Dynamic Rolling for a Modular Loop Robot

Jimmy Sastra Sachin Chitta Mark Yim
GRASP Laboratory,
University of Pennsylvania,
Philadelphia, Pennsylvania, USA
E-mail: {jsastra,sachinc,yim}@grasp.upenn.edu

Abstract

Reconfigurable modular robots have the ability to use different gaits and configurations to perform various tasks. A rolling gait is the fastest currently implemented gait available for traversal over level ground and shows dramatic improvements in efficiency. In this work, we analyze and implement a sensor-based feedback controller to achieve dynamic rolling for a loop robot. The robot senses its position relative to the ground and changes its shape as it rolls. This shape is such that its center of gravity is maintained to be in front of its contact point with the ground, so in effect the robot is continuously falling and thus accelerates forward. Using simulation and experimental results, we show how the desired shape can be varied to achieve higher terminal velocities. The highest velocity achieved in this work is 26 module lengths per second (1.6m/s) which is believed to be the fastest gait yet implemented for an untethered modular robot. One of the major findings is that more elongated shapes achieve higher terminal velocities than rounder shapes. We demonstrate that this trend holds going up as well as down inclines. We show that rounder shapes have lower specific resistance and are thus more energy efficient. The control scheme is scalable to an arbitrary number of modules, shown here using 8 to 14 modules.

I. Introduction

Locomotion is one of the most basic functions of mobile robots. Robotics researchers have demonstrated a wide variety of locomotion modes, including legged, wheeled, snake-like and even amoeba-like locomotion. Not every locomotion mode is suitable for all tasks. For example, a car like vehicle may be well suited for travelling over roads, but would not be suitable for climbing through a rubble pile doing search and rescue. A snake-like robot maybe well suited for climbing through tightly constrained environments, but is probably very inefficient going long distances.

Reconfigurable modular robots have been used as a platform to study different locomotion modes [Zhang et al., 2002] and indeed, hundreds of locomotion modes have been demonstrated [Butler et al., 2002], [Castano et al., 2000], [Chirikjian, 1996], [Kamimura et al., 2001], [Yim, 1994], [Yim et al., 2004], [Yim et al., 2006], [Yim et al., 2001]. The choice of configuration is usually task specific. One of the advantages of self-reconfigurable robots is the ability to reconfigure as the needs and environments change. For example, a self-reconfigurable robot in a search and rescue application might approach a scene using a fast and efficient rolling gait, reconfigure into a snake to squeeze through tight spaces and then reconfigure into a form with many limbs to move rubble or protect a victim. This paper will focus on a rolling loop configuration using a modular reconfigurable robot called CKBot [Park et al., 2006].

Multiple robotic systems have demonstrated the ability to perform specific tasks robustly. RHex [Weingarten et al., 2004], a legged robotics system, can traverse a remarkable variety of rough terrain. Tracked robots [Yamauchi, 2004] can climb stairs and small rocks very easily. Legged systems like Littledog [Chitta et al., 2007] and BigDog [Playter et al., 2006] have also demonstrated similar capability to traverse rough terrain. Wheeled vehicles, including autonomous cars [Buehler et al., 2005], are now capable of traveling large distances at high speeds. However, none of these systems can perform the variety of tasks that a single modular robot can. Modular robots, by reconfiguring either manually or autonomously, can slither through narrow passages, act as a manipulator, walk over rough terrain and climb up certain surfaces. This ability for a single robot to perform a set of tasks makes modular robots particularly applicable for tasks like search and rescue and planetary exploration where space and power requirements dictate the need for a versatile system.

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

In the loop configuration shown in Figure 1(a), the motion of the robot is like that of a tank tread. On flat terrain this gait is currently considered to be the most efficient as well as the fastest configuration under some conditions [Yim, 1994] (this has not been proven for the general case though). It has been implemented on various robots in [Kamimura et al., 2001], [Sugiyama et al., 2005], [Yim, 1994]. In most implementations, the motion of these gaits was not dynamic, *i.e.* there was no inertial component to the motion. Rather the motions were purely *kinematic*; changes in geometry cause locomotion, stopping the changes in geometry also stop locomotion. There is a limit to the rate at which this motion can accelerate and stay

kinematic. Accelerating too fast causes the loop roll backwards as shown in Figure 1(c). Its motion stops being kinematic and causes the robot to undulate in space. This however does not produce the intended behavior of forward locomotion. In this work we aim to produce a controlled dynamic rolling mode as shown in Figure 1(b). These different modes of locomotion are also illustrated in Extension 1.

In [Kamimura et al., 2005], Kamimura et al. implemented an *open-loop* dynamic rolling gait using CPGs (Central Pattern Generator) where the weights for the CPGs were determined using simulation. A deformable robot was actuated by SMA (Shape Memory Alloy) coils in [Sugiyama et al., 2005] to manipulate the shape into stable and unstable deformations for crawling and jumping. In [Matsuda and Murata, 2006], Matsuda and Murata proposed a robot whose links formed a closed chain where the actuators control the stiffness of a spring in each joints. This allows them to adjust the stiffness in each joint and drive the robot forward. In [Shen et al., 2006], a dynamic simulator was used to generate and simulate a dynamic rolling gait. Feedback was through accelerometers in the robot and an average velocity of about 1 m/s was reported. However, this gait was not implemented on an actual robot and no analytical insight was provided. A dynamic rolling gait was implemented on a Icosahedral modular robot called the Tetrobot in simulation in [Lee and Sanderson, 2000]. However, no implementation details were given for this work.

