode. To minimize the force needed to shape the springs, the gaits approximate the shapes needed given the restrictions of the *ends* modes. In the worst case, some slipping will occur, which in some cases is acceptable.

Rolling-Track Turning: Turning with the Rolling-Track is very different from the articulated gaits and can be seen in Figure 5.14. Since the robot is a loop, the vertical projection of any point intersects the robot twice. So to maintain closure, the robot must laterally bend in two places, one point closer to the ground and one point directly above it.

Let us first look at the point near the ground. Since the track is essentially laid down at one end and picked up at the other, no motion occurs between adjacent ground contact positions at any time in any of the motions. When the robot wants to turn at a specific point, a lateral segment bends with a + or −r*ends* mode and is placed over or near that point. It remains there until the robot passes the point entirely. Thus sliding will not occur.

As the robot moves over a point, the top part of the robot moves over the point at twice the speed that the robot is moving. When each lateral segment moves over a turning point, it must bend and unbend at twice the speed that each sagittal segment bends to locomote. Here all other revolute DOF should be in +*springs* mode

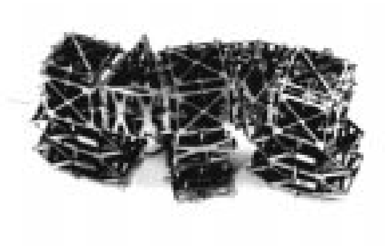


Figure 5.15: Differential translating turning caterpillar

to accommodate imperfect position control and synchronization. The simulation of this mode can be seen in [Yim video 1994].

5.4.2 Differential Translation Turning

The differential translation turns are achieved on Polypod by stopping forward motion and pivoting around a point.

One scheme to achieve differential translation was implemented using the caterpillar locomotion mode, Figure 5.15. A three-footed configuration is used. The center foot is used as a support when the other two feet are raised. Each end of the robot pivots around 45 degrees by bending the lateral segments with + or $-\mathrm{r}ends$. Once the joint limit has been reached, the outside feet are lowered, the center foot raised and the lateral segments straighten. Since the straightening process does not move in the null space of the robot chain between the two outside legs, the two feet slip on the ground by rotating with respect to each other. This gait is shown locomoting on linoleum tile in [Yim video 1994].

The combination of weak motors and the sliding caused this gait to fail on carpet, the connection plates tended to grip the carpet such that the actuators could not move.

Chapter 6

Vehicle and Terrain Evaluation

Since we have presented a taxonomy of locomotion and shown Polypod implementations of these classes of locomotion, presenting the usefulness of these different classes is the next step. We can take advantage of the different locomotion gaits if we know which gait is best suited for different tasks and environments.

6.1 Introduction

Consider the following problem for Polypod: Given a set number of modules, which configuration should Polypod reconfigure into, in order to best traverse an area? For example, Figure 6.1 shows a map of the area surrounding the robotics lab at Stanford and the neighboring Stanford Cardinal Cogeneration Plant. If Polypod were to follow the path marked out on the map, (where its goal is perhaps plant maintenance) it would have to do the following:

1. Cross a 20 yards of a raised wooden deck
2. Climb over or under a railing
3. Step down 2 feet onto hard earth
4. Traverse a moderately bumpy, grassy terrain
5. Step down a curb

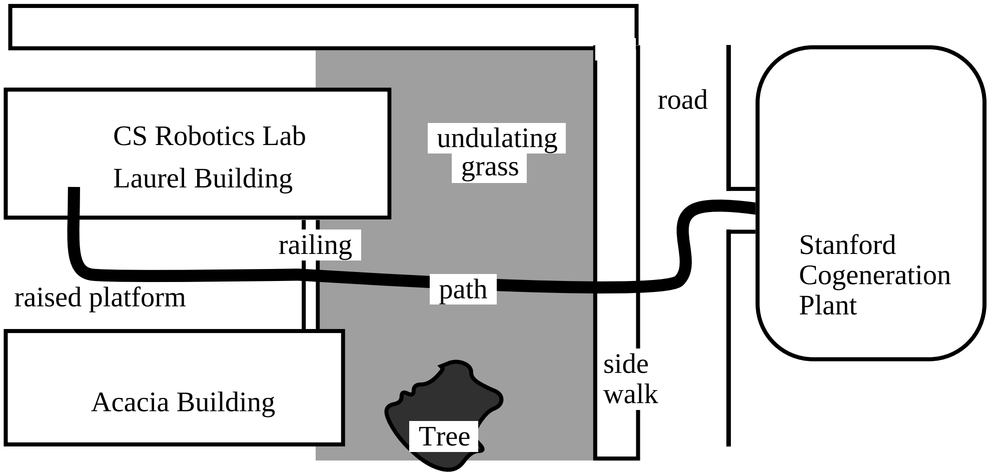


Figure 6.1: Example task (following a path) needing reconfiguration

6. Cross a paved road

7. Enter the Cogeneration Plant

8. Inspect some pipes...

Which locomotion gait would best suit this path? Or, which set of gaits could Polypod reconfigure into that would best suit different portions of this path? The first criterion is that the gait and the vehicle should be able to traverse the terrain.

In the previous chapter we presented a taxonomy of locomotion that was based on the functional method of locomotion not on the vehicle. When analyzing a locomotion gait for a terrain we must also consider the vehicle, the size and range of motions of each DOF will determine how well the vehicle interacts with the terrain. In this chapter we will present a taxonomy of terrain features and list corresponding vehicle parameters that give an indication of a vehicle's ability to cross that terrain feature. These parameters allow us to evaluate a vehicle and a gait for a given terrain.

When evaluating vehicle designs for a mission, there can be other factors besides terrain interaction that must be weighed. We assume the task to be little more than move from place to place possibly with a payload or that the mission does not effect locomotion, e.g. the task is not plowing a field or pushing boxes etc. So, we also consider the payload capacity, energy efficiency, and stability of a vehicle and gait.

6.1.1 Related Work

One of the classes of terrain effects is vehicle soil interaction There has been some work on characterizing mechanical properties of soils [Sinha 1992][Bekkar 1969]. For many applications, soil foot interaction is important, especially mud, soft soil, sand etc. One measure for locomotion on these types of soils is called wheel rolling resistance. This rating also applies to tracks and legs and is a measure of the force that is required to traverse through that soil [Bekkar 1969]. Bekkar is quite thorough in his book, however we will not go into detail except to say that better performance is obtained by having a larger ground contact surface area per unit weight.

Characterizing the geometry of environments with respect to locomotion has not been studied as extensively compared to soils. Geologically similar areas has been studied in [Strahler 1952]. However, this analysis is not suited for locomotion analysis [Bekkar 1969]. Song and Waldron examined the geometric design aspects for a vehicle crossing obstacles as part of the development of the OSU-ASV [Song 1986]. However, they limited their study to walking vehicles and to essentially two types of obstacles, crossing ditches and climbing walls with no generality in the study of terrain effects.

Power Spectral Density (PSD) is a useful method characterizing the roughness of terrain [Bekkar 1969]. It is a graph of the amplitude (height) of terrain waves versus the frequency (length). In other words it gives an impression of the amount and size of bumps in a terrain. If a PSD of a terrain is known, we can estimate the quantity of some terrain features and how they will effect a vehicle. For example if the PSD of a terrain indicates that there are no hills or valleys larger than the diameter of a wheel on a vehicle, most likely that vehicle will have no problems traversing that terrain.

In terms of obstacle-crossing, Bekkar identifies two types of failures, nose-in failure (NIF) and hang-up failure (HUF), Figure 6.2. Both of these failures describe a

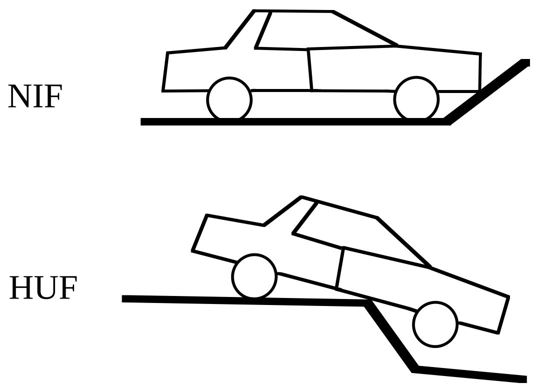


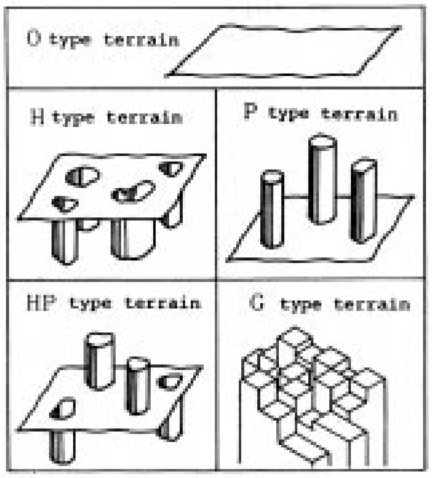
Figure 6.2: NIF and HUF.

situation when a portion of the body not intended for contact, contacts an obstacle. In the case of nose-in, the front part of the vehicle contacts. An example would be hitting a wall. Hang-up occurs when the underside of the vehicle contacts, as would happen when the front end of a truck goes over a cliff.

Hirose proposes a simple five type terrain classification for walking vehicles shown in Figure 6.3 [Hirose 1984]. O-type of terrain is entirely passable by the vehicle, any foot may be placed anywhere. H-type of terrain includes O-type of terrain with holes and ditches in it, so there are areas that the foot may not be placed. P-type includes O-type with poles and rocks in it so there are areas in which the entire vehicle may not enter. HP-type includes O-type with areas of H-type and P-type features. And lastly there is G-type which is essentially a height-field or elevation map. The G-type can be considered a raw terrain type which can be processed into O-type, H-type, P-type and HP-type of areas.

6.2 Terrain Feature Effects

We propose four types of effects a terrain feature can impose on locomotion. Features can act like a foot-obstacle, a body-obstacle, a CG-obstacle, or it can be like free terrain (have no effect).



from [Hirose 1984]

Figure 6.3: Five terrain type

Foot-obstacle: A foot-obstacle is defined as contiguous points on the ground surface which a G-foot may not be placed for use as a support. Foot-obstacles are features in the workspace and are easily identified. They could include such non-geometric things as water, or fragile areas that should not be stepped on such as flower gardens, small animals etc.

In the geometric sense, foot-constraints are classified purely by the slope of the terrain and the coefficient of friction between the robot and the terrain. If the slope is greater than some critical angle where sliding will occur, that point on the ground acts as a foot-obstacle. This terrain effect is similar to what Hirose called an H-type feature.

Body-obstacle: Body obstacles are portions of terrain in which the vehicle must collide in order to obtain a statically stable position at a given location.

In Hirose's classification, P-type terrain had obstacles which could be considered infinitely tall poles since no part of the vehicle could cross over the obstacles. Hirose's classification thus omits the obstacles which may collide with vehicle only in specific configurations.

CG-obstacle: CG-obstacles are areas of terrain which the center of gravity of the vehicle cannot enter due to collision or static stability constraints. For example, the center of a wide hole would be a CG-obstacle if the vehicle could not straddle the hole.

Free terrain: Free terrain are those areas free of foot and body and CG-obstacles.

6.3 Terrain Features and Vehicle Parameters

In this section we present a set of terrain features and the vehicle parameter that determines the feature's effect on the locomotion.

By comparing corresponding parameters of different vehicles, a judgement can be made about which vehicle can handle that type of terrain feature better. Thus if we know how much of each type of feature exists in a terrain, we can make a judgement on which vehicle can best traverse that terrain.

Figure 6.4 presents a graph of terrain features ordered according to assumptions that may be made in analyzing locomotion. Items on the left side of the graph are precursors to those on the right. For example, if the terrain is assumed to be static, then you only need be concerned with the left branch from the root, however if the terrain is dynamic, both sides are needed.

The leaves of the graph are terrain features that may present some effect to locomotion of the vehicle. Each feature may be combined with any or all other features. For example, in a wet hilly forest, there could be width constraints (moving between trees) height constraints (moving under branches) plasticity/viscosity (mud) slope (hills) at the same time.

The vehicle parameters are based on the assumption that the vehicle will use the DOF that are not used to achieve the simple locomotion gait, to cross or avoid obstacles as needed. For Polypod this means the DOF which are in *springs* modes for terrain following may need to be controlled differently to avoid obstacles or cross terrain features.

6.3.1 Static Terrain Features

Slope: The slope of a point on the ground combined with the friction characteristics determine whether a portion of ground acts as a foot-obstacle. The vehicle parameter we use to describe this is critical angle V_s at which sliding will occur.

