Platonic Beasts: Spherically Symmetric Multilimbed Robots *

Dinesh K. Pai Roderick A. Barman Scott K. Ralph
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z4, Canada
{pai|rodb|ralph}@cs.ubc.ca

Abstract

We describe a new class of spherically symmetric, high degree of freedom robots called "platonic beasts." A robot in this family is kinematically equivalent to a symmetric polyhedron, such as one of the Platonic solids, with identical multi-purpose limbs attached to its vertices. The symmetry and regularity of the design have several advantages including robustness to toppling, novel gaits such as the *rolling* gait, and fault tolerance.

We describe the design and programming of a prototype platonic beast robot that we have built in our lab. The robot has four limbs, each with three degrees of freedom, and is controlled by a network of four embedded 32-bit microcontrollers. We also discuss the general features of these robots, including locomotion using the rolling gait and the implications of its novel features.

1 Introduction

There has been considerable interest in legged robotics from the early 1960s and a recent resurgence of interest in mobile robotics in general (see the surveys in [1, 2, 3, 4, 5, 6]). Most legged robots are designed to operate in a small range of preferred orientations and therefore are vulnerable to toppling. Their degrees of freedom are specialized as legs or as manipulators due to the constraints of minimizing energy consumption during locomotion (see, for

*This work was supported in part by NSERC, the Institute for Robotics and Intelligent Systems, and the BC Advanced Systems Institute. Portions of this paper were presented at the IEEE Int. Conference on Robotics and Automation, 1994

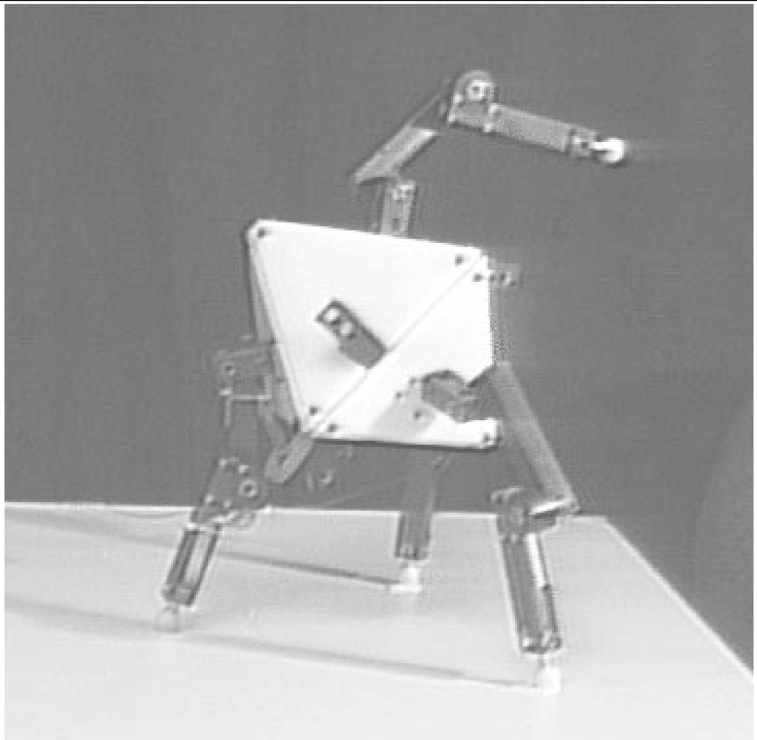

Figure 1: Prototype 4-beast. The robot has 4 limbs, each with 3 revolute joints.

example, the concept of gravitationally decoupled actuation [7]). The arrangements of legs are frequently modeled after those of insects and mammals [8, 9, 10, 11], and knowledge of animal locomotion motivates locomotion algorithms used.

The promise of legged robots as well as directions for future work was demonstrated in August 1994 by NASA's Dante II mission to Mount Spurr [12]. The mission was extremely successful, accomplishing its scientific goals, and demonstrated the utility of legged robots for locomotion on rough, hazardous terrain. However, the robot tipped over on its return and could not recover even though it was physically unharmed, requiring an expensive and dangerous helicopter rescue. Given rough and uncertain terrain, such falls are unavoidable; however, the ability to recover from falls has not yet been paid enough attention.

In this paper we describe a new class of robots called "platonic beasts". These are spherically symmetric, high degree of freedom robots with multi-purpose limbs. These robots are constructed by attaching a kinematic chain, i.e. a limb, at each vertex of a spherically symmetric polyhedron. The polyhedron can be one of the five Platonic solids[1] — hence the

[1]Platonic solids are the only five possible regular convex polyhedra in three dimensions and are the tetrahedron, the cube, the octahedron, the dodecahedron, and the icosahedron.

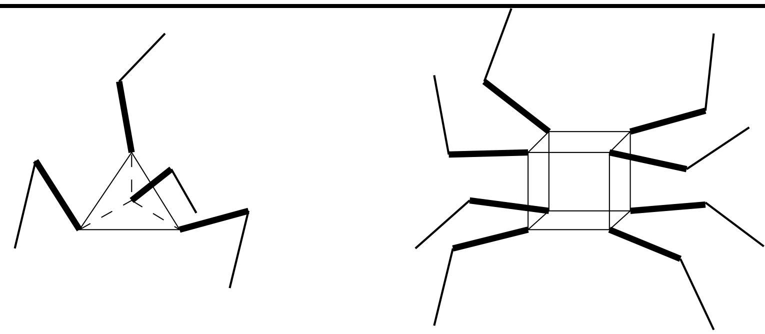

Figure 2: Examples of platonic beasts. The figure sketches the 4-beast, with RRR limbs placed at the vertices of a tetrahedron and an 8-beast with limbs at the vertices of a cube.

name of the family; however, robots based on other spherically symmetric polyhedra such as the Archimedian polyhedra are included in this family as well.

Figure 1 shows a prototype platonic 4-beast, i.e. a 4 limbed robot with a kinematic configuration equivalent to attaching its limbs to the vertices of a tetrahedron, that has been constructed in our lab. Each limb of the robot has three joints, for a total of twelve actuated degrees of freedom. Figure 2 sketches some members of this family, including a 4-beast, and an 8-beast, based on a cube. Figure 12 depicts a 6-beast. Note that the polyhedra are only used to specify the location of the limbs and not necessarily the actual geometry of the body of the robot. For instance, kinematically identical robots are generated by attaching limbs to the centroids of the faces of the dual polyhedra.

Each limb is an independent module with its own sensing, actuation, and computation. The limbs are intended to be used for both locomotion tasks (as "legs") and for manipulation tasks (as "fingers" or "arms")[2].

The unique spherical symmetry of such a robot leads to several novel features not present in conventional legged robots. Platonic beasts are capable of new gaits such as the "rolling" gait in which the robot approximates a rolling motion by synthesizing a sphere with its limbs. Another feature is the robustness with respect to toppling. The robot has no preferred "up" direction and hence can recover from loss of footholds more easily. A certain degree of additional fault-tolerance is possible due the uniform design of limbs and the large number of degrees of freedom (particularly in beasts with more than 4 limbs). The robot can rotate to a configuration in which the defective limbs are not used for locomotion or manipulation.