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

Different kinds of rolling robots have also been built and studied in recent years. Halme et.al. [Halme et al., 1996] introduced the first truly spherical robot that uses an internal steerable wheel to generate motion. Mukherjee, et.al. [Mukherjee et al., 1999] proposed motion planning strategies for the Spherobot, a proposed variant on Halme's robot that used internal reciprocating weights. Bhattacharya et.al. [Bhattacharya and Agrawal, 2000] used the principle of conservation of angular momentum to generate motion for a spherical robot. They also presented analytical results for control of the robot using an optimal control approach. Shores, et.al. [Shores and Minor, 2005] propose a bipedal robot that can also locomote by folding into a circular shape and using internal joint motions to initiate rolling through small shifts of the center of gravity of the robot. The Hexaball project [Phipps and Minor, 2005] proposes to build a spherical robot with legs that can climb over obstacles and roll by using small motions of its legs.

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

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

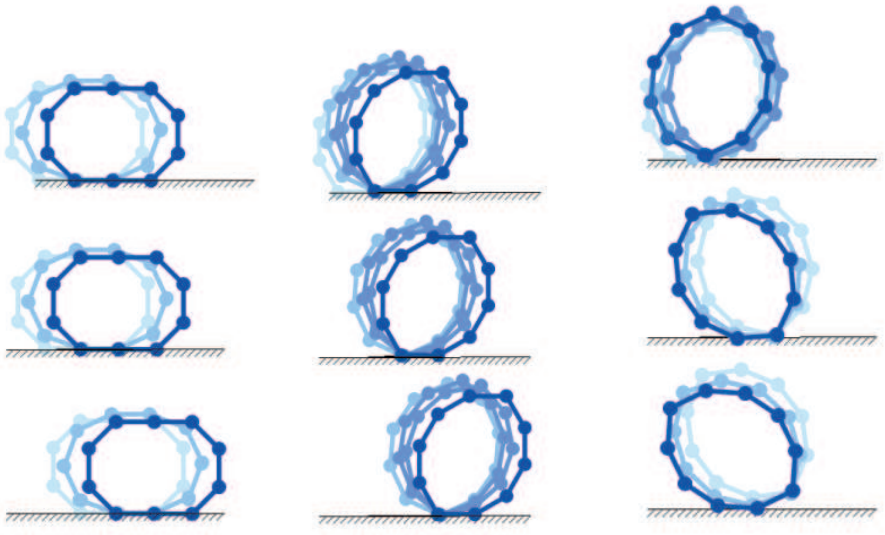
The resulting relatively simple controller gives us better insights into the dynamics of the system. Another benefit of this implementation is that the method scales to any number of modules or joints in a loop, within actuator limitations. In addition to simplifying the control algorithm, our approach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

This paper is structured as follows. In Section II, we present the robot used in this work. In Section III, we introduce the main idea behind *dynamic rolling* and compare it with *kinematic rolling*. In Section IV we present a four-bar like model for the robot that simplifies the analysis of dynamic rolling. In Section V, we propose the framework used for control. The experimental setup is described in Section VI. In Section VII, we present theoretical results derived using this model and experimental results with 10 to 14 module rolling module loops. In Sections VIII and IX, we follow up with a discussion on insight gained from the results and possible future applications.

II. The robot

The robot system used in this work is a modular system called CKBot (Connector Kinetic roBot) and is shown in Figure 2. An individual module is shown in Figure 3. Each module is made up of a hobby servo that drives a rotary degree of freedom, a frame made of acrylonitrile-butadiene-styrene (ABS) plastic, a microprocessor and a touch sensor. A summary of the hardware is shown in Table I.

The one degree of freedom has a range of +90 to -90 degrees. When at 0 degrees, the module closely resembles a cube 60 mm on a side. As the degree of freedom moves away from zero, the rounded edge of the frame is exposed as in Figure 3. As a loop rolls, this rounded edge will make impact the ground. This rounded edge smooths rolling to some extent, though the top face of the module still poses a corner that impacts the ground.



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

Fig. 1. Different modes of rolling.

Each module also has eight identical electrical connection ports, seven around four faces of the robot and one internal. These ports are used to electrically connect modules together as well as add extra computation, sensors or batteries. Power and communications are passed from module to module. Communication to each module is through a global bus based on the RoboticsBus protocol [Gomez-Ibanez et al., 2004] which is built on the CANbus standard (Controller Area Network).

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

To form a loop, each module is attached end-to-end using screws and the two ends are then screwed together to form a loop. It is possible to form loops in other ways. For example, instead of daisy-chaining head to tail each module: (head-tail)(head-tail)(head-tail) the modules could be attached head-head: (head-tail)(tail-head)(head-tail). This is the configuration that Superbot [Castano et al., 2000] and MTRAN [Kamimura et al., 2001] use. When tested with CKBOT, this configuration does not do as well in taking advantage of the rounded structure of the modules and thus the motion is not as smooth. As a result this was likely to be less efficient and was not tested extensively.