The measure of V_s is found by the following equation using the Coulomb friction model:

$$V_s = \arctan(\nu_s) \tag{6.1}$$

where ν_s is the coefficient of friction between the robot and the terrain.

Ditches: Any area of terrain which is characterized by two free terrain areas bordering foot-obstacles, with no ground that can be intersected by a line drawn from any point on one free terrain border to any point on the other border is defined as a ditch. Some examples of ditches are shown in Figure 6.5, where the bold lines represent foot-obstacles. Note that no ground intersects the gray line drawn across the top of each ditch. A ditch is crossable if the distance between the two free terrain areas is small enough, otherwise it acts as a CG-obstacle.

The parameter which describes this, V_d, is the vehicle's reach, or the maximum width pit that the vehicle can cross, In all cases V_d can at best be half the longest length.

Hang-up: This type of feature can stop the motion of a vehicle by colliding with the underside of the vehicle. For example, when a car moves over a speed bump that is too thin and high, the bump will hit the underside of the car lifting one set of wheels off the ground. Hang-ups occur only for vehicles which have portions of the body which are not G-feet, exposed on the underside of the vehicle. Hang-ups are characterized as small protrusions that may fit between G-feet and contact the body. Some examples of hang-ups are shown in Figure 6.6. A vehicle can cross a hang-up feature if the protrusion is small enough, or if the vehicle has no exposed portions underneath the vehicle, otherwise the hang-up acts like a body-obstacle.

We characterize the allowable hang-up, V_{hu} by finding the largest sphere that will fit underneath the vehicle such that the sphere touches the body portion and cannot

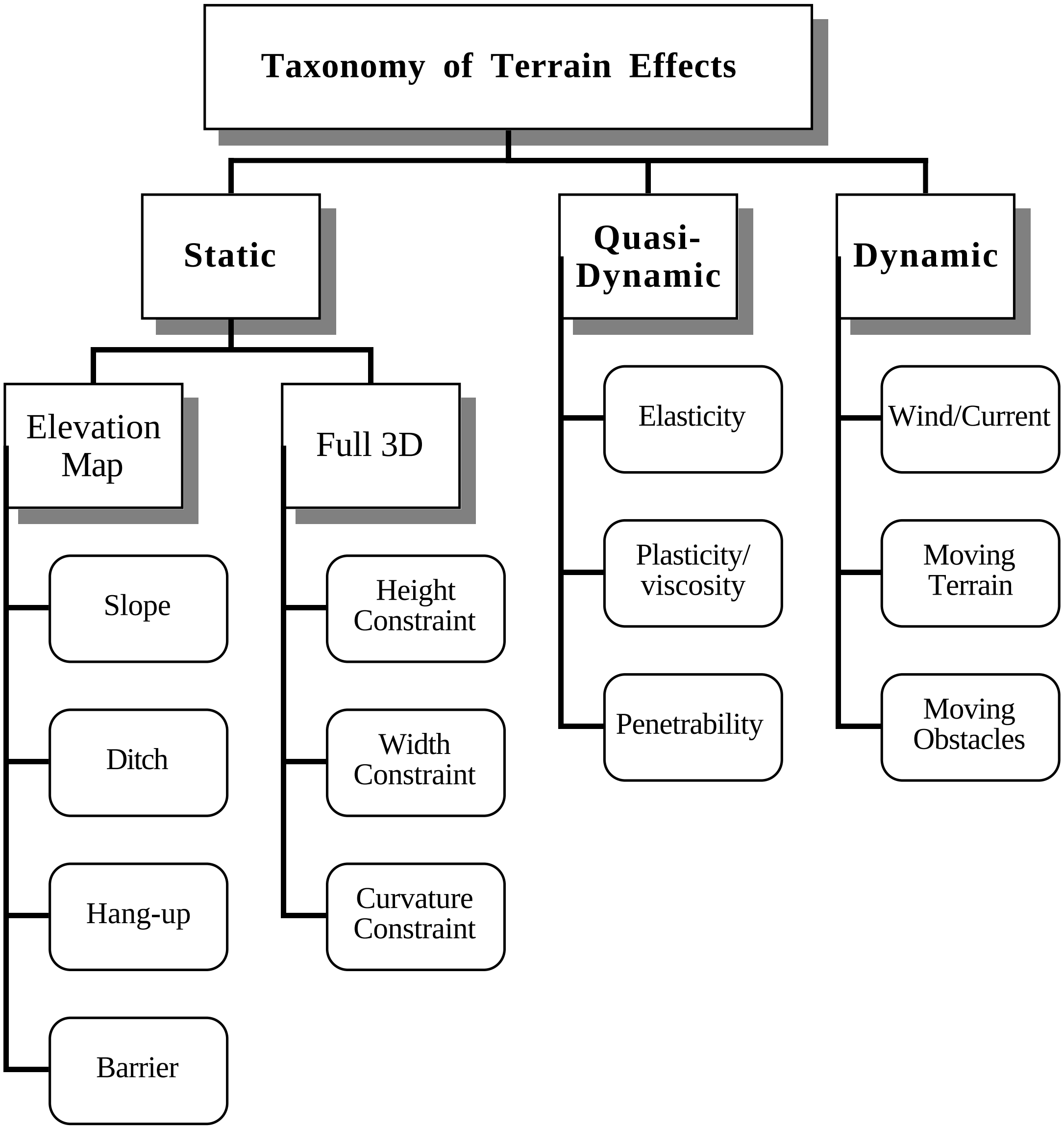


Figure 6.4: Taxonomy of terrain features

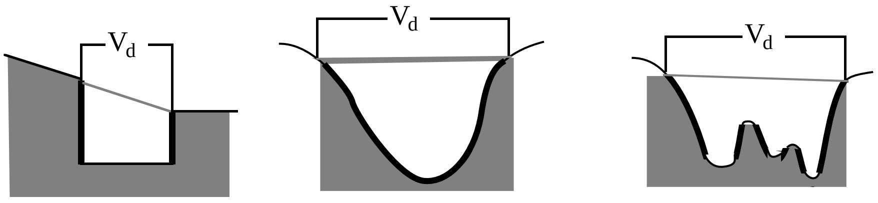


Figure 6.5: Example ditches

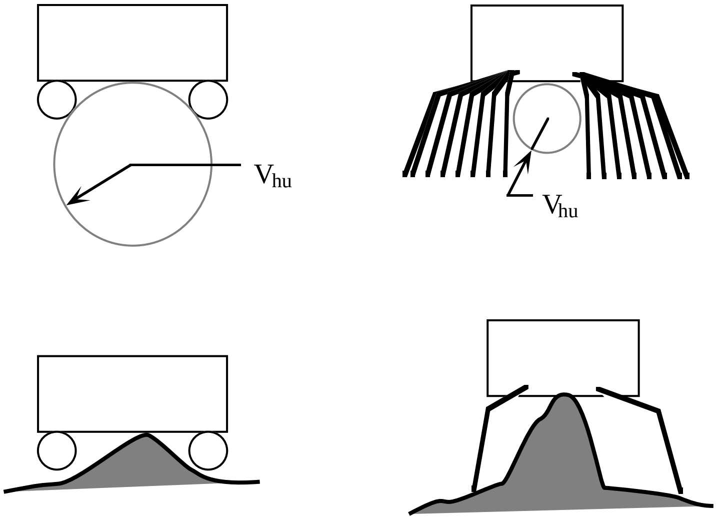
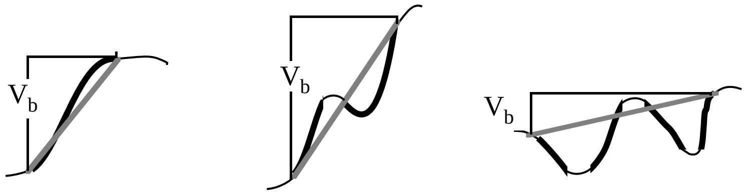


Figure 6.6: Example hang-ups, wheeled vehicle, legged vehicle.



The bold lines represent foot obstacles. The thick grey line joins the two free terrain areas.

Figure 6.7: Example of walls, step-up, single peak, multiple peak

be touched by a G-foot.

Barrier: The barrier terrain feature is the companion to the ditch and is defined as follows: Any area of terrain which is characterized by two free terrain areas bordering foot-obstacles that *has* some portion of the ground intersecting a line that is drawn from one free terrain border to the other. Some examples are shown in Figure 6.7, note that the gray line drawn across the barrier intersects with the ground at some point..

Walls are characterized chiefly by their height. For situations in which the vehicle can step over the wall, the feature can be treated as a crossable hang-up. So the vehicle parameter is the maximum height of a step that is achievable, V_b.

Height Constraint: Height constraints are obstacles such as ceilings, the top of doorways, branches etc. anything that the robot must travel under to get to a desired external configuration. These act like body obstacles when their height is below the height of the vehicle.

The vehicle parameter, V_h, is the minimum height under which the vehicle can pass.

Width Constraint: Width constraints describe the situation when pole type obstacles exist such as trees, or walls in a building. The width constraint acts as a body obstacle for vehicles which are a wider than the obstacles will allow to pass.

The vehicle parameter, V_w, is the minimum width which the vehicle can pass through.

Curvature Constraint: This constraint involves the radius of curvature that is allowable for a robot moving between obstacles, for example, in a turn in a tunnel.

The vehicle parameter, V_c, is defined to be the smallest radius of a semicircle in which a vehicle can turn 180 degrees. This can be visualized by the outer radius of a volume swept out by a vehicle turning its tightest turn.

6.3.2 Quasi-Dynamic Terrain Features

Elasticity Plasticity/Viscosity of Contact Points: The elasticity and plasticity of the ground can effect the dynamic behavior of locomotion. We will assume that all ground contacts are infinitely stiff. Realistically, the structural properties of the ground greatly effect what types of features can be important. For example, blades of grass or small plants can be ignored by treaded tanks and trucks, however, if they were made of steel it would be a different matter.

Penetrability of Contact Points: Penetrability refers to the sinkage of a foot into the soil. In more detailed studies other factors such as heterogeneous soil levels can effect the sinkage of the soil, like ice layers on water. Our main concern will be how much the foot penetrates the soil which we will approximate as linear to the stress seen at a G-foot.

The vehicle parameter, V_p, is the maximum stress seen at a foot, $\sigma = \frac{W}{fn}$, where W is the weight of the robot, f is the area of a footprint of one foot, and n is the minimum number of feet in contact with the ground.

6.3.3 Dynamic Terrain Features

Wind/Current: Wind or current in water effects the motion of a vehicle in addition to stability. We will only be concerned with stability.

The vehicle parameter, V_m, is the minimum Energy Stability Margin (a measurement of stability based on the energy required to tip the vehicle over, as described in the next section).

Moving Terrain (Earthquakes, Floating Terrain Etc): These terrain features effect the locomotion similarly to wind or current. Again we will only be concerned with stability, so we will use the same parameter V_m

Moving Obstacles: Moving obstacles must be avoided to prevent damage to the vehicle and also prevent tip-over. The vehicle parameter, V_o, is the maximum speed of the robot.

Others: While this taxonomy is an attempt to be complete, it is possible that there are other terrain effects not included. The following is a list of terrain effects that affect components of a robot moving in specific terrain and not the locomotion in general.

- Radiation may effect sensors and computers in nuclear power plants,

- Heat transfer rates may effect motors for fire rescue or outer space operations,

- Pressure may effect sensors and motors for underwater applications,

- Corrosive liquids and gases may effect structures and joints in industrial plants.

6.3.4 Polypod Vehicle Parameters

Some vehicle parameters for each of the Polypod gaits are listed in Table 6.1. The table lists all of the static features with the exception of the slope parameter V_s which

Polypod Static Feature Vehicle Parameters

Gait	V_d	V_{hu}	V_b	V_h	V_w	V_c
Earthworm *	$1.1(s+n)$”	0.0”	$1.1(s+n)$”	2.3”	2.3”	3.8”
Rolling-Track *	$0.55(s+n)$”	0.0”	$0.55(s+n)$”	6.4”	2.3”	3.8”
Spider	$0.10(s+n)$”	0.0”	$0.10(s+n)$”	$(0.14s+2.3)$”	8.3”	8.3”
Turning-cater	$0.26(s+n)$”	0.0”	1.1”	4.3”	2.3”	8.3”
Caterpillar	$0.26(s+n)$”	0.0”	$0.26(s+n)$”	4.3”	2.3”	∞
Rolling/Carrying	$0.13(s+n)$”	0.0”	$0.13(s+n)$”	6.3”	2.3”	∞
Moonwalk	$0.13(s+n)$”	0.0”	$0.13(s+n)$”	6.3”	2.3”	∞
Cater-cater	$0.55\sqrt{s+n}$”	2.3”	1.1”	4.3”	$0.92\sqrt{s+n}$”	$1.3\sqrt{s+n}$”
Slinky-slinky	5.2”	3.2”	$0.22(s-1)$”	$(0.35s+2n)$”	2.3”	∞
Slinky	5.2”	3.2”	$0.22(s-1)$”	$(0.35s+2n)$”	2.3”	∞
Cartwheel	4.0”	2.0”	$0.13(s-1)$”	$(0.18s+2)$”	2.3”	∞
3-seg slinky	4.0”	0.0”	0.1”	4.6”	2.3”	∞

* Straight versions yield the same numbers as the turning gaits except for $V_c = \infty$.
s is the number of segments, n is the number of nodes.