In this paper, we consider the features of platonic beasts, focusing mainly on the 4-beast, its design, and its the control and programming for locomotion. In section 3 we discuss locomotion and in particular the rolling gait. Section 2 outlines the hardware design of the prototype 4-beast that we have built in our lab. We have tested the prototype and

[2]We call these "limbs" rather than legs or arms or manipulators to emphasize their multiple functions.

demonstrated its ability to locomote using the rolling gait. Section 4 describes the kinematics of the robot required for programming and an interactive graphical simulator that we use for experimenting with robot designs, and verifying robot programs. Finally, in section 5 we discuss the novel features of these robots and their implications.

2 Design of 4-beast Prototype

We have built a prototype of the simplest member of the family, the 4-beast, based on the tetrahedron. Figure 1 is a picture of the prototype. Note that while the kinematic arrangement is based on the tetrahedron, the body need not be – in this case it is in the shape of an octahedron to facilitate limb attachment. The limbs are attached to the centroids of alternate faces; the limb computer and control electronics for a limb are mounted on an adjacent face. The robot is small – it weighs less than 5kg, the body is an octahedron of side 17cm, and each limb has a reach of 25cm from the surface of the body.

The prototype was constructed using the principles of modular design at two levels. First, all limbs of the robot are made of identical UBC-Zebra link modules which can be rapidly assembled to construct kinematic chains (see also [13, 14, 15] for other work on modular robot links). Second, each limb is a module with identical kinematic configuration, control and computing resources.

Figure 3 shows a UBC-Zebra link module used in the design, made by Zebra Robotics to our specifications. The module incorporates a miniature MicroMo DC motor with gearhead and magnetic encoder, a homing switch, and a fixed reduction worm and worm gear. The design of the link allows a variety of gearheads and motors to be used in the link. A single connector supplies power and data connections to the link controller. An adapter at the end of a link permits it to be attached to the drive shaft of another link in different kinematic configurations. The effective link length can be varied by adding an extender. A limb can thus be easily assembled in different kinematic configurations.

Each limb is an identical RRR kinematic chain. For the kinematic configuration we are currently using (see Figure 1) the Denavit-Hartenberg parameters using the notation in [16] of each chain are given in Table 1. The origin of the base coordinate frame is at the center of the body polyhedron. Lengths are in mm. The length a_3 varies depending on the foot module used. Each limb is controlled by a separate limb computer, based on the Motorola MC68332 32-bit microcontroller. The limb computer processes all sensing associated with the limb and communicates with other limb computers and the development host over serial lines. The communication with the host is via a serial line shared by the limb computers. Communication with the host computer uses a simple communication protocol that time multiplexes over the single serial line.

The robot has been designed to operate for short periods with onboard NiCd batteries – the goal is a completely untethered robot. At this time 12V DC power is provided by an

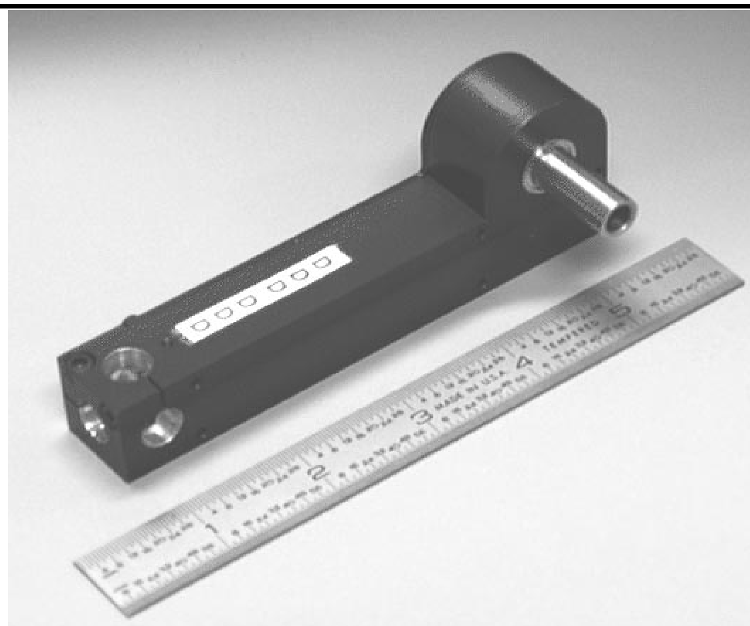

Figure 3: Link Module

Link	d	a	α
1	112.0	0	$\pi/2$
2	0	98.3	0
3	0	140.0	0

Table 1: Kinematic parameters of link modules

external power supply for convenience.

The robot is equipped with joint encoders which, due to the high gear reduction used, provides joint position at a resolution of less than a fraction of a degree[3], much smaller than the joint uncertainties due to backlash. Twelve 1-bit tilt sensors are mounted on the body and can be combined to determine the coarse orientation of the robot with respect to gravity.

The robot prototype has been built and tested in our lab. Motor sizing and power consumption have been verified to satisfy the design specifications – the beast is capable to lifting itself off the ground from even the most difficult configuration (i.e. with the limbs straight out and horizontal) which produces the highest static loads at the joints. Power consumption, including the power consumed by the four MC68332 MCUs, is less that 12W with all limbs moving. Finally, we have verified the ability of the 4-beast prototype to perform the canonical tumble step. Figure 4 shows a typical tumble.

3 Locomotion and the Rolling Gait

In general, if the limbs are long enough relative to the characteristic length of the body, a platonic beast can place four or more of its limbs on the ground simultaneously. In particular, this is true for our 4-beast design. Locomotion is thus possible using the crawl or other statically stable creeping gaits [17].

The spherical symmetry of the platonic beasts permits a new gait called the *rolling gait*. This gait for the 4-beast rolling on level ground can be understood as follows (see Figure 5). The gait is generated by a sequence of isomorphic steps we call *tumble*s. Suppose the limbs are numbered 1 to 4, using a right hand screw rule. A canonical tumble step starts with limbs 1, 2, and 3 in contact with the terrain and limb 4 free, on top of the robot. The contact point of limb i with the terrain (the "foot") is labeled p_i and the projection of the center of mass of robot on the ground is labeled p_0. The points p_1, p_2, and p_3 define the support polygon for the robot. As the beast performs a tumble, p_0 is moved out of the support triangle across the (p_1, p_2) edge, causing the p_3 contact to break. Before the robot tips across the (p_1, p_2) edge, limb 4 is moved into a position such that its foot contacts the ground after the subsequent tip. The ending configuration of the tumble thus has the robot with limbs 1, 4 and 2 on the ground and limb 3 in the air.