A separate microcontroller board, the **brain**, serves as a centralized controller. It plugs into one of the ports on the robot and provides control position commands for all modules. The touch sensors are infra-red proximity sensors that measure reflectance as an indication of distance to the ground. These touch sensors plug inside the module as shown in Figure 3. Sensors use empirically derived thresholds for different surfaces to determine whether the module they are plugged into is touching down or not. The touch sensors send process messages to the brain upon a touchdown event. The brain then calculates the angles required for each module to track the desired shape and sends these commands to the microcontroller on each module.

III. KINEMATIC VS. DYNAMIC ROLLING

Statically stable locomotion is a term that is often used to characterize robot gaits. At any moment in a statically stable gait, the robot could stop moving its joints and the robot would not fall over. The projection of the center of gravity is always maintained to be within the convex hull of the ground contact points. Dynamic locomotion characterizes robot gaits in which the inertia of the robot plays an important role in the locomotion. In general, gaits (which are assumed to be stable) are either statically stable or dynamically stable, but not both. Traditionally static and dynamic stability refers to legged robot gaits. When applied to rolling gaits things become less clear. An automobile has its center of gravity always within the convex hull of its four tires. If it moves slow enough, the inertia of the vehicle can be ignored and it might be said that the vehicle is

TABLE I

TECHNICAL SPECIFICATIONS FOR A CKBOT MODULE.

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

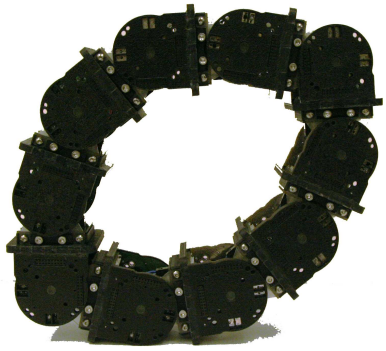


Fig. 2. Ten CKBOT modules forming a football shape.

statically stable. However, if it gained any significant speed, the inertia cannot be ignored and the vehicle might be said to be dynamically stable. The line delineating the two conditions is not clear. In the case of loop robots, we refer to the gaits in which inertia plays no role as kinematic rolling. Here, the equations of motion can be determined directly from the geometry (no mass terms).

A. Kinematic Rolling

A kinematic rolling gait is implemented by repeatedly moving the shape of the loop such that the long axis rotates. This motion is similar to the motion of a tank tread. One rotation of the long axis corresponds to one cycle of the gait. The frequency of rotation is directly proportional to the speed, *i.e.* stopping the tread causes the whole robot to stop. For a closed loop robot like the one used in this work, one typical loop shape has two lines of modules one on top of the other attached by an intermediate set of modules forming arcs as shown in Figure 1(a). A kinematic roll for this configuration is executed by smoothly interpolating the joint angle of each module to the joint angle of the neighboring one in the loop. This type of motion can be easily represented using a *gait table* [Yim, 1994].

An example gait table for a kinematic rolling gait for a 10 module loop robot is shown in Table II. The neighboring columns of the table correspond to neighboring modules in the loop. The rows of the table correspond to steps (or time). The elements

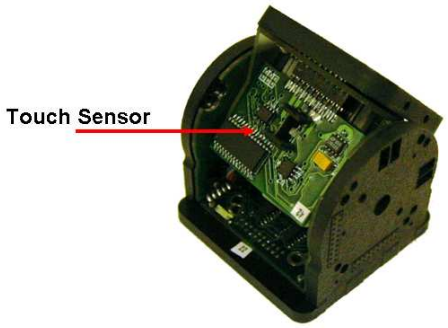


Fig. 3. An individual CKBOT module.

TABLE II

GAIT TABLE FOR KINEMATIC GAIT (ALL ANGLES ARE SPECIFIED IN DEGREES).

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

of the table are the joint angles for the corresponding module at the corresponding time. Note that there are only five rows in the gait table since the gait cycles back to the first row after the fifth.

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

B. Dynamic Rolling

Unlike the kinematic gait a dynamic gait continues to move the robot even after all joints have stopped moving, *i.e.* a dynamic gait utilizes momentum. To create a dynamic rolling motion for a modular loop robot, one approach is to move the center of mass beyond the pivot point for the module currently on the ground as shown in Figure 1(b) and Figure 5. This results in a moment contribution from the weight of the robot in the direction of rolling and the robot accelerates in that direction.

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

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

One way to view this method of control is that of a modified gait control table where the speed of motion between rows of a gait table is based on sensor feedback.

IV. ANALYTICAL MODEL

We make a simplifying assumption that serves to reduce the complexity of the dynamic analysis for the robot. We choose a "backbone" curve [Chirikjian and Burdick, 1994] to which we map the modules. We restrict this backbone curve to a shape that is formed by joining two equal arcs of a circle whose sector subtends an angle less than 180 degrees. This results in a shape which resembles an American football as shown in Figure 2. In the limit, as the two arcs approach 180, the shape reduces to a circle. The modules of the robot approximate this backbone curve by fitting the position of the joints to lie on the arcs.