Table 6.1: Static terrain vehicle parameters for implemented Polypod gaits

depends on the terrain. Quasi-dynamic and dynamic features are left out since we are examining only statically stable gaits. Each of the equations in the chart represents the largest allowable feature size for V_c, V_{hu}, and V_b and the smallest allowable feature size for V_h, V_w, and V_c in inches. A detailed example of the determination of these equations is shown in Appendix D.

A significant feature to note is that in every case, the Earthworm gait has the best numbers for traversing over static terrain features.

6.4 Task Parameters for Polypod

The ability to cross obstacles is one of many task parameters that can be used in evaluating a vehicle for a given task. In this section we will present some other common ones: efficiency, payload and stability. We will also see how the different gaits implemented on Polypod compare with respect to these parameters.

Efficiency: The efficiency of a gait refers to the amount of energy needed to go some unit distance. Since the Polypod gaits are statically stable, and we assume

Polypod Efficiency

Gait	d	$\Updownarrow$	$\leftarrow$	$N(s)$	$N(n)$	$n=16, s=56$
3-seg slinky	4.0”	3	0	3	0	0.31
Turning Track	2.5”	4	0	$s/2+4$	$n/2+$ 4	0.0073
Rolling-Track	2.5”	4	0	$s/2+4$	$n/2+$ 4	0.0073
Cartwheel	4.0”	4	4	s	n	0.0064
Slinky-slinky	5.2”	6	4	s	n	0.0062
Slinky	4.0”	6	4	s	n	0.0048
Spider *	2”	12	0	$2s/5$	$n/2$	0.00036 *
Rolling-while-carrying	6”	4+4s/8	0	$s/4+10$	$n/2+4$	0.0031
Cater-cater	2.5”	6s/5	0	7	2	0.0028
Turning-cater	2.5”	2s	0	5	2	0.0020
Caterpillar	2.5”	2s	0	5	2	0.0020
Earthworm	2.5”	8s	0	4	$4n/s$	0.00075
Moonwalk	(-)2”	4+3s/4	0	$s/4+10$	$n/2+4$	0.00041

* The spider efficiency rating uses estimated equivalent parameters

Table 6.2: Efficiencies for implemented Polypod gaits

quasi-static motion, the energy lost is not a function of velocity or acceleration. Also due to the self-locking nature of the actuators and large amount of friction in the transmission, the dominant factor in energy usage is the total distance in joint space that the segments move.

Thus the efficiency is the ratio between the amount of motion in the joint space versus the amount of forward movement. As more modules are added to the system the weight of the robot becomes more of a factor and will effect the amount of energy lost from friction due to the increased surface normal forces. With this in mind the equation for the efficiency ϵ is presented below:

$$\epsilon = \frac{d}{d + (\Updownarrow + \leftarrow /2)(N(s) + 3 * N(n))} \tag{6.2}$$

where d is the distance travelled in one cycle of the gait, $\Updownarrow$ is the sum of the number of times any DOF moves through its full range in *ends* mode during one gait,$\leftarrow$ is the same for $-$*springs* mode (the energy used in +springs mode is assumed to be negligible), s is the number of segments in the robot, n is the number of nodes, $N(s)$ and N_n are the maximum number of segments (nodes resp.) that occur between two

Polypod Stability

Gait	R_X	R_Y	S	$n = 16, s = 56$
Spider *	4.4”	4.1”	1.91	1.91
Cater-cater **	$6.2(\sqrt{n} - 3)$”	3.4”	$\approx 2\sqrt(n)$	1.78
Earthworm	1.1”	2.0”	0.28	0.28
3-seg slinky	1.1”	2.6”	0.22	na
Turning-Track	1.1”	3.2”	0.18	0.18
Rolling-Track	1.1”	3.2”	0.18	0.18
Turning-cater	1.1”	3.4”	0.17	0.17
Caterpillar	1.1”	3.4”	0.17	0.17
Rolling-while-carrying	1.1”	6.2”	0.097	0.097
Moonwalk	1.1”	6.2”	0.097	0.097
Cartwheel	1.1”	$0.26s$”	0.14	0.04
Slinky-slinky *	1.1”	$3.3n + .4 * s$	$<< 0.01$	0.008
Slinky *	1.1”	$3.3n + .6 * s$	$<< 0.01$	0.007

* assumed nodes distributed evenly among segments

** assumed n >> 16

Table 6.3: Stability for implemented Polypod gaits

ground contact points.

One trend to note here is that the more modules used the less efficient the locomotion. The only way to have perfect efficiency is to have no moving modules. A more meaningful test is to examine gaits which have the same total number of modules.

Using Equation 6.2 the efficiencies are listed for each of the modes assuming that we have 56 segments and 14 nodes in the last column of Table 6.2. A more detailed example of how this efficiency is determined is included in Appendix D. The Rolling-Track is the most energy efficient mode of locomotion for this many modules.

Stability: Messuri et. al. [Messuri 1985] presented a good measure for the stability of a given configuration called Energy Stability Margin (ESM). The ESM of a configuration is the minimum amount of energy that must be input into the system in order to tip the vehicle over. This is a function of the weight of the vehicle and the change in height of the center of mass from the stable configuration to the point of just tipping over.

Polypod Payload

Gait	min. ground legs	load surface	payload	$n = 16, s = 56$
Cater-cater	n	$16n$ in^2	$3n$ lbs	48 lbs
Turning-cater	n	$8n$ in^2	$3n$ lbs	48 lbs
Caterpillar	n	$8n$ in^2	$3n$ lbs	48 lbs
Rolling-While-Carrying	$\frac{1}{2}(n-8)$	$2(n-8)$in^2	$\frac{3}{2}(n-8)$lbs	12 lbs
Spider	4	4 in^2	12 lbs	12 lbs

Table 6.4: Payload for implemented Polypod gaits

We will define the stability of a gait to be the minimum ESM of all configurations that the vehicle goes through as it traverses in a straight line over flat terrain. The two parameters that define the stability for a given set of modules and a gait is the minimum distance from the CG to any point on the support polygon, R_X, and the height of the center of mass, R_Y.

$$S = \sqrt{{R_X}^2 + {R_Y}^2} - R_Y \tag{6.3}$$

where S is the ESM. As the robot moves, the support polygon changes. The least stable configuration is used to determine the ESM.

Table 6.3 shows the relative stabilities of all the Polypod gaits presented in Chapter 5. More detailed examples of the determination of ESM for different gaits are included in Appendix D. The spider gait is the most stable.

Payload: Payload is an important parameter for the task of delivering objects or carrying tools. Hirose presents an argument for carrying large loads in a nuclear power plant setting [Hirose IJRR1990]. One of the loads that is carried is a manipulator arm. In the case of reconfigurable modular robots, a separate manipulator is not needed since the robot itself can reconfigure into a manipulator arm. The manipulator arm and the controller for this arm can often be significant portions of a payload.

For Polypod not all gaits are suited to carrying loads. The ones that are easily suited are the Cater-Cater gait, Caterpillar gait, Spider and the Rolling-While-Carrying gait. The weight of load that each can carry is a function of the way the load can be distributed. In Table 6.4 the amount of weight that each mode can carry given that each module (moving through its full range of motion) can carry 3 lbs. The Cater-Cater gait is the best suited for carrying large payloads given its weight capacity and its large load carrying surface area.

6.4.1 Polypod Summary

Here we present a solution to the problem introduced at the beginning of the chapter. This example will illustrate the use of the best gaits for each of the tasks and also presents an useful example of reconfiguration. The following list shows the steps in following the path marked out on the map in Figure 6.1 and the Polypod gaits that will be used for each section.

1. Leave an indoor environment: would use Turning Loop gait
2. Cross a wooden deck
3. Climb over or under a railing: reconfigure into Earthworm gait
4. Step down onto hard earth
5. Traverse a moderately hilly grassy terrain: reconfigure into Spider gait...

The first two steps the locomotion is on flat terrain. The Turning Loop is the most energy efficient locomotion mode (besides the three segment slinky) so it is the gait of choice and is shown in Figure 6.8.

The railing has a height of 40" (barrier) and lower gap of 5" (height constraint). The Loop cannot overcome either of these terrain features. The Earthworm gait is the only one that can overcome both of these features. It is clearly the optimal gait for overcoming individual obstacles. Once past the railing, the Earthworm gait can climb down the large step down onto the earth. (See Figure 6.9.)

The next terrain is bumpy, and could cause a problem for the Earthworm gait in terms of stability. The terrain feature parameters assume that the vehicle knows the terrain exactly. In reality the terrain model will have errors as well as sensing the