Figure 6 shows the gait diagram for the rolling gait along a straight path. The dark bars indicate the duration in which the limb is in the support phase. In the general case there is a brief dynamic "tip" episode during which only two limbs are in contact with the ground. For the choice of kinematic parameters of our prototype, all four limbs can simultaneously contact the ground, and thus the duration of the tip episode can be reduced to zero. Such a zero tip gait will be slower, but will minimize the impact loads on the limb. Note that the zero tip gait is a creeping gait [17] for the 4-beast in that at most one limb is off the ground

[3]Theoretically less than 0.0008 degree.

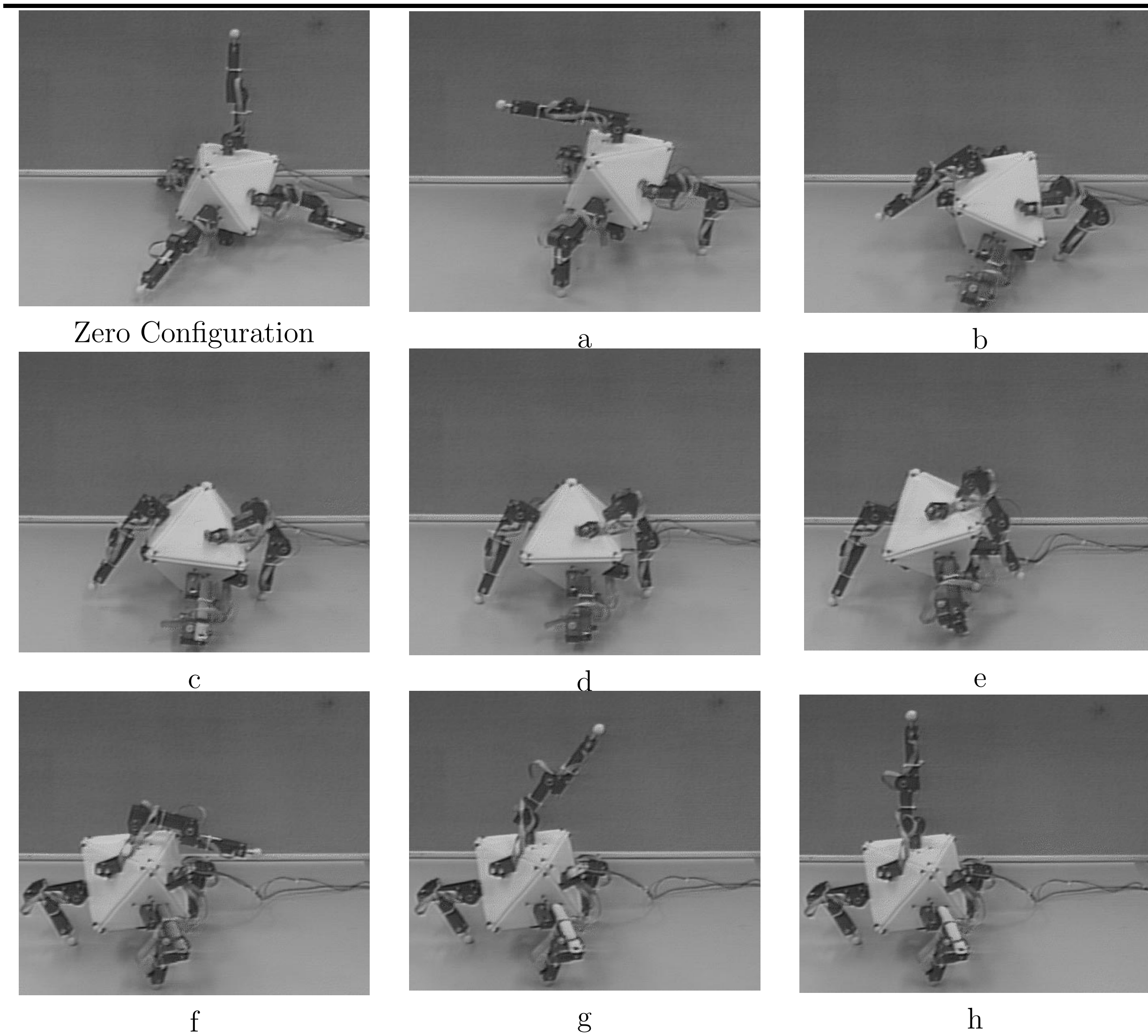

Figure 4: Canonical tumble with 4-beast prototype.