The shape can be defined using a single parameter θ_a, the angle between the modules at the top and bottom apex of the shape (Figure 4). All the other joint angles are equal to each other (to say θ_s) and can be derived in terms of θ_a from Equation 1

$$2\theta_a + (n-2)\theta_s = 2\pi, \tag{1}$$

where n is the number of modules in the loop.

It will be clear from our choice of control strategy in Section V that local shape changes of the robot will involve only four modules moving at a time just as the kinematic gaits in Section III-A. We can thus simplify the model of the loop to that of a floating four-bar mechanism hinged at the contact point during the entire motion of one touchdown to the next touchdown. In this model the four moving modules represent the four joints. The two longer arcs (nodes 2 through 5 and nodes 7 through 10) represent two of the links of the four-bar while the other two links (comprising node 1 and 6) are made up of single modules. Reducing the model in this manner to the one degree of freedom four bar linkage means that the shape of the robot can be parameterized using a single parameter, θ_a (or similarly θ_s). This framework is shown in Figure 4.

The equations of motion for this simplified version of the robot are derived using a standard method by first defining the Lagrangian for the system and deriving Lagrange's equations. The generalized coordinates used in the analysis are the apex

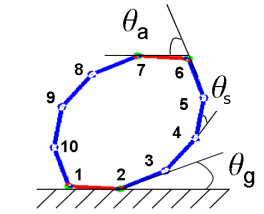


Fig. 4. Four-bar model used for analysis. The two longer arcs (nodes 2 through 5 and nodes 7 through 10) freeze their joints and can be considered rigid. The thus represent two of the links of the four-bar while the other two links (comprising node 1 and 6) are made up of single modules.

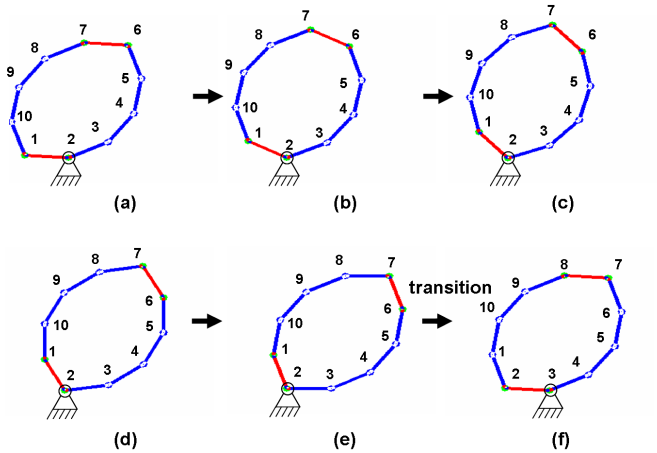


Fig. 5. Different phases of the rolling motion illustrating the effect of shape change at touchdown and subsequent falling motion of the robot. Modules change shape in (a) - (e) note apex nodes 1,6 change to 2,7 and start to pivot rigidly about node 3 (f).

angle θ_a and the global angle made by the robot with the ground θ_g (Figure 4). Each module is considered to be a thin rod of length 0.06 m with mass 0.138 kg (from Table I). The resultant equation for the evolution of θ_g can be expressed in the form:

$$\ddot{\theta}_g = f_1(\theta_a, \theta_g)mg + f_2(\theta_a, \theta_g)\tau. \tag{2}$$

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

Equation 2 shows that there are two contributing terms to the angular acceleration of the robot: (1) the moment due to gravity about the point of contact with the ground and (2) a coupling term arising from the coupling of θ_a and θ_g. While the moment arm due to gravity is always towards the direction of rolling, the direction of the coupling term is initially against the direction of rolling. Thus, the robot first decelerates during the beginning of the shape change phase (as shown by the local deceleration part of the graphs in Figure 6) and then accelerates due to a change in sign of the coupling term in the remaining part of the shape change phase (shown in local accelerasion phase in Figure 6). The coupling term is initially against the direction of rolling since it is applying torque at the appropriate modules to set the shape change in motion. It then reverses sign to slow down this motion to lock its joints in place before the free fall phase. After finishing the shape change phase, it then continuously accelerates in the free-fall phase (shown in free fall phase in Figure 6) solely due to the influence of the moment-arm due to gravity. As the robot rolls faster, the duration of the free-fall phase gets shorter. Beyond a certain speed, it is possible that the robot is unable to go through its complete shape change before touchdown in which case the simplified four-bar model we use in our work is no longer valid. This corresponds to touchdown happening at a time before the free fall region in Figure 6. Since our model is no longer valid in these cases, we do not report or use these results for further analysis.

We define a *step* of the gait as the sequence of events between consecutive touchdowns of two adjacent modules. At touchdown, we reassign the nodes to the different links based on the global positions of the nodes. This is illustrated in Figure 5. In Figure 5(a) joints at 1, 2, 6 and 7 form the joints of the four-bar linkage. After the transition to Figure 5(f) the four bar is represented by the joints 2, 3, 7 and 8 and joint 3 becomes the pivot point around which the fourbar linkage is hinged.