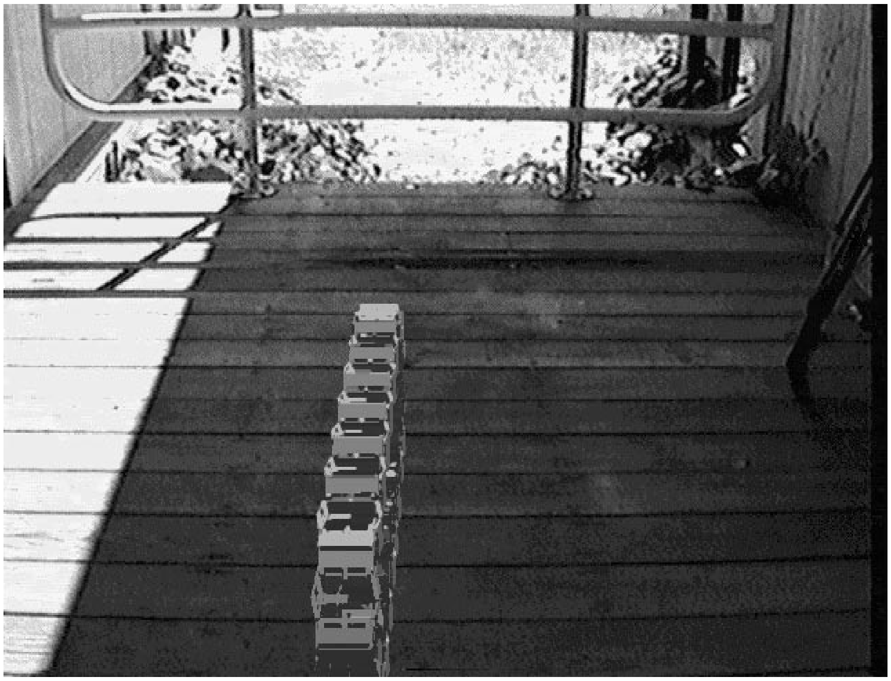


Figure 6.8: Polypod using the turning loop gait on a wooden deck

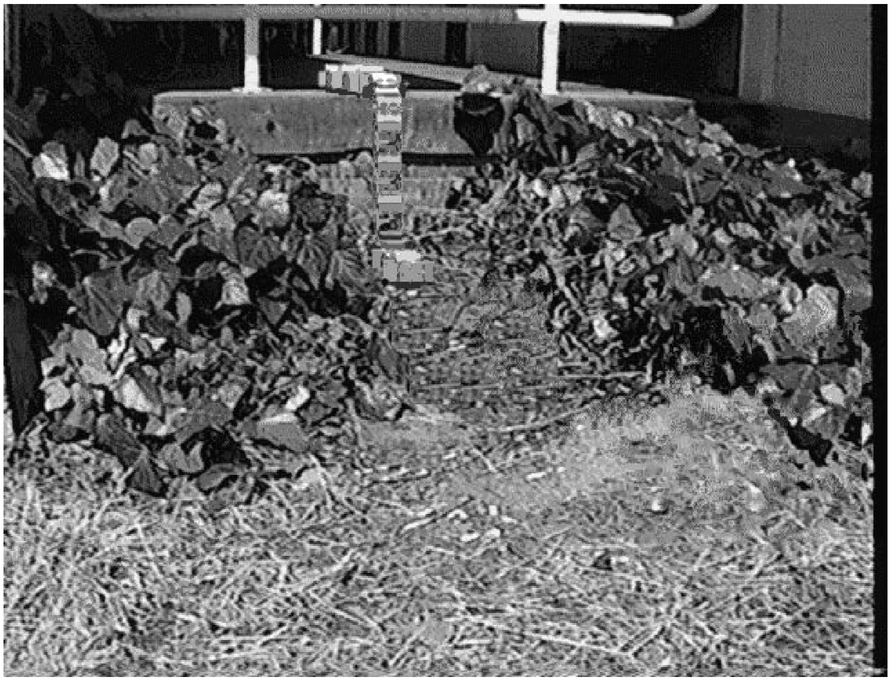


Figure 6.9: Polypod using the earthworm gait to overcome obstacles

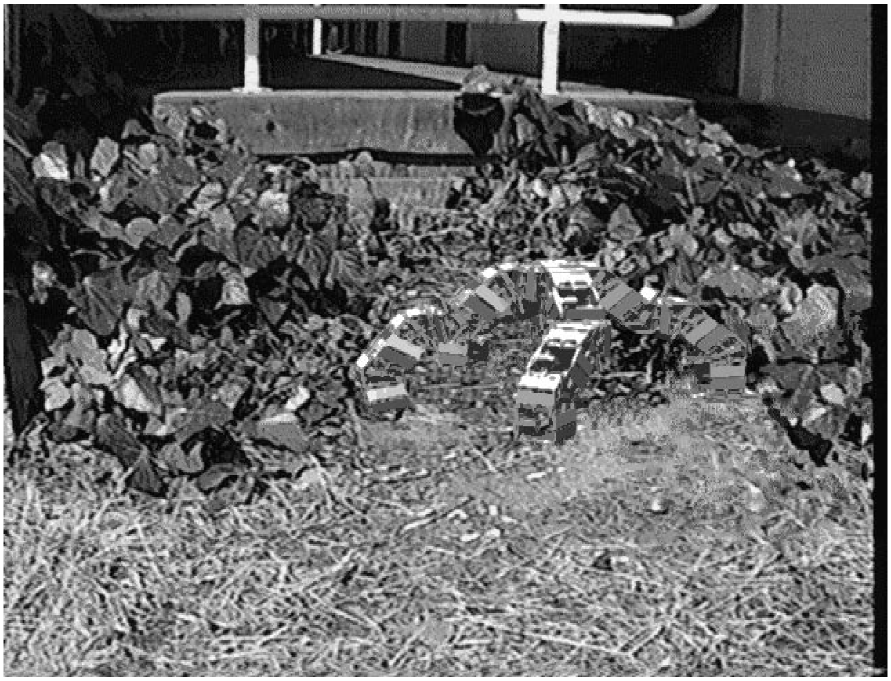


Figure 6.10: Polypod using the spider gait on rough earth

terrain. These errors could cause Polypod to tip over. In most cases, Polypod tipping over is not a catastrophic event since it can easily reconfigure itself, however tipping over multiple times would greatly impede the progress. The Spider gait is the most stable gait, so this is the one chosen to traverse this section (Figure 6.10).

Optimal Polypod Gaits: The turning loop is the most efficient gait for flat terrain. The earth worm gait is the best gait for crossing obstacles. The spider gait is the most stable. The caterpillar and cater-cater are the gaits with the largest payload.

In terms of locomotion classification for Polypod we can say the following:

- A Rolling Continuous gait is the most efficient for statically stable locomotion.
- A Swing Continuous (articulated) gait is the best at crossing obstacles.
- A Swinging Discrete Little-footed Gait is the most stable.
- A Swinging Discrete Big-footed Gait can carry the most load.

The one class of gait not represented above, the Rolling Discrete class of gaits. It is interesting to note here that if we were to analyze these gaits for their dynamic performance, the cartwheeling gait, (the archetypical Rolling Discrete gait) would be the easiest to implement since it most closely approximates a wheel in shape and function. The Rolling-Track is too oblong. Thus each class of locomotion for Polypod has a situation for which it is best suited.

Chapter 7

Prospects for Scaling

This chapter presents the general issues and some analysis techniques for the problem of scaling up the number of modules, or scaling down the size of the modules.

7.1 Introduction

What can be done with a modular reconfigurable robot? The question that this chapter explores is "What cannot be done with such a robot?" That is, what are the limitations on the number of modules that can be added together. To answer this, we will use scaling analysis which has been used in animals and animal locomotion. Scaling methods have also been used extensively in model analysis. We will also make use of continuous approximations of modular structures.

Schmidt Nielsen proposes scaling to be defined as the structural and functional consequences of a change in size or in scale among similar shaped animals (or mechanisms in the more general case) [Schmidt-Nielsen 1977]; for a modular system, scaling the size of the robot means adding modules to the system while maintaining the same approximate shape.

For similarly shaped animals equations can be constructed to determine size effects in a constant fashion. For example, body mass to skeletal mass $M_{sk} = 0.1M_b^{1.13}$ is a well known relationship in animal physiology [Schmidt-Nielsen 1977]. The same type of equations can be applied to scaling mechanisms. However, discontinuities exist when changes in design, material or environmental conditions occur. For example there is a limit to the length of a stone arch bridge due to compressive strength of stone, but by changing the design or material, the length of the bridge could be increased several magnitudes.

Another example of this type of dimensional analysis applies as we shrink mechanisms. Friction effects are often modelled as dependent on the surface area of contact (linear dimension squared), while inertial effects are dependent on volume (linear dimension cubed). Thus, as mechanisms are scaled down, friction becomes more and more dominant.

We will analyze the scaling effects by dividing the robotic system in to three components: the physical structure, the actuation, and the control (which includes sensing and computation).

7.2 Structure

The static analysis of the bone structure of animals shows how much mass the bones can hold while standing still. In terms of size, bone strength is a function of the cross sectional area, a square of the linear dimension, while the mass is a function of the volume, a cube of the linear dimension. Thus, as the size increases it is not enough for the bone structure shape to increase proportionately.

For analyzing a scaled structure we can use a substitute continuum approach. This approach has been used successfully on large lattice structures (specifically for space applications). Two properties that make this method especially appropriate for unit-modular robotics are that it simplifies the system by reducing the number of DOF and second, an increase in the number of modules results in a better approximation of the model.

Noor has shown a method for analyzing elastic stress analysis, buckling and free vibration of both beam like lattices structures and plate-like lattice structures [Noor 1988]. This is done by first modeling the displacement and material properties of a single repeating lattice. Then the system is reduced to the various continuum material and displacement parameters that approximate the lattice.

We will use a continuum approach to analyze the buckling of a single chain of modules and then discuss the scaling issues in the inverse kinematics of structures.

7.2.1 Buckling under self weight with Polypod

The simplest question is how many modules can we attach end to end? An upper bound on this can be found by seeing how many modules stacked on top of each other in a column would cause the column to buckle due to its own weight. We assume that this column doesn't need to move as the next section on actuators will present scale limits due to actuation limitations.

We use the continuum approach [Noor 1988] to find the material and displacement parameters that approximate the stacked modules. We then use standard Timoshenko elastic column buckling analysis as if the modules were one continuous beam.

The equation for buckling under its own weight one end fixed, the other free is given below:

$$w_c r = 7.83 \frac{\mathrm{E}I}{L^2} \tag{7.1}$$

where $w_c r$ is the critical weight of the entire column, E is the modulus of elasticity and I is the moment of inertia [Timoshenko 1961].

The modulus of elasticity about an actuated axis is not defined, however the actuator may simulate a material property. The measure of E depends on the strength of actuators and the control law used, in effect the robot has a variable E. This concept is used widely in the area of research called smart structures. Although this term usually implies sensing built into structures and limited range of actuation, the same principles apply.

For non-backdriveable actuators, as in the case of Polypod, the joint can be locked and E can be measured in a more traditional way. Force-torque sensors measuring the torque about the bending axis of the column often are much more compliant than the rest of the robot since they need to measure flexure. Thus E for the force sensor can be used for analyzing the approximate stress characteristics of the whole robot [Vischer 1990].

Polypod has a force/torque sensor that has limit stops that engage at compressive forces greater than 7.4 lbs. Thus at this point continuous approximation of the modulus of elasticity changes to that of the rest of the module, 2.1 MPA. The corresponding approximation for the moment about the bending axis $I_{yy} = 5.6 \times 10^{-9} m^4$.

For Polypod this means that 112 segments may be attached end to end before the segments may buckle under its own weight when vertical. The only conceivable time that the robot should need to be in this configuration would be if the robot needed to reach as high as possible. In most cases the robot could form a lattice structure with many modular chains forming the lattice links.

7.2.2 Euler Buckling

In the case of legged locomotion, the robot may form long legs on top of which it may hold some load including the weight of the body of the robot. Also, in lattice structures each strut within the lattice will see some load that may cause buckling. Again we can use the continuum approach. For these situations, Euler buckling can be used as a rough approximation for the limit in the number of modules for a given force F.

The Euler buckling occurs when a force is applied to a beam axially with a force of the following magnitude,

$$F = 0.25\pi^2 \frac{EI}{L^2} \tag{7.2}$$

Using the same values from the previous section results in $\frac{F}{m^2} = 46$kg. where F is the load and m is the number of modules.

7.3 Inverse Kinematics

One problem with scaling the structure is in finding a scalable inverse kinematics method. For serial chains the inverse kinematics problem is defined to be: Given a base frame and goal frame, find the joint angles for the robot that will cause the end-effector to reach the goal frame if the base of the robot is at the base frame. To be scalable the method must be able to find all the joint angles in a fixed amount of time, independent of the number of joints.

As more modules are added to the system, the number of DOF increases. The standard manipulator inverse kinematics methods for non-redundant and moderately redundant manipulators cannot be used as the computational time complexity is typically exponential in the number of DOF. The ideal inverse kinematics method would be independent of the number of DOF, or at worst linear in the number of DOF. As stated in Chapter 2, Chirikjian and Burdick propose an inverse kinematics method that can be made linear in the number of DOF at best for a standard single processor machine. They also propose a constant time method which can be used with multi-processor modular robots such as Polypod [Chirikjian 1992]. Kobayashi also presents a local method for solving the inverse kinematics although their method is subject to local minima [Kobayashi 1992]. Banon investigated inverse kinematics for a robot that had infinite DOF [Banon 1994], however he did not extend his method to any robot that could be physically realizable.

We present two inverse kinematics solutions for Polypod for both the 3-DOF planar case and full 6-DOF 3D case that is independent of the number of modules and DOF of the robot in Appendix A. These methods are based on fitting the robot to sequences of arcs of circles.

A new problem which has not yet been addressed is the inverse kinematics of structures such as surfaces or solids. The problem can be posed as follows: Given a description of the shape of a surface or solid, what are the joint angles of the robot that can approximate that shape.

One solution to the problem is to extend the work of Chirikjian and Burdick. Instead of using an inverse kinematics method based on the curvature of a curve we could use B-splines as the back bone curve and use arrays of modules with these curves joining them. Given the points of a surface or solid there are many ways in computer graphics to approximate the surface or solid with a B-spline surface or solid.

To map Polypod to the surface, nodes could be placed at intervals in two dimensions on the surface (corresponding to constant intervals in the B-spline parameter space depending on how many nodes are available and the size of the surface) and

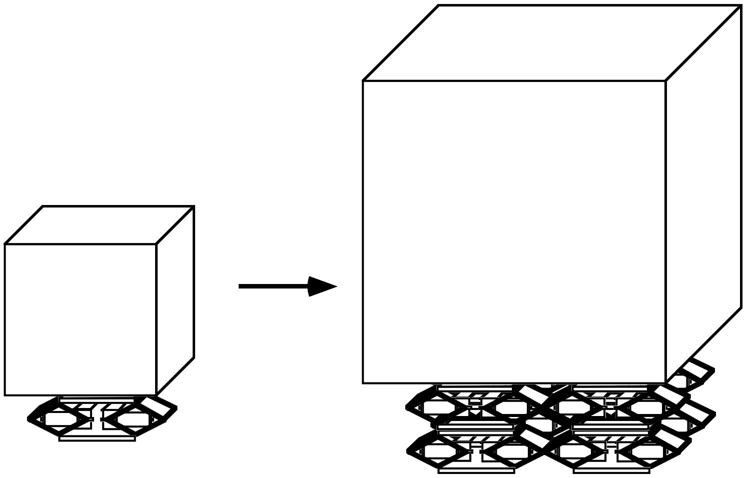


Figure 7.1: Scaling one module supporting a cube by a factor of 2 in each dimension

serial chains could attach the nodes in a grid fashion. Each serial chain could use one of the inverse kinematics methods presented in Appendix A to do this. This same method can be applied to B-spline solids.

The advantage of using a B-spline representation rather than explicitly controlling all the degrees of freedom is that the control needs only to modify the finite number of control points which are independent of the number modules or DOF.