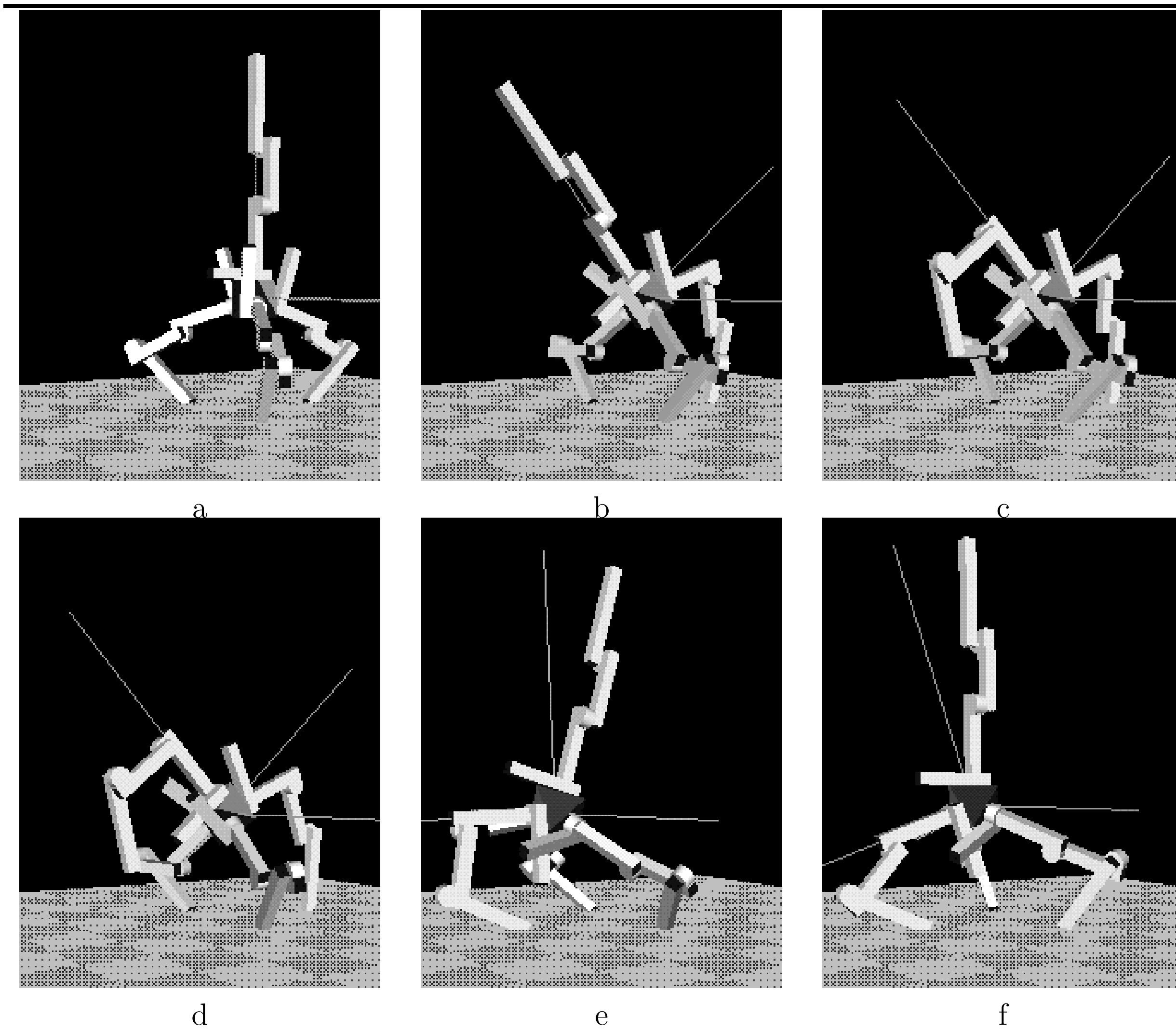

Figure 5: Simulation of a canonical tumble

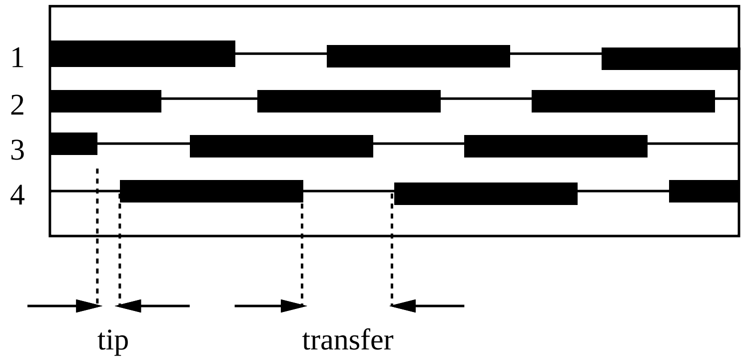

Figure 6: Gait diagram for rolling gait locomotion along a straight path.

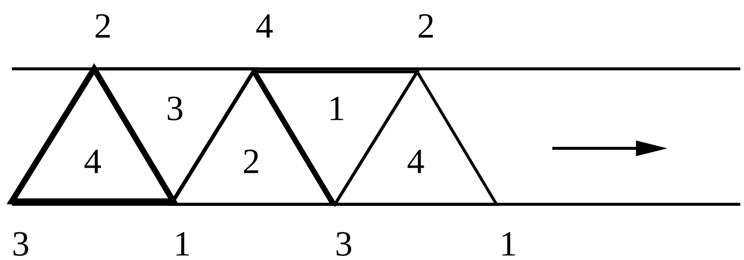

Figure 7: Successive support polygons for rolling gait locomotion along a straight path.

at any time; for beasts with 6 or more limbs, the rolling gait will have more than one limb in the air at the same time. The rolling gait is regularly realizable in the sense of [18].

Figure 7 shows the successive support polygons for the above rolling gait. Each triangle in the figure is a support polygon; the numbers at the vertices indicate the supporting limb and the number at the center of the polygon indicates the limb that is in the transfer phase.

Note that the rolling gait produces a sequence of non-overlapping support triangles, reducing the number of footfalls during locomotion. This is similar to the circulating gait in the Ambler robot [19]. The rolling gait is a type of circulating gait — while the recovery in Ambler is essentially in the horizontal plane[4] and relies on the circular symmetry of the leg stacks, the rolling gait utilizes the spherical symmetry of the robot to recover a limb over the top of robot. Thus the rolling gait has the same general advantages of circulating gaits pointed out by Bares and Whittaker [19], i.e., the number of footfalls are minimized and locomotion requires smaller footholds (or equivalently, can use larger feet) than the

[4]The vertical motion of the feet is only used for ground clearance.

follow-the-leader gaits for rough terrain.

We can denote a tumble step τ as $(l_1 l_2 l_3 l_4)$. By convention, the tumble occurs with the limbs l_1 and l_2 in contact, limb l_3 starts the tumble in contact and ends free, while limb l_4 starts free and ends in contact. Thus the canonical tumble described above is denoted (1234).

Since a tumble does not change the handedness (orientation in the topological sense) of the limb numbering, it is easy to see that there is a one-to-one correspondence between the set of possible tumbles and the set of even permutations of (1234)[5]. This fact simplifies locomotion planning since a single tumble plan can generate all possible tumbles by merely permuting limb numbers.

4 Programming

4.1 Kinematics

The link modules can be configured to form limbs in several ways, and the kinematics of a limb depends on the configuration. In the prototype, each limb has 3 joints, with the Denavit-Hartenberg parameters given in Table 1. The forward and inverse kinematics for positioning the end point of the limb relative to the base of the limb is thus simple. However, complications arise due to the coordination of limbs and the specification the motion of the robot is relative to the terrain; note that the body can be in any orientation with respect to the terrain.

Geometrically, the body of the robot is a regular octahedron. The faces of the octahedron are of two types: limb faces – to which limbs are attached, and control faces on which are mounted the control, computing and communications hardware. The limbs are numbered 1,. . . ,4 as shown in figure 8. Two different assignments which are mirror images of each other are possible. We have picked the "right-handed" assignment.

We use the following notation below. A coordinate frame with index $A \in \mathbb{Z}$, is denoted ${}_A\mathsf{E}$. It consists of a reference point ${}_A\mathsf{E}_o$ called the "origin" of the coordinate frame, and set of orthonormal basis vectors ${}_A\mathsf{E}_x$, ${}_A\mathsf{E}_y$, ${}_A\mathsf{E}_z$. The matrix of coordinates of a vector[6] p in coordinate frame ${}_A\mathsf{E}$ is denoted ${}^A\mathsf{p}$. The homogeneous transformation of coordinate vectors from frame A to frame B is written as ${}^B_A\mathsf{E}$ and is given by the 4×4 matrix

$$ {}^B_A\mathsf{E} = \begin{pmatrix} {}^B_A\mathsf{E}_x & {}^B_A\mathsf{E}_y & {}^B_A\mathsf{E}_z & {}^B_A\mathsf{E}_o \end{pmatrix} $$

where, for instance, a column ${}^B_A\mathsf{E}_x$ is the vector ${}_A\mathsf{E}_x$ expressed in homogeneous coordinates with respect to the frame B.

[5] The total number of possible tumbles is therefore $4!/2 = 12$; this is of course easy to directly enumerate.

[6] Or other contravariant quantity.