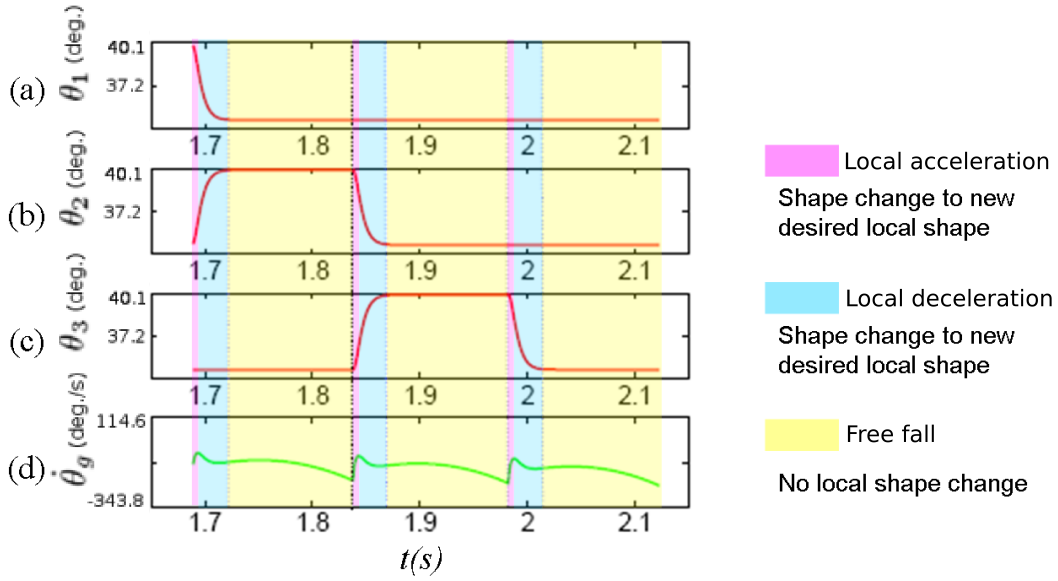


Fig. 6. Simulation results: (a) Joint angle for Module 1 (b) Joint angle for Module 2 (c) Joint angle for Module 3 (d) Angular velocity of the robot over time interval corresponding to three consecutive module touchdowns.

When the module comes into contact with the ground, a transition condition is defined at impact of the module on the ground. Joint angles and the position of the robot stay fixed at transition while velocities are transformed using the transition conditions. The transition condition relates the angular momentum L_- of the whole body of the robot about the new pivot point on the ground just before impact with the angular momentum L_+ of the whole body about the new pivot point after impact. Using a coefficient of restitution η (found empirically to be 0.94 on carpet flooring), the transition condition is given by the momentum transfer equation 3 on impact.

$$L_+ = \eta L_-. \tag{3}$$

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

A. *Scalability*

The particular choice of parameterization made for the controller earlier in Section V has the advantage of making the controller *easily scaleable* to configurations with a different number of modules. This has an advantage in designing controllers for modular robots since it reduces design and computational requirements for control and makes the controller invariant to the number of modules in the loop. Consider, for example, Figure 7 where a loop robot with n modules is shown. In Figure 7(a), the apex nodes are 1 and m where $m = 1 + n/2$. The link joining nodes 1 and 2 (Link 1) and the link joining nodes m and $m+1$ (Link 2) form two links of the four-bar used for analysis and control. The third link (Link 3) is formed by a combination of the links joining joints 2 through m and the fourth link (Link 4) is formed by a combination of the links joining joints $m+1$ through n and 1. Thus, a multi-degree of freedom rolling loop with n modules can be reduced to the same four-bar linkage used for analysis.

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

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

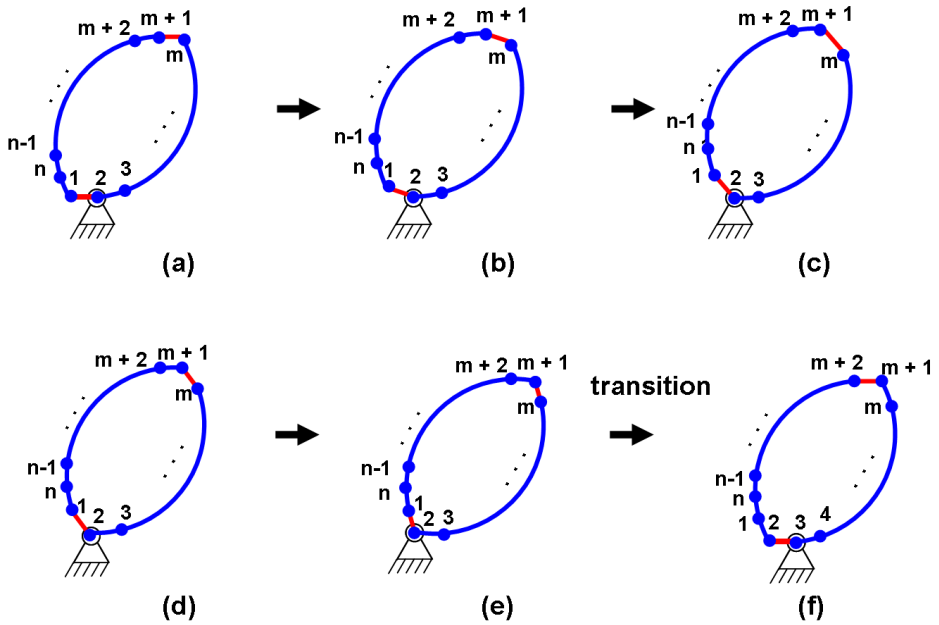


Fig. 7. Scalability of the controller to different number of modules.