7.4 Actuators

How does the actuator strength scale? In the ideal case, each module under position control would be able to exert an infinite amount of force. In this case adding modules end to end would multiply the speed of the end module by the number of modules. Of course this is not realistic. A set of modules attached end to end would be able to exert a maximum force equal to the force that *one* module could exert. Forces are not additive in series. However, they are in parallel.

Animal	Mass m	Char. length $m^{1/3}$	Acceleration distance	Jumping height
Flea (Pulex)	0.49 mg	0.079	0.075 cm	20 cm
Click beetle (Athous)	40 mg	0.34	0.077 cm	30 cm
Locust (Schistocerca)	3 g	1.4	4 cm	59 cm
Man (Homo)	70 kg	41	40 cm	60 cm

Table 7.1: Jumping heights for animals are on the same order [Schmidt-Nielsen 1983].

As a more graphic example of scaling, take one module lifting a cube. If the approximated shape of the module and the cube is scaled up by a factor of two in each dimension, we have a cube and eight modules as shown in Figure 7.1. The cube weighs eight times the original weight and the modules are arranged in four columns of two segments. Note that each column supports one fourth the weight. Thus, the force each module must support is twice that of the original. The strength of the whole group is increased on the square of the linear dimension while the mass is increased by the cube.

We can again look at animals to see how they compare. Muscles are, for the most part the same from one animal to next. A set of muscle cells under a microscope from a mouse would be indistinguishable from that of an elephant. The strength of a muscle, that is the maximal static force that can be exerted, is proportional to the cross sectional area roughly 4 to 5 Kg force/cm^2. The length of the muscle does not effect the strength. This is exactly analogous to the modular mechanism case.

Another interesting fact is observed when looking at the standing high jump of various organisms. Table 7.1 shows for that for organisms ranging in size from a flea to a man, roughly four orders of magnitude range, each can jump to roughly the same order of magnitude height. Part of this is due to the similar strength of muscles and part due to the canceling effects of the length of take off versus mass.

The implication of this is that for many instances the dynamic performance scales in the same manner as static strength. For static strength, as in Figure 7.1, if we do not scale the height of the cube, but use the height as a measure of strength, then as the rest of the system scales, the height remains constant. The height of the cube is analogous to the height of jumping.

One effect of scaling that is not encompassed in the analysis above is that as the number of modules increase, the robot's own weight becomes more significant compared to the modules that must support it, i.e. the modules are not massless. This limits the number of modules that may be added in a supportable fashion.

Each module in Polypod is able to lift roughly 10 lbs (160 ounces), and each module weighs 3 ounces. So, the limit of modules that will allow the bottom module to move in a vertical serial chain is 53 modules. If more modules are needed for vertical height, modules may be stacked in parallel or in a lattice configuration. This way the system can make use of the varying mechanical advantage and self-locking properties of the segments as described in Section 2.3.3.

The continuum method of structural analysis can be applied to actuator strength as well.

7.4.1 Power Consumption

Another highly studied characteristic in animal scaling is metabolic rate. It has often been theorized that the metabolic rate of an animal is proportional to the surface area, $M_r = kM_b^{2/3}$ where M_r is the metabolic rate, M_b is the mass of the body and k is some constant. Note that surface area is proportional to $M_b^{2/3}$. This is thought to be the case because the amount of heat lost is proportional to the surface area, and the metabolism is responsible for temperature regulation [Sarrus 1839]. In actuality it is not exactly true, $M_r = kM_b^{0.75}$ [Wilke 1977] however, the general trend is there. That is, the smaller the animal the higher the metabolic rate per unit mass.

How does this apply to mechanisms? Metabolic rate is analogous to the power consumption,P, in a mechanism. The power consumption for actuators generally scales with the mass of the mechanism ($P = kM_b$). However, for intelligent mobile robots that must supply their own power, a large portion of the power consumption (often the majority) is due to the on-board computers. As a robot is scaled down the computation is not, thus the amount of power consumed for the computers remains constant. The net effect is that the smaller the robot, the higher the power consumption per unit mass.

For unit-modular robots where a computer is on every module, CPU power is scaled with the number of modules. However there still must be some overhead to coordinate the processors, in the case of Polypod with one master computer. Also, very often as the number of modules increase, many of the modules spend much of the time doing nothing at which time if the modules were ideal, the CPU's would shut down or go into a low-power mode. The result has the same effect on scale and is a benefit for increasing the number of modules and does not present a limitation.

One result of this trend as related to locomotion is in moving vertically. Traveling up an incline is much harder than on flat terrain and can be seen as an increase in metabolic rate. This cost is roughly constant for most animals for each unit of body weight per distance climbed. However, since the metabolic rate varies in each animal, this constant change has a different effect on different sized animals. Mice which have a very high metabolic rate per unit mass would see a proportionately small increase as compared to a horse. This holds true when analyzing the increase in O_2 (a direct measure of metabolic rate), the mouse has a +23% O_2 usage while the horse has a +630% O_2 usage [Schmidt-Nielsen 1977]. This trend would hold for robots with the analogous power consumption.

7.5 Communications and Control

In the limit we will have a great many modules and thus a great many DOF. Control of large numbers of DOF is a complex issue. There are several approaches that one can take. A brute-force approach where each module is programmed individually for each given situation is clearly impractical. The ideal goal would be to have a control method that would be scalable, that is applicable to an arbitrary number of DOF. This section will explore the issues involved.

The analysis of continuum mechanics as it applies to lattice structures presented earlier is one area that is also applicable to control. It provides an effective tool for parameter and system identification [Noor 1988].

7.5.1 Communications

Current research in control that applies to this situation is control of large-scale systems and decentralized control. The application in which most researchers in this area are interested, is for large flexible space structures. There are also several who investigate control of large systems such as power station control, and traffic control.One of the main research topics in this area is the amount of information needed at each controller, which leads to the amount of inter-processor communication.

On existing highly redundant robots such as Polypod, it is not practical to have control lines (sensor and motor control) running from every DOF to a single controller as the jumble of wires would be enormous. On Polypod a serial communications bus is used with each module containing a unique address to which it responds. The limit in number of modules is then shifted to the bandwidth of serial bus. In addition to the unique address, there is an address on which all modules listen to allow for broadcast communications. Since broadcast communications is only limited by the power of the transmitter (usually not a limiting factor), it can be used to greatly reduce the communications time. Otherwise, just communicating commands to each individual DOF would take time linear in the number of DOF.

One example is the inverse kinematics problem. Using the methods in Appendix A, three or six groups of segments would have the same joint angles, and so broadcast communications to each group would allow constant time joint angle communications to each group (as opposed to communicating different joint angles to every joint).

Control strategies can be developed that minimize inter-processor communication [Fukuda 1990(IROS)], however initializing processes can still become a difficult task when large numbers of modules exist. There is also usually a fundamental limit in address space for globally connected systems, as well as bus loading (power) limitations. As electronic technology continues to quickly advance, this limitation may not be a concern as compared to the other limitations.

7.5.2 Computational Complexity issues

One example of the limitations that computational requirements present is inverse kinematics. The ideal algorithm will find the joint angles in constant time $O(1)$ irrespective of the number of joint angles or modules. Since we have one computer per modules, our computational power goes up proportionately with the number of modules. Thus, if an algorithm has linear complexity, $O(n)$, where n is the number of modules, and the algorithm is parallelizable, the result will be the same as O(1) allowing an infinite number of modules.

For other computational problems, we must find algorithms that have linear complexity in the number of modules or better, if the computational power is not to limit the number of modules, plus the algorithms should be parallelizable to run on each module.

7.6 Summary of Limitations

In this summary we will use dimensional analysis to determine the relative effects of the issues presented. We will use the term s to represent a characteristic length of one dimension of a module, so the number of modules is proportional to s^3.

- Buckling under self weight and Euler buckling limits the number of vertical modules in a single module width chain. If we scale the modules, changing widths as well as length, the amount of weight supportable by the structure is proportionate to I/L^2 which reduces to s^2, while the weight increases by s^3. Thus performance degrades proportionate to s.

- Actuator strength increases by s^2, while the mass (load) increases by s^3. Again performance degrades proportionate to s.

- Non-actuated power consumption varies by $s^3 + C$ where C is some constant, typically large compared to s. Here, performance is enhanced slightly by a factor of $\frac{s^3+C}{s^3}$,

For the next items, no measure is mentioned as the limitation is based on design.

- If communications is required at every module, serial communication speed degrades by s^3. A hierarchical communications structure can achieve $\log(s^3)$ though in more practical situations will degrade proportionate to s.

- The computational power available is s^3 (at best?). The amount of computation that is required depends on the task and the algorithm. We present a constant time algorithm for inverse kinematics in Appendix A. We also use the independent joint control method which needs s^3 computational power.

Chapter 8

Conclusions

8.1 Summary

The capabilities for modular reconfigurable robots are large. We have explored just one possible task in detail, statically stable locomotion. For this case, we have shown that a unit-modular approach is potentially very effective for achieving versatility. With the added ability to reconfigure, the robot becomes even more versatile as it can dynamically adapt to changing situations. In fact, the unit-modular approach is so versatile that we must pose the question of not what it can do, but rather what can it not do. We presented some analysis techniques and issues for finding the limits of the total number of modules that can be combined.

This versatility is examined for the task of statically stable locomotion. A functional taxonomy of locomotion is created in order to study the possible forms of locomotion. All of the analysis is meaningless if it cannot be applied to a real robot. Thus, we introduce the design and issues involved in the building of Polypod. It is shown that Polypod can implement all classes of locomotion in the taxonomy. This is the only robot that has been shown to be able to do this as far as the author has found.

In implementing these locomotion gaits a scalable control strategy had to be developed such that an arbitrary number of modules (and thus degrees of freedom) could be added without affecting the computational time complexity for the given computational resources. An added result is that many of the modes implemented can be applied to an arbitrarily large number of modules. The key element to the control is simplicity. Since, currently, each gait is designed by a human, this simplicity facilitates the development of novel gaits.

An analysis of the terrain effects on each form of locomotion is then addressed along with Polypod's performance for stability, efficiency, obstacle crossing and payload. It is seen that a different class of locomotion tends to perform better for each of these performance parameters on Polypod.

In studying locomotion in general and from the experiments with Polypod, these recommendations are reached for designing vehicles for given tasks:

- Pointed feet are better for discrete little-footed gaits as pointed feet reduce the number of DOF needed for each generalized foot, or it reduces the loss in motor efficiency that can be caused by sliding.
- Rolling gaits tend to be more efficient than swinging gaits.
- Articulated gaits tend to be very good for cluttered environments, and in most cases would be the optimal design for traversing what would be obstacles for other vehicles.

8.2 Contributions

In this thesis the following contributions were presented:

- Analysis of the problems involved in designing, building, and controlling a unit-modular reconfigurable robot by designing, building and controlling such a robot.
- A solution to the design of a modular robot which can combine into a large set of morphologies and which has a relatively large range of motion.
- A simple scalable modular control strategy that is easy to implement and requires little resources yet can generate complex results.

- A functional taxonomy for statically stable locomotion with analysis of the different classes of locomotion for different terrain tasks.

Furthermore, we introduce and develop the following new concepts:

- A unit-modular dynamically reconfigurable robot can be extremely versatile. While the concepts of modularity, reconfigurability and unit-modularity have existed, and the versatility of a modular reconfigurable approach has been studied, the versatility of a unit-modular *and* dynamically reconfigurable approach has not been researched.
- The decomposition of locomotion into simple gaits, compound and turning gaits, and path planning/navigation, with the property that gaits can be combined into hierarchically combined or articulated gaits.

8.3 Future Work

The field of micro-robotics holds much promise for unit-modular reconfigurable robots. At this time a robot such as Polypod is not feasible on the micro-robot scale. However, many of the techniques presented in this thesis do not require a robot as complex as Polypod.

The first control strategy presented does not need a master nor inter-processor communications, and the minimal synchronous control strategy does not need position feedback from the actuators - only an on/off limit switch identifying when a joint limit has been reached.

Statically stable locomotion is also appropriate for micro-robotics as friction effects greatly outweigh inertial effects at this level making dynamically stable land locomotion difficult. The property of pointed feet also applies to micro-robots. The one drawback to small feet is that locomotion on soft soils is improved with larger feet, however for micro-robots, soil sinkage will not be a concern as surface tension and other surface effects out way any gravity ones.

Another interesting area to investigate further would be the automatic generation of gaits. That is given a set of modules, what configurations and gait control tables will result in locomotion? Currently, all of the implemented gait control tables were generated manually. The analysis techniques presented earlier could then be used to determine the wanted properties of the generated gait. One method could be the "artificial evolution" approach [Sims 1991][Spencer 1994] that has had some success in simulation.