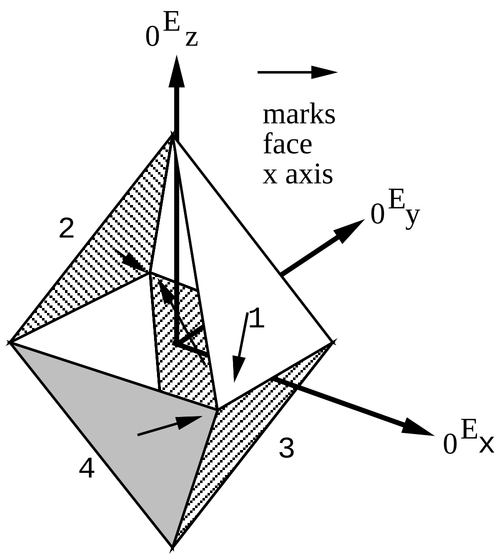

Figure 8: Assignment of body and limb coordinate frames

A reference frame ${}_l\mathsf{E}$ for the base of each limb $l = 1, \ldots, 4$ is attached to a limb face of the platonic beast. Specifically, the origin of the ${}_l\mathsf{E}$ frame is located at the center of the face. The z-axis unit vector is along the outward normal to the face. The orientation of the x-axis is also shown in Figure 8. Note that there is no "natural" assignment of these vectors[7]. The body reference ${}_0\mathsf{E}$ is located at the centroid of the body as shown in figure 8. The location of the limb reference frame ${}_l\mathsf{E}$ relative to ${}_0\mathsf{E}$ is given in table 2.

[7] If there were any symmetric, smooth tangent vector field without zeros on the inscribed sphere, we could pick the base x-axes from this field and obtain a natural assignment – unfortunately, the famous "hairy ball" theorem [20] precludes the existence of such a field.

Limb l	${}^0_l\mathsf{E}$
1	RotY($\cos^{-1}(1/\sqrt{3})$) Trans(0,0,112) RotZ(-π/3)
2	RotY(-$\cos^{-1}(1/\sqrt{3})$) Trans(0,0,112) RotZ(2 π/3)
3	RotX(π + $\cos^{-1}(1/\sqrt{3})$) Trans(0,0,112) RotZ(7 π/6)
4	RotX(π - $\cos^{-1}(1/\sqrt{3})$) Trans(0,0,112) RotZ(π/6)

Table 2: Location of limb frame ${}_l\mathsf{E}$ relative to body frame ${}_0\mathsf{E}$, based on kinematic parameters in Table 1

The body frame ${}_0\mathsf{E}$ is convenient for dealing with the kinematics of limbs but is not convenient for specifying motions of the robot, especially for locomotion. To simplify programming, we define some new reference frames that depend on which feet are in contact with the ground as well as the intended direction of motion.

As discussed in section 3, we can specify the intended direction of motion by specifying an even permutation of the limb numbers (1234). Assume, without loss of generality, that the current permutation of the limbs is (1234) and by convention limbs 1, 2 and 3 are on the ground, while limb 4 is possibly free. Let us denote the location of the foot of limb i as $\mathsf{p}(i)$.

We define the *ground frame*, ${}_g\mathsf{E}$, as a frame fixed with respect to the ground with origin at $\mathsf{p}(1)$, the location of the foot of limb 1. The y-axis of the frame points towards foot 2; specifically, it is

$$ {}_g\mathsf{E}_y \stackrel{def}{=} \frac{(\mathsf{p}(2) - \mathsf{p}(1))}{|\mathsf{p}(2) - \mathsf{p}(1)|}. $$

The z-axis is orthogonal to the plane defined by the feet 1, 2 and 3; it is taken to be the unit vector along ${}_g\mathsf{E}_y \times (\mathsf{p}(3) - \mathsf{p}(1))$. This determines the ground frame and the transformation ${}^0_g\mathsf{E}$ completely.

For specifying motions relative to the body, it is often convenient to define a second frame called the *heading frame*, ${}_h\mathsf{E}$, parallel to the ground frame but located at the centroid of the body at the start of the specified motion. The origin of ${}_h\mathsf{E}$ coincides with that of ${}_0\mathsf{E}$. We also define the *instantaneous heading frame* at a time t, ${}_{h'}\mathsf{E}$, parallel to the heading frame, but located at the centroid of the body at time t during the motion.

A tumble step can be achieved by translation of the body along the x-axis of the heading frame while rotating the body about an axis through the centroid, parallel to the y-axis ${}_h\mathsf{E}_y$. During the motion, the location of the feet will remain constant relative to the ground frame.

For instance, to translate the body a distance d while rotating the body by an angle ϕ, we can define the motions

$$ \mathsf{Ry} = \begin{pmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathsf{Tx} = \begin{pmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

We can compute the joint angles at the end of the motion as follows. The foot position of limb l relative to base frame ${}_l\mathsf{E}$ (denoted ${}^l\mathsf{p}(l)$) is easily computed from the forward kinematics of the three link chain. The foot position relative to the ground frame is then given by

$$ {}^g\mathsf{p}(l) = {}^g_0\mathsf{E}\ {}^0_l\mathsf{E}\ {}^l\mathsf{p}(l), \tag{1} $$

where ${}^g_0\mathsf{E} = ({}^0_g\mathsf{E})^{-1}$. The new position of the body frame after the motion can be computed relative to the ground frame as

$$ {}^g_{0'}\mathsf{E} = {}^g_h\mathsf{E}\ \mathsf{Tx}\ \mathsf{Ry}\ {}^h_0\mathsf{E}. \tag{2} $$

Therefore, inverting and expanding, we have

$$ {}^{0'}_{g}\mathsf{E} = {}^{0}_{g}\mathsf{E}\ {}^{g}_{h}\mathsf{E}\ \mathsf{Ry}^{-1}\ \mathsf{Tx}^{-1}\ {}^{h}_{g}\mathsf{E}. \tag{3} $$

Hence the new location of the foot of limb l relative to its limb reference frame is given by

$$ {}^{l}\mathsf{p}(l)' = {}^{l}_{0}\mathsf{E}\ {}^{0'}_{g}\mathsf{E}\ {}^{g}\mathsf{p}(l). \tag{4} $$

From this the new joint angles can be obtained from the inverse kinematics of the limb (see, e.g., [21]).

4.2 Trajectory Interpolation

For large displacements of the body we interpolate the motion of the robot to achieve a uniform motion of the body as follows. We will focus on the limbs in contact with the ground; the rest are treated using standard trajectory interpolation techniques [21]. The total motion of the body relative to the heading coordinate frame is given by

$$ {}^{h}_{h}\mathsf{M} = {}^{h}_{g}\mathsf{E}\ {}^{g}_{0'}\mathsf{E}\ {}^{0}_{g}\mathsf{E}\ {}^{g}_{h}\mathsf{E} = \mathsf{Tx}\ \mathsf{Ry}. \tag{5} $$

We will allow Tx and Ry to be general translations and rotations, respectively. The twist ${}^{h}\mu$, corresponding to ${}^{h}_{h}\mathsf{M}$ can then be computed (see, e.g, [22]). We use the following notation [23]: if

$$ {}^{h}\mu = \begin{pmatrix} w \\ v \end{pmatrix}, $$

then

$$ [{}^{h}\mu] \stackrel{def}{=} \begin{pmatrix} 0 & -w_z & w_y & v_x \\ w_z & 0 & -w_x & v_y \\ -w_y & w_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{pmatrix}, $$

and by definition

$$ {}^{h}_{h}\mathsf{M} = \exp\,[{}^{h}\mu]. $$

One possible interpolation strategy is to move the body with a constant spatial velocity $\frac{1}{T}({}^{h}\mu)$ where T is the desired duration of motion. While this is uniform motion in $SE(3)$, the centroid of the body need not move in a straight line and is therefore difficult for programmers to visualize.