V. Control

In Section IV, we made a simplifying assumption that allowed parameterization of the desired shape at touchdown using a single parameter, the apex angle θ_a. The controller used for dynamic rolling can now be described by specifying a new desired shape for the robot at touchdown such that the robot is falling forwards with respect to the pivot point describing the contact of the robot with the ground. This corresponds to designating node 7 and 2 in Figure 5(a) as the new apex angles of the shape. When a new desired shape is specified the loop changes shape as is illustrated in Figure 5(b)-(c)-(d). Once it reaches the new desired position, the local shape does not deform anymore. Now, the robot undergoes a pure falling motion (Figure 5(d)). The robot falls like an inverted pendulum until node 3 touches down on the ground (Figure 5(e)).

Shapes that are more elongated (corresponding to higher values of θ_a) will result in a larger moment arm and higher angular acceleration. However, the amount of shape change (represented by the net change of joint angles) is also higher. Rounder shapes correspond to a smaller value of θ_a and will result in a smaller moment arm and smaller amount of shape change. Figure 8 shows the effect of the choice of shape on the moment arm due to gravity. Figure 8(a) shows the smaller moment arm corresponding to a rounder shape and Figure 8(b) shows the larger moment arm corresponding to a more elongated shape. We should expect that more elongated shapes will give us higher accelerations while rounder shapes may be more efficient. We will examine the effect of the desired shape on the speed of the robot by varying the parameter(θ_a).

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

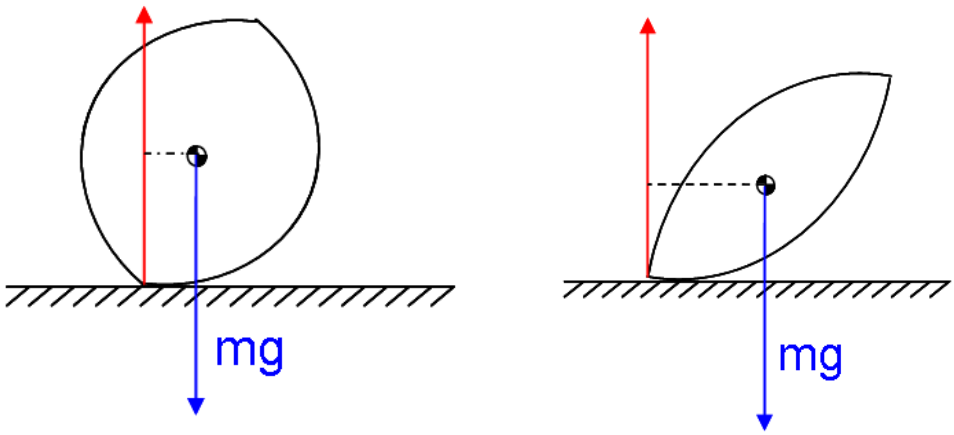
VI. Experiments

A. Terminal velocity and specific resistance

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

1) The robot achieves a terminal speed during rolling and this speed increases with increase in θ_a.
2) Rounder shapes are more efficient.

Reflective markers were placed on the robot to track a single module and its joint angle by a high speed motion capture system (VICON). Measurements were recorded at a speed of 100 Hz and a resolution of 0.1 mm. The overall workspace of the VICON however was limited to 3 m $\times$ 3 m. To determine the terminal velocity the position of the robot was measured manually from video to increase the available workspace. Each trial to determine the terminal velocity consisted of two parts. In the first part a running start of 4 m was given to the robot to allow it to reach terminal velocity. No measurements were



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

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

Fig. 8. Effect of shape on moment arm at touchdown.

taken in this part. In the second part, the robot would continue rolling and position was determined manually from the video footage by marking time stamps as the robot crosses markings on the carpet spaced at 1 foot intervals. The field of view of the camera covered only the second part of the trial. No significant accleration was seen in this second part.

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

Specific resistance (ϵ) measures the energy cost of locomotion per unit distance and robot weight and is thus a good measure of efficiency [Weingarten et al., 2004]. It is calculated as follows:

$$\epsilon = \frac{P}{mgv} \tag{4}$$

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

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

B. Motion on inclines

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

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

C. Scalability of the controller

As noted earlier, the particular parameterization chosen for the rolling loop makes it easier to scale the controller to loops with different number of modules. This was tested by implementing the controller on rolling loop configurations with 8, 10, 12 and 14 modules. The controller maps the links and joints of all these configurations onto the four-bar like *backbone curve*

used earlier for analysis. The user chooses the value of θ_a and θ_s can be easily determined from θ_a using Equation 1. The controller then designates the module touching the ground and the one diametrically opposite as the apex and sets all other joint angles to a constant value of θ_s. The control algorithm for n modules is thus the same as the one used for the loop with 10 modules. This demonstrates the versatility and scalability of the controller.