Linked with this problem is the determination of morphologically distinct configurations. If we assume that the robot starts in an initial configuration and then must reconfigure itself to attain the final configuration, morphologically distinct configurations are not enough as we must plan the sequence of reconfigurations to attain that configuration as well (which may not exist).

Several subjects which were touched upon but which deserve further investigation include:

- the surface and solid inverse kinematics
- studying the reconfigurability of Polypod. Initially the reconfiguration space would be the space of all possible configurations the robot could take
- a redesign of Polypod to investigate dynamically stable locomotion

There are many issues left open from this thesis. As with knowledge in general, the more questions we try to answer the more questions we are left with.

Appendix A

Inverse Kinematics of Snake-Like Arms

A.1 Introduction

Solving for the joint angles of a serial chain, when given the desired position and orientation of the end-effector[1] is called inverse kinematics. Inverse kinematics for an arm in a three dimensional workspace involves six degrees of freedom (DOF), three Cartesian DOF and three angular DOF . Highly redundant manipulators have many more DOF than six. With the development of Polypod [Yim 1993, 1994] and other modular robots, a robot with one hundred degrees of freedom is feasible.

The most related work on inverse kinematics for highly redundant manipulators is that of Chirikjian and Burdick [Chirikjian 1991]. They introduce the idea of using a "backbone" curve, and fitting the robot to the curve.

[mention work of banon,kobayashi,Lumelsky? here.]

We present a closed form inverse kinematics method whose complexity is constant-time, that is independent of the number of DOF. Chirikjian and Burdick's method is not independent of the number of DOF, the fitting process is at best linear in the number of DOF, and typically worse, depending on the fitting method used. [

[1] The position and orientation of the end-effector is an imaginary frame that we attach to the distal point of a serial chain

mention that they do have a modular method ... problems with it?] As the number of DOF gets higher and higher this can clearly become a problem for real time control.

Our method applies to repeated sets of joints, which can be made up of one or more unit-modules. We further restrict our method to modules which consists of at least one revolute and one prismatic joint. In particular we implement this inverse kinematics method on Polypod, both in simulation and on the physical robot. To our knowledge, this is the first time an inverse kinematics method that is scalable to an arbitrarily large number of DOF has been explicitly implemented for the full six dimensional case. Although this case is theoretically possible with Chirikjian/Burdick's method, they have yet to explicitly show how to do it.

Method Overview: The inverse kinematics problem can be viewed as connecting a base frame $\{B\}$ to a goal frame $\{G\}$ with a robot as in Figure A.1. We use a two step procedure using the idea of a backbone curve. First a curve is found which connects the Z-axis' of the two frames, then the robot is fitted to this curve.

This curve has C1 continuity, it is tangent to the Z-axis and approaches $\{B\}$ from the positive side and $\{G\}$ from the negative side. The curve used is a c1 continuous set of arcs of circles.

A.2 Generating the Backbone Curve

The backbone curve that we use is composed of three arcs of circles attached end to end. Each end of each curve is tangent to the adjacent curve or tangent to the Z-axis of $\{B\}$ or $\{G\}$ ($\hat{Z}_B$ or $\hat{Z}_G$) as in Figure A.2. The six parameters needed to describe these three arcs are θ_1, θ_2, θ_3, r_1, r_2, r_3 where θ and r correspond to the angle and radius of the subscripted arc respectively.

The angles of the planes in which each arc of circle lie depends on the link configuration (i.e. Denavit-Hartenberg parameters.) We will present the analysis for a robot in which all non-arbitrary twist angles, α, are 90 degrees. That is each joint axis is perpendicular to its neighbors. While this simplifies the analysis, it does not prevent the method from being applied to other robots.

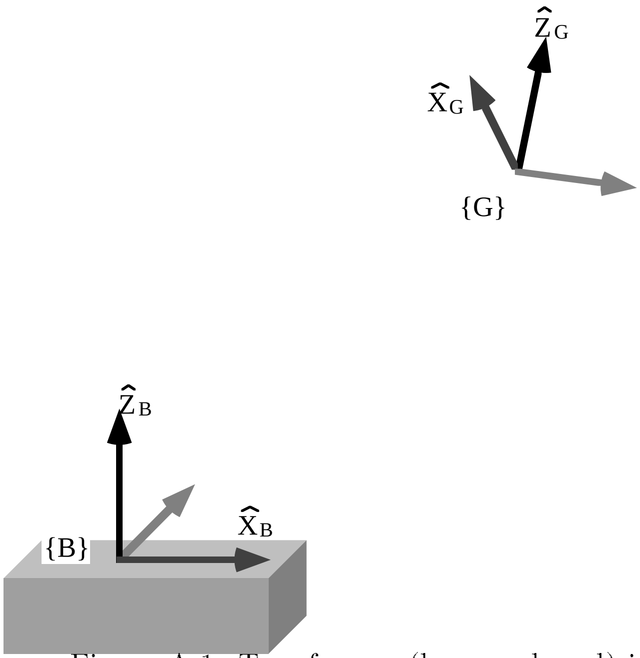


Figure A.1: Two frames (base and goal) in space

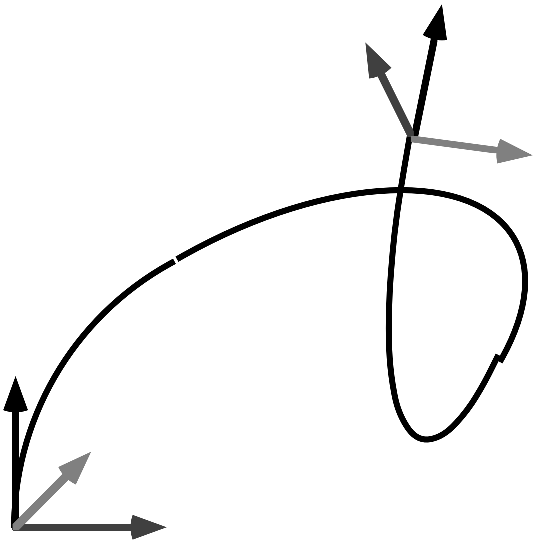


Figure A.2: Three arcs connecting the two frames

The first arc of circle lies in the X-Z plane of $\{B\}$, the third lies in the X-Z plane of $\{G\}$, and the second arc lies in a plane that is perpendicular to both as shown in Figure A.2.

Solving for the six parameters θ_i and r_i (for $i = 1, 2, 3$) can be done by examining the geometry. This breaks down into three steps

1. find the θ' : (-180:180] by analyzing the geometry.

2. solve the kinematic constraints for r' : $(-\infty, +\infty)$.

3. convert θ' and r' to be within valid ranges for θ : (-360,360] and r : (0,+∞).

A.2.1 Finding θ'

It turns out that since we specified the link configuration to be all mutually perpendicular, the θ' values for the arcs correspond to the Z-Y-Z Euler angle representation for orientation.

Representations of orientation can also be interpreted as rotation operators. The Z-Y-Z Euler angle operator for rotating a frame $\{G\}$ as described in [Craig 1986] follows:

> Start with the frame coincident with a known frame $\{B\}$. First rotate $\{G\}$ about $\hat{Z}_G$ by an angle θ_1, then rotate about $\hat{Y}_G$ by an angle θ_2, and then rotate about $\hat{Z}_G$ by an angle θ_3.

If the representation is not already in the Z-Y-Z Euler angle representation, most representations are easily converted to the rotation matrix representation. The equivalent rotation matrix representation of the Z-Y-Z Euler angles (also found in [Craig 1986]) is shown below.

$$R = \begin{bmatrix} c1c2c3 - s1s3 & -c1c2s3 - s1c3 & c1s2 \\ s1c2c3 + c1s3 & -s1c2s3 + c1c3 & s1s2 \\ -s2c3 & s2s3 & c2 \end{bmatrix} \tag{A.1}$$

where H is the rotation matrix,ci is Cos(θ'_i) and si is Sin(θ'_i). If we substitute h_{ij} for the elements in the matrix,

$$R = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{A.2}$$

then the solution for extracting the Z-Y-Z Euler angles from a rotation matrix follows by simple algebra and trigonometry.

$$\theta'_1 = \text{Atan2}(h_{23}, h_{13}) \tag{A.3}$$

$$\theta'_2 = \text{Atan2}(\sqrt{h_{31}^2 + h_{32}^2}, h_{33}) \tag{A.4}$$

$$\theta'_3 = \text{Atan2}(h_{32}, -h_{31}) \tag{A.5}$$

When $\theta'_2 = 0$ or 180 degrees, H becomes singular and $h_{23}, h_{13}, h_{32}, h_{31} = 0.0$. One solution is to set $\theta'_1 = \theta'_3$. This way we can distribute the angle over the two arcs, which will have a tendency to lessen the maximum angle needed for any individual module.

If $\theta'_2 = 0$ degrees then

$$\theta'_1 = \theta'_2 = \text{Atan2}(-r12, r11)/2 \tag{A.6}$$

If $\theta'_2 = 180$ degrees then

$$\theta'_1 = \theta'_2 = \text{Atan2}(r12, -r11)/2 \tag{A.7}$$

For any non-singular rotation matrix there are exactly two solutions, one with θ'_2 in the open range (-180,0) and the other in the range (0,180). Given one solution, the second solution can be found by the following operations:

$$\theta'_1 = \theta'_1 + 180.0 \tag{A.8}$$

$$\theta'_2 = -\theta'_2 \tag{A.9}$$

$$\theta'_3 = \theta'_3 + 180.0 \tag{A.10}$$

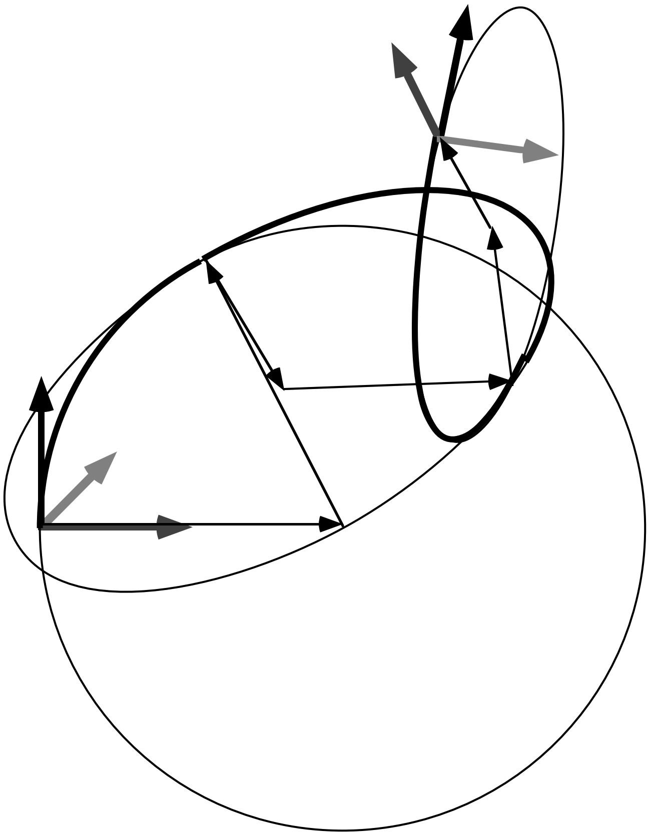


Figure A.3: Three arcs connecting the two frames with radius vectors

If any angle is greater than 360, then 360 is subtracted from that angle, so θ_i' is always in the range (-180,180].

For link configurations which are not mutually perpendicular, equations can still be found to get θ' values. The simplest method is probably to examine the rotation matrix and generate equations algebraically.

A.2.2 Finding r'

Once the θ_i' values have been found the three r_i' values can be found by solving three constraint equations. One set of constraint equations describes the position vector $\vec{P}$ from $\{B\}$ to $\{G\}$ as the sum of the radial vectors of the arcs as in figure A.3.