Instead we will move the body with constant linear velocity relative to the heading frame ${}_{h}\mathsf{E}$ with uniform rotation about the *instantaneous* heading frame ${}_{h'}\mathsf{E}$ located at the instantaneous centroid of the body. Specifically, define ${}^{h}\xi$ and ${}^{h}\eta$ from

$$ \exp([{}^{h}\xi]T) = \mathsf{Tx}, \exp([{}^{h}\eta]T) = \mathsf{Ry}. \tag{6} $$

Clearly, these are of the form

$$ {}^{h}\eta = \begin{pmatrix} \omega \\ 0 \end{pmatrix}, {}^{h}\xi = \begin{pmatrix} 0 \\ \nu \end{pmatrix}. \tag{7} $$

where ω is the angular velocity and ν is the translational velocity. We define the motion at time t from the start of the motion segment as $\mathsf{Ry}' = \mathsf{Rot}(\omega t)$, $\mathsf{Tx}' = \mathsf{Trans}(\nu t)$

From equation 4,

$$ \begin{aligned} {}^{l}\mathsf{p}(l)' &= {}^{l}_{0}\mathsf{E}\ {}^{0'}_{g}\mathsf{E}\ {}^{g}\mathsf{p}(l) & (8) \\ &= {}^{l}_{0}\mathsf{E}\ {}^{0}_{g}\mathsf{E}\ {}^{g}_{h}\mathsf{E}\ \mathsf{Ry}'^{-1}\ \mathsf{Tx}'^{-1}\ {}^{h}_{g}\mathsf{E}\ {}^{g}\mathsf{p}(l) & (9) \\ &= {}^{l}_{h}\mathsf{E}\ \mathsf{Ry}'^{-1}\ \mathsf{Tx}'^{-1}\ {}^{h}\mathsf{p}(l). & (10) \end{aligned} $$

Therefore,

$$ \begin{aligned} \frac{d}{dt}{}^{l}\mathsf{p}(l)' &= {}^{l}_{h}\mathsf{E}\ \left(\mathsf{Ry}'^{-1}\ [-{}^{h}\eta]\ \mathsf{Tx}'^{-1} + \mathsf{Ry}'^{-1}\ [-{}^{h}\xi]\ \mathsf{Tx}'^{-1}\right)\ {}^{h}\mathsf{p}(l) & (11) \\ &= {}^{l}_{h}\mathsf{E}\ \mathsf{Ry}'^{-1}\ (-[{}^{h}\eta + {}^{h}\xi])\ \mathsf{Tx}'^{-1}\ {}^{h}\mathsf{p}(l), & (12) \end{aligned} $$

where

$$ [{}^{h}\eta + {}^{h}\xi] = \begin{pmatrix} 0 & -\omega_z & \omega_y & \nu_x \\ \omega_z & 0 & -\omega_x & \nu_y \\ -\omega_y & \omega_x & 0 & \nu_z \\ 0 & 0 & 0 & 0 \end{pmatrix}. $$

In equation 11 we have used the fact that $\frac{d}{dt}\exp([\alpha]t) = [\alpha]\exp([\alpha]t) = \exp([\alpha]t)[\alpha]$.

Note that ω, ν, ${}^{l}_{h}\mathsf{E} = {}^{l}_{0}\mathsf{E}\ {}^{0}_{g}\mathsf{E}\ {}^{g}_{h}\mathsf{E}$, and ${}^{h}\mathsf{p}(l) = {}^{h}_{l}\mathsf{E}\ {}^{l}\mathsf{p}(l)$ can all be computed at the beginning of a motion segment. Hence, during the motion, we can simultaneously compute both the position and velocity of the foot relative to the limb frame as

$$ \begin{aligned} \Gamma &= {}^{l}_{h}\mathsf{E}\ \mathsf{Ry}'^{-1}, & (13) \\ P &= \mathsf{Tx}'^{-1}\ {}^{h}\mathsf{p}(l), & (14) \\ {}^{l}\mathsf{p}(l)' &= \Gamma P, & (15) \\ \frac{d}{dt}{}^{l}\mathsf{p}(l)' &= \Gamma(-[{}^{h}\eta + {}^{h}\xi])P. & (16) \end{aligned} $$

A piecewise cubic joint trajectory which produces this body trajectory approximately is then computed. At knot points[8] we can compute the joint angles θ_l as corresponding to ${}^{l}\mathsf{p}(l)'$ from the inverse kinematics of the limb. The joint velocities $\dot{\theta}_l$ are computed using the limb Jacobian matrix J_l, by solving the linear system $J_l\dot{\theta}_l = \frac{d}{dt}{}^{l}\mathsf{p}(l)'$. The interpolating polynomials can then be computed (see, e.g., [21]).

[8] We currently use constant knot spacing; it is not difficult to modify this to adaptive knot spacing schemes.

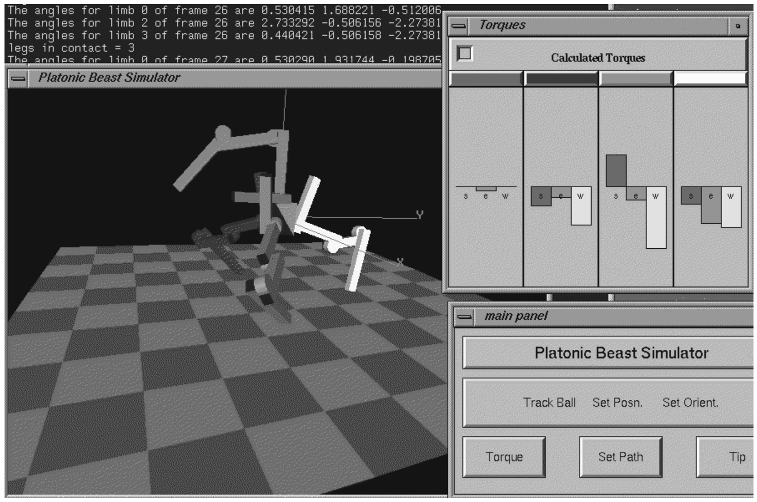

Figure 9: Simulator display

4.3 Simulation

The platonic beast simulator is an interactive graphics program running on a Silicon Graphics Crimson/VGX workstation, and is designed to allow the user to explore design decisions, by varying link lengths, body size, and masses of the components, as well as simulating sequences of motions of a specified gait.