D. Speed control

Results from the terminal velocity experiments showed that more acceleration occurs in the falling phase at greater apex angles. This led to experiments to demonstrate arbitary speed control with a human specifying the desired speed using a joystick in real time. The robot could be sped up by increasing the apex angle specified by the controller. Braking motion to slow the robot was achieved by designating the module in front of the current touchdown module as the apex of the new desired shape. Snapshots are shown in Figure 15 and in Extension 2.

E. Experimental setup

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

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

VII. Results

A. Tracking of desired joint trajectories

Figure 9 plots experimental tracking results for one of the joint angles of the robot and also the global position of the robot. Figure 9(a) shows the height of one of the modules and it should be noted that the crests in the z positions in Figure 9 represent touchdowns for the module diametrically opposite the tracked module while the troughs represent touchdowns for the tracked modules themselves. The joint angle of this module is shown in the middle figure and we can verify that the module reaches $\theta_a = \pi/4$ and goes back to $\pi/6$ and that this motion is triggered upon touchdown of the module *i.e.* when the z position is at a minimum. Note that the duration where the module holds the apex angle θ_a is very short.

B. Terminal velocity and specific resistance

Figure 10 plots simulation and experimental results for the final speed of the robot for different desired shapes at touchdown. As predicted in the analysis of the model, the observed behavior of the system was that a terminal velocity was reached. In addition, as the desired shape becomes more and more elongated (corresponding to increase in the value of θ_a), the terminal velocity achieved by the robot increases. Also as the desired shape grows elongated, the angular acceleration of the robot in its free fall phase also increases thus resulting in a higher terminal speed. Shapes with an apex angle greater than 70^o cannot keep up with the speed because the servo cannot move fast enough to reach the next shape before the next touchdown.

Below a certain magnitude of shape change, the robot has zero terminal velocity. In other words, even when given an initial velocity there exists acertain $\theta_a < \theta_{critical}$ at which point the robot will eventually roll down to a stop. Where $\theta_{critical}$ is defined as the apex angle with which the robot has just enough energy to continue motion and continue rolling after touchdown rather than rolling back to the previous touchdown. Note in the continuous case (with infinite modules) $\theta_{critical}$ approaches 0 at which point it is a perfect circle. Geometric observation shows that when $\theta_a = 37^o$, the center of gravity sits over the new touchdown point. In simulation, with values of θ_a less than 37^o, the robot slows down to a halt even if it has some initial momentum. For $\theta_a \geq 37^o$, the robot is able to sustain its momentum in simulation and roll continuously. Experimentally, the robot does not achieve continuous motion unless $\theta_a > 40^o$. The experimental terminal velocities are close to the predicted velocities.

Figure 11 plots simulation and experimental values for the specific resistance for different desired shapes. The power determined analytically should be lower than the actual power input to the robot, because the simulation only takes into account the power used to change shape. This is shown in our results. The experimental measurements show larger specific resistance than the theoretical measurements in all trials. More importantly the trend stays the same, *i.e.* rounder shapes exhibit lower specific resistance and are more efficient.

Another estimate of energy efficiency of a gait is the amount of travel in joint space that each module must move in order to move forward. This is measured by the difference between the two angles $\theta_a - \theta_s$. By this measure, rounder shapes also use less energy than the more elongated ones as $\theta_a - \theta_s$ is smaller. It is worthwhile noting that, based on this measure, dynamic gaits with rounder shapes are also more efficient than kinematic gaits. Maintaining any velocity using a purely kinematic gait