Let $\{F1\}$ denote the frame found by rotating $\{B\}$ by θ_1 about $\hat{Z}_B$ and translating the origin to the end point of the first arc. Similarly, let $\{F2\}$ denote the frame found by rotating $\{F1\}$ by $\hat{Z}_{F1}$ and translating the origin to the end point of the second arc.

$$\vec{P} = r'_1\hat{X}_B + (r'_1 + r'_2)\hat{X}_{F1} + (r'_2 + r'_3)\hat{Y}_{F2} + r'_3\hat{X}_G \tag{A.11}$$

Thus in matrix form

$$\begin{bmatrix} \vec{P}\cdot\hat{X}_B \\ \vec{P}\cdot\hat{Y}_B \\ \vec{P}\cdot\hat{Z}_B \end{bmatrix} = \begin{bmatrix} (\hat{X}_B + \hat{X}_{F1})\cdot\hat{X}_B & (\hat{X}_{F1} + \hat{Y}_{F2})\cdot\hat{X}_B & (\hat{Y}_{F2} + \hat{X}_G)\cdot\hat{X}_B \\ (\hat{X}_B + \hat{X}_{F1})\cdot\hat{Y}_B & (\hat{X}_{F1} + \hat{Y}_{F2})\cdot\hat{Y}_B & (\hat{Y}_{F2} + \hat{X}_G)\cdot\hat{Y}_B \\ (\hat{X}_B + \hat{X}_{F1})\cdot\hat{Z}_B & (\hat{X}_{F1} + \hat{Y}_{F2})\cdot\hat{Z}_B & (\hat{Y}_{F2} + \hat{X}_G)\cdot\hat{Z}_B \end{bmatrix} \begin{bmatrix} r'_1 \\ r'_2 \\ r'_3 \end{bmatrix} \tag{A.12}$$

and in condensed form

$$\vec{P} = A\vec{r} \tag{A.13}$$

To solve for $\vec{r}$ we just invert A which is always a 3x3 matrix and has a simple closed form solution.

A.2.3 Converting r' and θ' to r and θ

The range of solutions returned for r and θ by the above method is $[-\infty, +\infty]$ and $(-180, 180]$ degrees respectively. However, the range of valid solutions barring joint limit and auto-collision constraints is $(0, +\infty]$ and $(-360, 360]$ respectively. The mapping between the valid solutions and the obtained ones is as follows.

$$r_i = \begin{cases} r'_i & \text{if } r'_i > 0, \\ -r'_i & \text{if } r'_i < 0, \\ \text{invalid} & \text{if } r'_i = 0 \end{cases} \tag{A.14}$$

$$\theta_i = \begin{cases} \theta'_i & \text{if } r'_i > 0, \\ \theta'_i + 360 & \text{if } r'_i < 0 \text{ and } \theta'_i < 0, \\ \theta'_i - 360 & \text{if } r'_i < 0 \text{ and } \theta'_i > 0, \\ \text{invalid} & \text{if } r'_i = 0 \text{ or } \theta'_i = 0. \end{cases} \tag{A.15}$$

A.3 Fitting the Robot to the Curve

The ideal fitting of the robot to the curve would have the end-effector frame coincide exactly with $\{G\}$, and all other points on the robot, a minimum distance away from the curve.

The chain is divided into 3 sections proportionate to the arc length of each arc. Each section then is divided into units of 3 modules (which thus have 6DOF) and is has the minimum requirements to use the 3 arc method, each module forms one arc. Thus the frame of every 3rd module lies exactly on the curve.

A.4 2D Workspace Planar Case

The previous method degenerates for the planar case since the middle arc should lie in a plane perpendicular to the X-Z planes of $\{B\}$ and $\{G\}$. However, they are parallel, so there is no unique angle for the perpendicular plane normal. For this case we propose a two arc method.

Assuming we start at an origin $\{B\}$, we are solving for three parameters, two position and one angle. Two arcs of circles present four parameters, radius (r_1, r_2) and arc length for each arc (θ_1, θ_2). We remove r_2 by constraining the radius of each circle to be the same $(r = r_1 = r_2)$. The consequence of doing all this is presented later in the discussion section.

We first solve for r by summing the four vectors of length r formed by joining each center of each arc with $\{B\}$ and $\{G\}$ as in Figure A.4. The $\hat{X}_B$ component of the line segment joining the two centers of the arc's is,

$$x - r - r\cos(\theta) \tag{A.16}$$

and the $\hat{Y}_B$ component is

$$y - r\sin(\theta) \tag{A.17}$$

where x and y are the $\hat{X}_B$ and $\hat{Y}_B$ components of the line segment joining $\{B\}$ and $\{G\}$, and θ is the angle between $\hat{X}_B$ and $\hat{X}_G$.

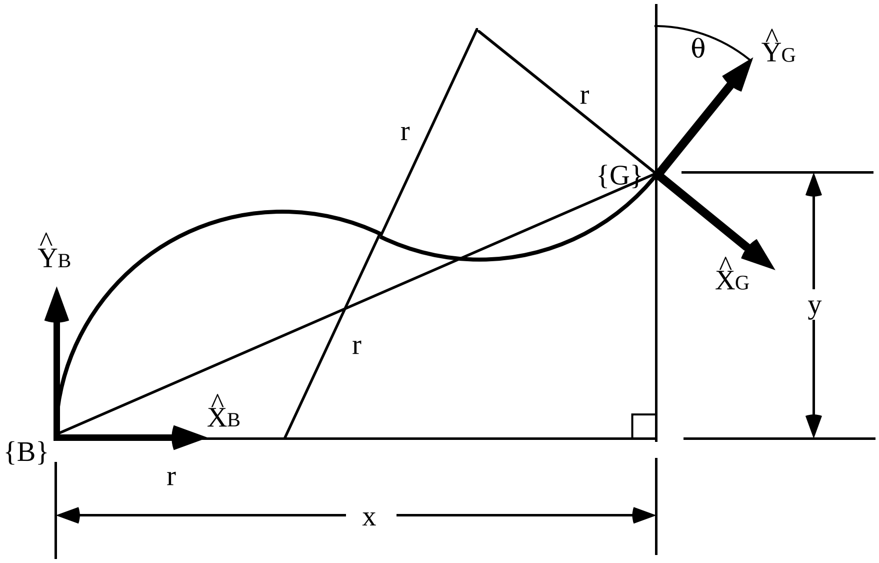


Figure A.4: Planar case composed of two arcs of circles

We know the length of the segment joining the two centers is $2r$, so,

$$(x - r - r\cos(\theta))^2 + (y - r(\sin\theta))^2 = 4r^2 \tag{A.18}$$

which can be rearranged to be a simple quadratic equation in r:

$$ar^2 + br + c = 0 \tag{A.19}$$

where $a = 4-(1-\cos(\theta))^2-\sin(\theta)^2$,$b = -2x((1-\cos(\theta))+y\sin(\theta)$, and $c = -x^2-y^2$.

The arc lengths θ_1, θ_2 are solved with the following equations:

$$\theta_1 = acos\left(\frac{\hat{X}_B\cdot}{2r}\right) \tag{A.20}$$

$$\theta_2 = \theta - \theta_1 \tag{A.21}$$

Once r, θ_1 andθ_2 are found, the robot is fitted to the curve in the same manner as the three arc method.

A.5 Discussion

For the two arc method for planar inverse kinematics, it can be shown that this method provides the locally minimum angle that any revolute joint in a module would have to attain to reach the goal. This is especially advantageous for highly redundant modular robots since typically the revolute joints have a very restricted range $\approx \pm 20$ degrees (see Section 1.4). In addition this method has the characteristic that more modules increases the accuracy of curve matching and decreases the minimum angle

For the full six dimensional case the same optimality applies but only to the class of modules that cannot provide a twist angle.

Analogies to 3R3P: An analogy can be made between the three arc method and inverse kinematics for manipulators with three revolute joints and three prismatic joints (non-redundant). Pennock and Vistra presented a kinematics method for a three-cylindric robot, where cylindrical joints are revolute (R), prismatic (P) joint pairs [Pennock 1990]. However Roth and Raghavan showed that the inverse kinematics for any 3R3P robot is quadratic, irrespective of the arrangement of R's and P's [Raghavan 1993]. They do this in a similar way that we solve for the 6 variables in the back bone curve, solving the revolute joints (analogous to θ_i) and prismatic joints (analogous to r_i) separately.

The main difference here is that the arcs cannot be modeled by a simple revolute joint and prismatic joint. They must be modeled as three joints with some constraint on them as the segments are.

Singularities: A problem with ZYZ Euler angles and with the 3 arc method is that a singularity exists in the exact middle of the work space when the goal and base frames differ by only a translation along the Y-axis. The 3 arcs reduce to a straight line, $r_1 = r_2 = r_3 = \infty$ and the arc lengths are underconstrained. This is especially difficult when this is the nominal position that a modular robot chain may reside, for example if the chain represents one link in a lattice structure.

Complexity: The complexity of the backbone curve is O(1) as it is closed form and independent of the number of modules or DOF. The fitting process is also constant time since once θ_i and r_i (for i=1,2,3) are found and the dividing points are determined, all joint angles are known. In contrast, fitting methods for arbitrary curves require each joint angle to be calculated and is thus usually linear in the number of joints or worse.

Appendix B

Balancing with –springs on the Revolute DOF

The *–springs* mode on rDOF acts to keep the vertical projection of the center of gravity in the middle of the segment supporting the rest of the robot. Figure B.1 shows a free body diagram of a supporting foot segment. Since the robot should be in static equilibrium:

$$\mathbf{M} = \mathbf{R} \times \mathbf{F} \tag{B.1}$$

where **M** and **F** are the moment and force vectors resp. seen at the top of the segment due to the weight of the rest of the robot. **R** is the distance vector from the reaction force on the ground to **F**. We may evaluate Equation B.1 to the scalar and vector components:

$$M\mathbf{k} = FR\sin(\psi)\mathbf{k} \tag{B.2}$$

where **k** is the normal vector out of the page, M, F and R are the magnitudes of their respective vectors. From the figure we see

$$R\sin(\psi) = d\sin(\theta) \tag{B.3}$$

so substituting Equation B.3 into B.2 and replacing M with the sensed torque T:

$$T = Fd\sin(\theta) \tag{B.4}$$

Since the pDOF is in *no* mode, d is constant. The weight of the robot, F, is also constant, so we may substitute a constant K for F d, and using the small angle

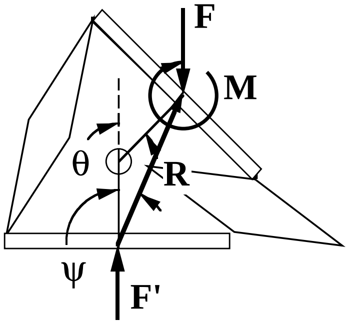


Figure B.1: Free body diagram of supporting segment

approximation yield:

$$T = K\theta \tag{B.5}$$

which is the negative form of the *springs* mode.

The angles are not actually small, so this approximation is not valid, however the large width of the foot will accommodate the errors here. Since we have imposed quasi-static motions on the actuators, the control will be stable.

Appendix C

Vehicle Figures

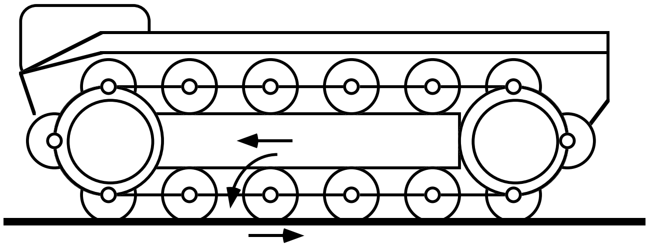


Figure C.1: Airoll (Ingersoll). Wheeled track combination. In soft soil track motion provides grousing action.

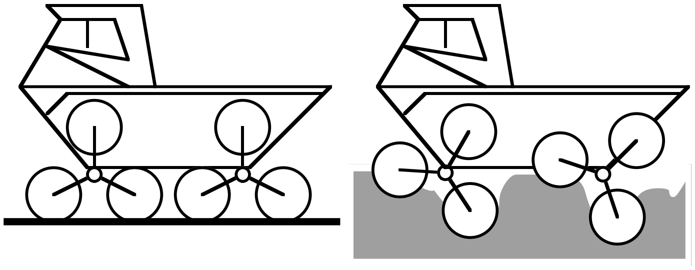


Figure C.2: Lockheed-Forsyth. Wheeled wheel combination. In soft soil superior wheel motion provides grousing action.

Appendix D

Task Parameter Example Derivations

D.1 Earthworm Vehicle Parameters

For all the vehicle parameters in Table 6.1 we assume that the robot has some algorithm that will allow the robot to take whatever pose would maximize the vehicle parameter when confronted with the associated obstacle with the given configuration.

Ditch Crossing V_d: For V_d, this would mean the robot assumes the longest possible form in order to stretch out over the ditch. Segments and nodes have the same length when fully extended, 2.25 inches (2.0 inches for the actuator plus 0.25 inches for the connecting plates). Since the Earthworm has all nodes and segments in a line, the maximum ditch length it can cross is simply one half the sum of the lengths of the modules. $V_d = 1.125sn$ inches where s is the number segments, and n is the number of nodes.

If we do not have the assumption that the robot is smart enough to optimize its motions, the Earthworm would have $V_d = 0.75s + 1.125n$ inches. The segments now have a nominal length of 0.75 inches instead of fully extended since half of the segments are fully compressed and half are fully extended during the Earthworm locomotion. Also, the motions are straight lines through the RPjoint space, so the sum of the lengths do not change even between steps.