The position and orientation of the body, the position of the feet on the ground, and the configuration of the free limbs can all be changed using the mouse. The user may also specify which the contact conditions of the feet. The inverse kinematics for the contact limbs are re-computed for the new configuration, and the beast is redisplayed in real-time.

The user may specify a path by giving a set of via-points through which the beast is to follow. Each via-point consists of the position and orientation of the body, as well as the position of each limb – the latter can be specified either in joint space or in the space of foot positions. The beast is then animated by interpolating the via points in joint space. Tipping, if present during the path, is also animated. Figure 5 shows a tumble motion generated by the simulator. Figure 9 shows a closeup of the simulator display.

The torques at the joints of the robot due to static forces are computed and can be displayed during motion. The program also detects tipping and signals the user by changing the color of the body. At present the contact friction is not taken into account, but we plan to do so in the near future. Note that since the robot moves relatively slowly and is statically stable at all times except while tipping, the static model is adequate. The computed torques can then be used in the design process to guide the choice of motors, as well as investigating the feasible configuration space of the beast.

Despite the current simplicity of its dynamic models, the simulator is an extremely useful tool for evaluating designs and for programming the robot. The ability to animate robot programs allows the user to verify the correctness of a given robot program before executing it on the actual hardware.

5 Discussion

A key benefit of the spherical symmetry of limb placement is robustness with respect to toppling. This is particularly important for locomotion on rough terrain where it is difficult to measure terrain orientation, friction and integrity. On such terrain, it is not possible to guarantee toppling avoidance. Even if there is no physical damage to the robot after toppling, most legged robots may not be able to recover since the limb placement is specialized for operation in a small range of body orientations and the robot can land on its "back". The platonic beast design, on the other hand, has no direction specialized as the "up" direction, as can be seen from the rolling gait. A statically stable foot placement is available in all orientations of the body in three dimensions, allowing the robot to recover from a topple. We are not aware of any other robot with this ability.

Another consequence of spherical symmetry is ease of changing directions. This is actually a feature of having at least circular symmetry in the horizontal plane. This feature is shared by the ODEX 1 robot [24], the original design of the Ambler robot [25] and to a lesser extent by the prototype [19] which breaks the symmetry by separating the legs into two stacks. Bilaterally symmetric walking machines such as the ASV [4] have to either perform a turning maneuver to change the heading of the robot or perform a crab gait.

Figures 10 and 11 show the gait of a 4-beast turning a sharp corner. The robot can use this gait to pass an obstacle such as a wall. Limb 1 is held in contact while the robot rolls about it, effectively synthesizing a cone.

In this paper we have concentrated mainly on the locomotion task in a 4-beast, but the uniform design of the limbs and the relatively large number of degrees of freedom permit more flexibile limb utilization. The multi-purpose limbs can be recruited to solve different tasks based on need. A limb could serve for a period as a "leg" and later switch to a "finger." Figure 12 depicts possible uses of this flexibility. A 6-beast could use four of its limbs for locomotion and 2 limbs as a hand to carry or manipulate simple objects. To manipulate more complex objects, 3 or more limbs could be used for grasping with fewer limbs available for locomotion. The figure also shows how a 6-beast could be used as a 3-fingered mobile hand. The robot first locomotes using a rolling gait to the desired location. Then with three limbs in point contact with the ground, the body can have any rigid motion in a small range[9]. Multiple objects could also be manipulated, e.g., 4 limbs can be used as 2 "hands"

[9]This can be seen qualitatively from the fact that each limb has three revolute joints, so that the entire system has $6 + 3 \times 3 = 15$ dof; each point contact reduces 3 dof leaving 6 dof for the body. A study of the range of motion possible for the body using our simulator indicates that the range is acceptably large.

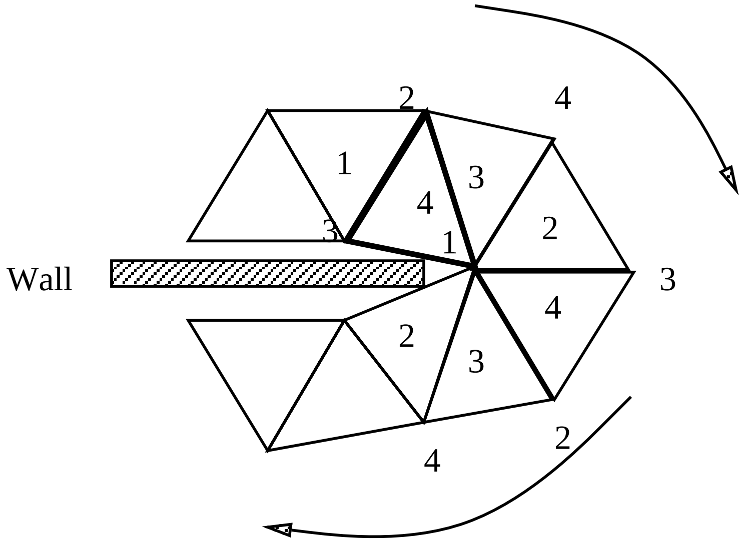

Figure 10: Successive support polygons for sharp turn.

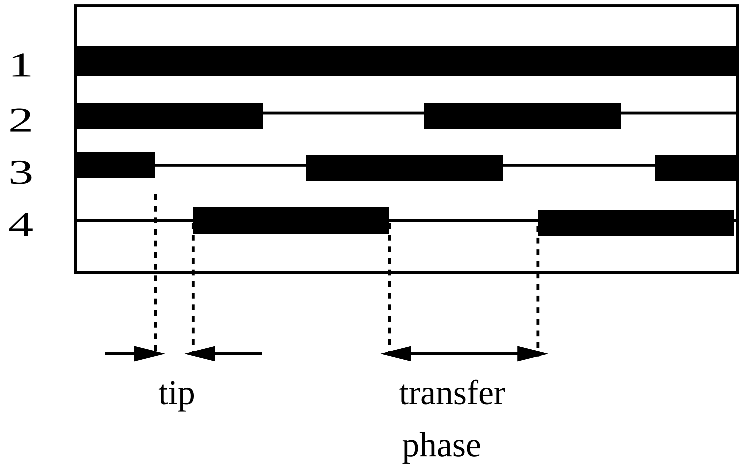

Figure 11: Gait diagram for sharp turn.

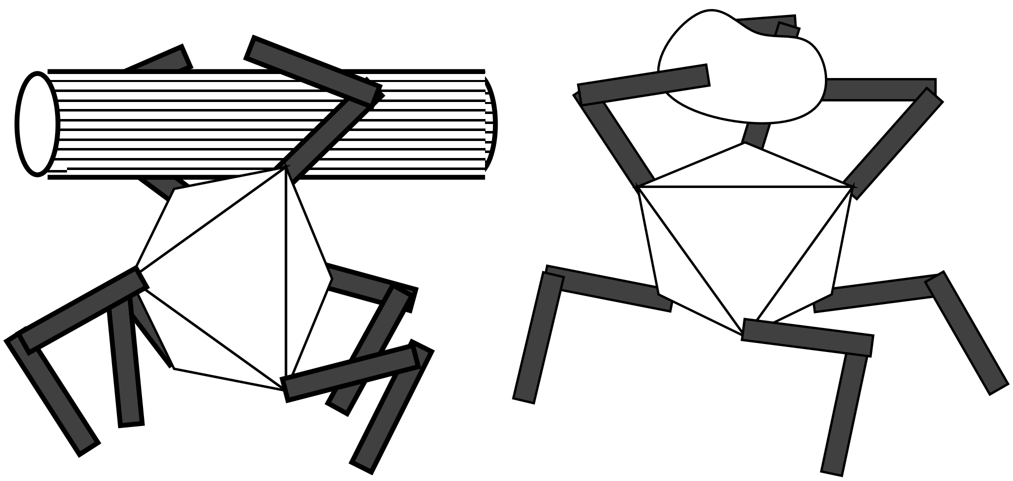

Figure 12: A 6-beast can be used as a 4-limbed crawling machine with a 2 fingered hand or as a 3-fingered mobile hand.

with 2 fingers each.

The flexible use of limbs provides a certain degree of tolerance to loss or catastrophic failure of a limb. The uniformity of the limbs means that any limb can be used as a leg. For example, if a 6-beast loses operation of a limb while rolling, it can simply reorient the body so that the defective limb is on top and no longer used for locomotion, and walk with the slower quadruped crawl gait.

The spherical symmetry of the platonic beasts and rolling gait can be disadvantageous in some applications. During the rolling gait, the body does not maintain constant attitude – hence it is difficult to carry a payload that is sensitive to orientation (e.g., human passengers) without adding gimbals or other stabilizing devices. Since the limbs are intended to perform a variety of functions, they are not optimized for energy efficiency during locomotion or for simplicity of kinematics and motion planning. The simplicity of the kinematics will decrease in significance over time given the constant improvement in performance and price of computational resources. A tether or an internal combustion engine may be necessary to supply power to operate a platonic beast for extended periods. Energy efficiency is important for legged locomotion in many appications such as exploring remote areas [7, 25] and for walking robots for the consumer market. However, we believe there is an important niche of applications for which the robustness to toppling and fault tolerance are more critical factors.

Finally, in addition to the possible utility of the robot design, it is interesting to study locomotion in these and other robots such as [25] which differ significantly from animal locomotors. They focus our attention on the essential requirements of legged locomotion rather than on the emulation and implementation of locomotion algorithms perceived to be used in nature.

6 Conclusions

We have described the concept a new family of multi-limbed robots called Platonic Beasts and outlined the design and construction of an experimental protype. This family has a number of interesting features, including the spherical symmetry of limb placement, uniform modular multipurpose limbs, and a large number of degrees of freedom. These features provide the following advantages: novel gait — in addition to typical quadruped creeping gaits, a new *rolling* gait is possible; robustness to toppling and fault tolerance; flexible utilization of limbs; and modular distributed control. We believe this class of robots is well suited for some application areas, particularly locomotion in rough and uncertain terrain, such as the outdoor environments found in exploration tasks and the indoor environments found in fire-fighting and security tasks. For future work, we plan to develop high level constraint-based programming tools for these complex robots [26], and to incorporate additional sensing, including low-resolution vision sensors, making the robot more reactive and terrain-adaptive. We also plan to develop a 6-beast and explore locomotion and manipulation by these robots.

References

[1] J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. Whittaker. Ambler, an autonomous robot for planetary exploration. *IEEE Computer*, 22(6):18–26, June 1989.

[2] John E. Bares and W. L. Whittaker. Walking robot with a circulating gait. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 809–818, 1990.

[3] R. W. Brockett. Robotic manipulators and the product of exponentials formula. *Mathematical Theory of Networks and Systems*, pages 120–129, 1984.

[4] R. A. Brooks. A robot that walks; emergent behaviors from a carefully evolved network. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 692–694, 1989.

[5] I.M. Chen and J. W. Burdick. Enumerating the non-isomorphic assembly configurations of modular robotic systems. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 1985–1992, 1993.

[6] I. J. Cox and G. T. Wilfong. *(ed.s) Autonomous Robot Vehicles.* Springer Verlag, 1990.

[7] J. J. Craig. *Introduction to robotics : mechanics and control.* Addison-Wesley, 1989.

[8] Marc D. Donner. *Real-time Control of Walking.* Birkhauser, 1987.

[9] M. H. Raibert (ed.). Special issue on robot locomotion. *International Journal of Robotics Research*, 3(2), 1984.

[10] Cynthia Ferrell. Robust agent control of an autonomous robot with many sensors and actuators. M.S. Thesis, MIT, May 1993.

[11] S. Hirose. A study of design and control of a quadruped walking vehicle. *International Journal of Robotics Research*, 3(2):113–133, 1984.

[12] M. W. Hirsch. *Differential Topology*. Springer-Verlag, 1976.

[13] R. Hui, N. Kircanski, A. Goldenberg, C. Zhou, P. Kuzan, J. Wiercienski, D. Gershon, and P. Sinha. Design of the iris facility - a modular, reconfigurable and expandable robot test bed. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 155–160, 1993.

[14] Dante II. NASA/CMU mission to mt. spurr. URL http://maas-neotek.arc.nasa.gov/dante/.

[15] R. B. McGhee. Some finite state aspects of legged locomotion. *Mathematical Biosciences*, 2:67–84, 1968.

[16] R. B. McGhee and A. A. Frank. On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, 3:331–351, 1968.

[17] R. Murray, Z. Li, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC Press, 1994.

[18] Dinesh K. Pai. Least constraint: A framework for the control of complex mechanical systems. In *Proceedings of the American Control Conference*, pages 1615 – 1621, 1991.

[19] Richard P. Paul. *Robot Manipulators: Mathematics, Programming, and Control*. The MIT Press, 1981.

[20] Marc H. Raibert. *Legged Robots That Balance*. The MIT Press, 1986.

[21] Marc H. Raibert and J. K. Hodgins. *Biological Neural Networks*, chapter Legged Robots, pages 267–286. Academic Press, 1993.

[22] M. Russell. Odex 1: The first functionoid. *Robotics Age*, 5(5):12–18, 1983.

[23] D. Schmitz, P. Khosla, and T. Kanade. The CMU reconfigurable modular manipulator system. CMU-RI-TR-88-7, The Robotics Institute, Carnegie Mellon University, 1988.

[24] Shin-Min Song and Kenneth J. Waldron. *Machines That Walk: The Adaptive Suspension Vehicle*. The MIT Press, 1989.

[25] D. J. Todd. *Walking Machines: An Introduction to Legged Robots*. Kogan Page, 1985.

[26] H. J. Weidemann, J. Eltze, and F. Pfeiffer. Leg design based on biological principles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 352–358, 1993.