Hang-up Crossing V_hu: The Earthworm has one G-foot comprising the entire underside of the robot. No sphere can be placed underneath the robot which can touch a non-G-foot portion of the body. $V_hu = 0$.

Barrier Crossing V_b: The maximum height that the Earthworm can reach is equal half the maximum length of the robot minus the amount needed to bend in order to reach up, ($V_b = 1.125((s-2)+n)$ when crossing a perpendicular wall). Since the angle between the supporting ground and the barrier is not known, the amount needed to bend is unknown. For Table 6.1 we ignore the (-2) value since it is uncertain and most likely small compared to s and n.

Height Constraint V_h: The minimum height that the earthworm can obtain is the width of a segment (y-axis direction in Figure 2.4) when fully extended, 2.25 inches. Note that the robot can still locomote as long as a height constraint has some non-zero distance above V_h, each step gets smaller and smaller approaching zero as an obstacle approaches V_h.

Width Constraint V_w: The width constraint does not effect motion since the width profile of the Earthworm gait does not change when moving in a straight line. V_w is the horizontal width of the segment (z-axis direction in Figure 2.4) which is 2.25 inches.

Curvature Constraint V_c: The volume swept out by the turning Earthworm has a radius of 3.8 inches. This radius is composed of 4 segments angled at 45 degrees and 4 fully collapsed segments whose plane of action is perpendicular, interspersed between these 4 segments. The fully collapsed segments would normally be used for forward locomotion. When fully collapsed the width (in the x-axis direction of Figure 2.4) is 0.75 inches (0.5 inches for the actuator plus 0.25 for the connection plates).

D.2 Rolling-Track Efficiency

This section will explain each parameter in Equation 6.2 for the Rolling-Track.

Distance per cycle, d: The distance travelled in one cycle of the gait is 2.5 inches. In one cycle of the gait, two segments diametrically opposite go from fully angled to nominal +*springs* position, $D = 1.0''$ (where D is the prismatic DOF length). At the same time two others segments do the reverse. The distance moved is equal to the distance between the right side of segment #1 in Figure 5.1 and the right side of segment #16. However, it is assumed that track is composed of alternating perpendicular segments (in case the robot might need to turn) and that those segments are running in +*springs* mode. Thus there is an extra segment between #1 and #16. The distance d is then equal to twice the nominal width (in the x-axis direction of Figure 2.4) running in +*springs* mode, 2.5 inches (2.0 for two actuators plus two pairs of connection plates 0.5).

Number of full DOF moves, $\Updownarrow$: The number of times the DOF move through its full range in *ends* mode $\Updownarrow$ for one step is 4, the two segments going from full angled to $D = 1.0$ and two more doing the reverse. Each segment has two DOF going through half of its full range of motion, so we count each segment as adding one to $\Updownarrow$.

Number of -*springs* mode DOF, $\rightarrow$: There are no DOF in -*springs* mode, $\rightarrow = 0$.

Modules between ground contact points $N(s), N(n)$: The maximum number of modules between two ground contact points occurs in the top part of the loop. This is equal to half of all the modules in the straight portions ((s-8)/2 + (n-8)/2), plus the two curved portions (8 segments, 8 nodes). $N(s) = s/2 + 4, N(n) = n/2 + 4$.

D.3 Earthworm Stability

For an Earthworm robot made up of more than seven segments, the minimum distance from the CG to the support polygon is along the transverse axis. The width of one segment (z-axis direction of Figure 2.4) defines the width of the support polygon along this axis, 2.25 inches. The CG lies in the middle of this width, so $R_X = 1.125$ inches. The height of the CG is one half the width of one fully compressed segment (x-axis direction in Figure 2.4) $R_Y = 4.0/2$ inches.

Bibliography

[Atluri 1988] Atluri, S. N.,Amos, A. K.,editors *Large Space Structures: Dynamics and Control*, Berlin, Springer-Verlag, 1988.

[Bazant 1980] Bazant, Z., P., Christensen, M., "Analogy Between Micropolar Continuum and Grid Frameworks Under Initial Stress," *Int. J. Solids & Struct.,* Vol. 21, No. 2, pp. 249-263 (Feb. 1980).

[Bekkar 1969] Bekkar, M. G., *Introduction to Terrain-Vehicle Systems*, Ann Arbor MI, The University of Michigan Press, 1969.

[Burdick 1993] Burdick, J. W., Radford, J., and Chirikjian, G. S., "A "Sidewinding" Locomotion For Hyper-Redundant Robots," *Proc. of the 1993 IEEE Int. Conf. Robotics and Automation*, pp. 101-106.

[Byrd 1990] Byrd, J. S. and DeVries, K. R., "A Six-Legged Telerobot for Nuclear Applications Development," *The Int. J. of Robotics Research*, Vol. 9 No. 2, pp. 43-52, (1990).

[Chen 1993] Chen, I., et. al., "Enumerating Non-Isomorphic Assembly Configurations of Modular Robotic Systems," *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS)* pp. 1985-1992, 1993.

[Chirikjian 1991] Chirikjian, G. S. and Burdick, J. W., "Kinematics of Hyper-redundant Locomotion with Applications to Grasping," *Proc. of the 1991 IEEE Int. Conf. Robotics and Automation.*

[Chirikjian 1993] Chirikjian, G. S. "Metamorphic Hyper-Redundant Manipulators," *Proc. of JSME Int. Conf. on Advanced Mechatronics*, pp. 467-472 1993.

[Craig 1986] Craig, J. J., *Introduction to Robotics Mechanics & Control*, Reading MA., Addison-Wesley Publishing Co., 1986.

[Fukuda 1988] Fukuda, T. and Nakagawa, S., "Dynamically Reconfigurable Robotic System," *Proc. of the 1988 IEEE Int. Conf. Robotics and Automation*, pp. 1581-1586.

[Fukuda 1990] Fukuda, T. and Kawauchi, Y., "Cellular Robotic System(CEBOT) as One of the Realization of Self-organizing Intelligent Universal Manipulator," *Proc. of the 1990 IEEE Int. Conf. Robotics and Automation*, pp. 662-667.

[Fukuda 1990] Fukuda, T. and Kawauchi, Y., "Analysis and Evaluation of Cellular Robotics (CEBOT) as a Distributed Intelligent System by Communication Information Amount," *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS)* pp. 827-834, 1990

[Full 1993] Full, R. J., et. al., "Locomotion Like a Wheel?," *Nature*, Vol. 365, No. 7, pp. 495 (Oct. 1993).

[Gray 1968] Gray, J., *Animal Locomotion*, W. W. Norton & Co. Inc., New York NY, 1968.

[Hirose 1981] Hirose, S., Umetani, Y., and Oda, S., "An active cord mechanism with oblique swivel joints and its control." *Proc. 4th ROMANSY Symp.*,Zaborow, Poland, pp. 395-407, 1981.

[Hirose 1984] Hirose, S, "A Study of Design and Control of a Quadruped Walking Vehicle" *The Int. J. of Robotics Research*, Vol. 3 No. 2, pp. 113-133, (1984).

[Hirose 1990] Hirose, S, and Morishima, A., "Design and Control of a Mobile Robot with an Articulated Body," *The Int. J. of Robotics Research*, Vol. 9, No. 2, pp. 99-114, (1990).

[Iagolnitzer 1992] Iagolnitzer, M., et. al., "Locomotion of an all-terrain mobile robot," *Proc. of the 1992 IEEE Int. Conf. Robotics and Automation*, pp. 104-109.

[Kawauchi 1992] Kawauchi, Y., Inaba, M., and Fukuda, T., "Self-organizing Intelligence for Cellular Robotic System "CEBOT" with Genetic Knowledge Production Algorithm," *Proc. of the 1992 IEEE Int. Conf. Robotics and Automation*, pp 813-818.

[Koralov 1992] Koralov, K. D., *Optimal Geometric Design of Robots for Environments with Obstacles*, Stanford Univ. Dept. of Mech. Eng., Ph.D. Dissertation, 1992.

[Maes 1990] Maes, P., Brooks, R.A., "Learning to coordinate behaviors," *AAAI-90 Proceedings. Eighth National Conf.on Artificial Intelligence.* MIT Press, p. 796-802.

[McGeer 1990] McGeer, T., "Passive Dynamic Walking," *The Int. J. of Robotics Research*, Vol. 9, No. 2, pp. 62-82 (April 1990).

[Messuri 1983] Messuri, D., Klien, C., "Automatic Body Regulation for Maintaining Stability of a Legged Vehicle During Rough-Terrain Locomotion," *IEEE J. of Robotics and Automation*, Vol. RA-1 No. 3. pp. 132-141 (Sept. 1983).

[Noor 1988] Noor, A. K., "Continuum Modeling of Large Lattice Structures: Status and Projections", *Large Space Structures: Dynamics and Control*, Berlin Germany, Springer-Verlag, pp.1-34 1988.

[Ott 1988] Ott, Henry W., *Noise Reduction Techniques in Electronic Systems*, John Wiley & Sons Inc., 1988.

[Pennock 1990] Pennock, G. R.,Vistra, B. C. "The Inverse Kinematics of a Three-Cylindric Robot," *The Int. J. of Robotics Research*, Vol. 9, No. 4, pp. 75-85 (Aug. 1990).

[Pugh 1990] Pugh, D. R., et. al., "Technical Description of the Adaptive Suspension Vehicle," *The Int. J. of Robotics Research*, Vol. 9, No. 2, pp. 24-42 (April 1990).

[Raghavan 1993] Raghavan, M., Roth, B., "Inverse Kinematics of the General 6R Manipulator and Related Linkages," *Transactions of the ASME J. of Mechanical Design*, Vol.115, No.3, p. 502-8 (Sept. 1993).

[Raibert 1990] Raibert, M., "Forward: Special Issue on Legged Locomotion," *The Int. J. of Robotics Research*, Vol. 9, No. 2, pp. 2-3 (April 1990).

[Salisbury 1988] Salisbury, K., "Whole Arm Manipulation," *Robotics Research: The Fourth Intl. Sympm.*, pp. 183-189 (1988).

[Sarrus 1839] Sarrus and Ramaux, Rapport sur un memoire adresse a l'Academie Royal de Medicine. *Bull. Acad. Med.*, Paris, Vol 3, pp. 1094-1100 (1839).

[Schmidt 1983] Schmidt-Nielsen, K. *Animal Physiology: Adaptation and environment*, Cambridge England, Cambridge University Press, 1983.

[Simmons 1991] Simmons, R., Krotkov, E., "An Integrated Walking System for the Ambler Planetary Rover," *Proc. of the 1991 IEEE Int. Conf. Robotics and Automation*, pp. 2086-2091.

[Sims 1991] Sims, K., "Artificial Evolution for Computer Graphics," *Computer Graphics*, Vol. 25, No. 4, p. 319-28 (July 1991).

[Sinha 1992] Sinha, P. R., Bajcsy, R. K., "Robotic Exploratoin of Surfaces and its Application to Legged Locomotion," *Proc. of the 1992 IEEE Int. Conf. Robotics and Automation*, pp. 221-226.

[Song 1986] Song, S. M., Waldron, K. J., "Geometric Design of a Walking Machine for Optimal Mobility," *J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 109, pp. 21-28 (March 1987).

[Spencer 1994] Spencer, G., "Automatic Generation of Programs for Crawling and Walking," *Advances in Genetic Programming*, edited by Kinnear, K. E. Jr., The MIT Press, 1994.

[Timoshenko 1961] Timoshenko, S. P. and Gere, J. M., *Theory of Elastic Stability*, New York NY, McGraw-Hill, 1961.

[Wilcox 1993] Wilcox, B., et. al., "Robotic Vehicles for Planetary Exploration," *Proc. of the 1992 IEEE Int. Conf. Robotics and Automation*, pp. 175-180.

[Wilkie 1977] Wilkie, D. R., "Metabolism and Body Size," *Scale Effects in Animal Locomotion*, Edited by T. Pedley, Academic Press, 1977.

[Wurst 1986] Wurst, K. H. "The Conception and Construction of a Modular Robot System," *Proc. 16th Int. Symp. Industrial Robotics (ISIR)*, pp. 37-44 (1986).

[Yim 1993] Yim, M., "A Reconfigurable Modular Robot with Many Modes of Locomotion", *Proc. of JSME Int. Conf. on Advanced Mechatronics*, pp. 283-288, 1993.

[Yim video 1994] Yim, M., "Locomotion Gaits with Polypod" video, *Video Proc. of 1994 IEEE Int. Conf. Robotics and Automation.*

[Yim 1994] Yim, M., "New Locomotion Gaits," *Proc. of the 1994 IEEE Int. Conf. Robotics and Automation* pp. 2508-2514.