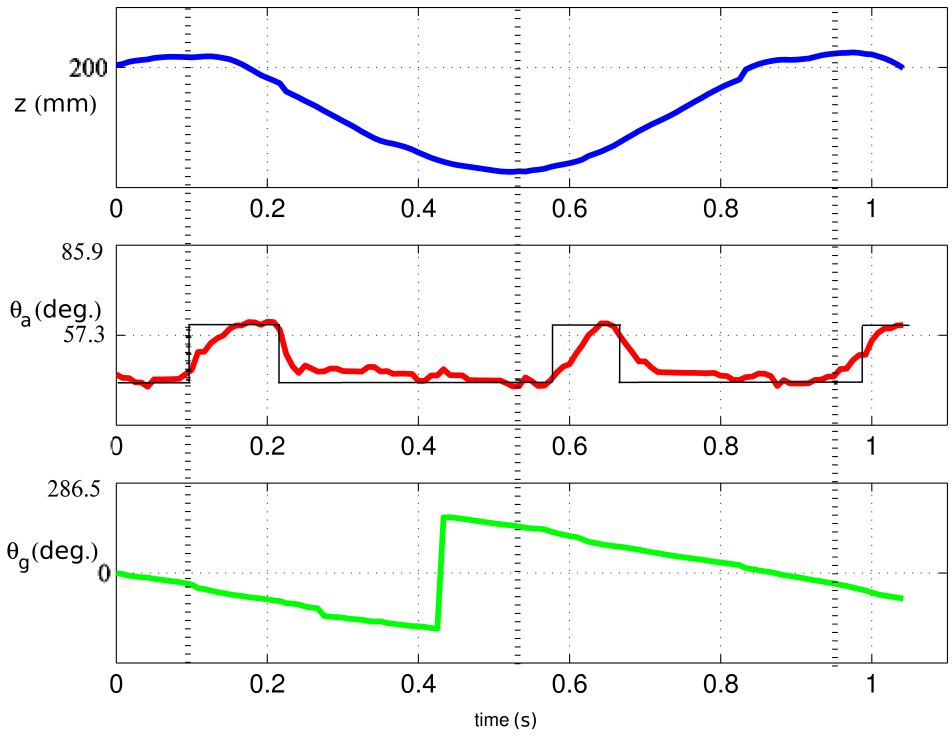


Fig. 9. Experimental results: Tracking results for one module using a motion capture system: z represents height of tracked module above the ground for one cycle, α represents joint angle of tracked module, α_g represents pitch of tracked module with respect to global reference frame (the discontinuity in the data is because of a jump from $-\pi$ to π).

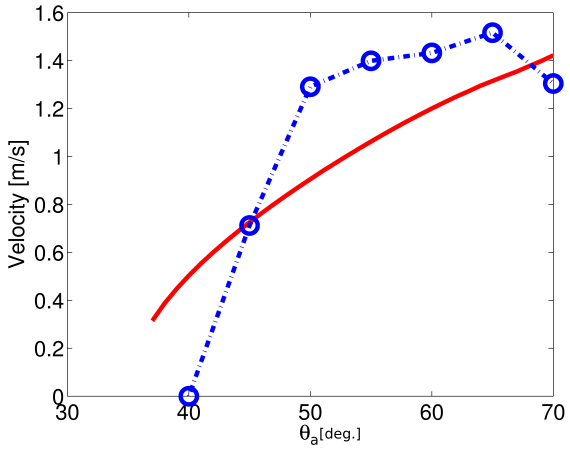


Fig. 10. Terminal velocity vs. θ_a. Simulation shown by a solid line and experimental results by a dotted line (with (o)'s).

typically requires a large traversal of modules in joint space while, once some speed has been built up, dynamic gaits can be sustained using smaller effort in joint space.

To compare these numbers with those for a kinematic gait, specific resistances for different dynamic rolling gaits as well as kinematic rolling gaits are plotted against terminal velocity as shown in Figure 12. The kinematic rolling gait has higher specific resistance than all the dynamic rolling gaits which implies that the amount of energy used to move a unit distance is lower in dynamic rolling than in kinematic rolling, which is what one would expect. For completeness, the electrical power consumed by the total robot is presented in Table III for the kinematic as well as the dynamic gait. This gives an indication of the absolute power consumed by the robot over time. However, specific resistance represents a normalized non-dimensional measure of efficiency that can be better used to compare the performance of this robot with other similar locomotion systems.

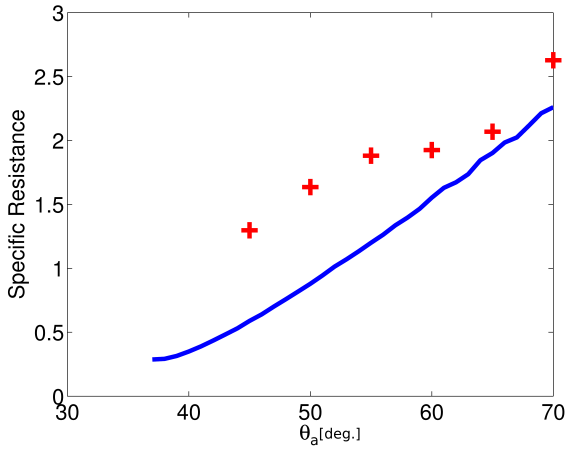


Fig. 11. Specific resistance vs. θ_a. Simulation results shown with a solid line and experimental results with a '+'.

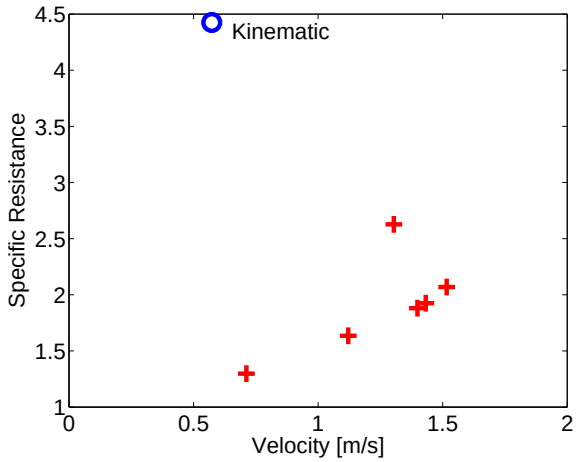


Fig. 12. Experimental results: specific resistance vs. velocity. Dynamic rolling (shown with '+') and kinematic rolling (shown with 'o').

C. Motion on inclines

Figure 13 summarizes the terminal velocities of the robot in a dynamic rolling gait on different inclines (-5^0, 0^0, and 5^0). On each incline, values of θ_a between 36^0 to 70^0 were used. Video of the incline experiments can be found at Extension 4.

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

D. Scalability

Figure 14 shows that terminal velocity increases logarithmically with an increase of 3.7 times between 8 and 10 modules. There is only a small increase found between 10 and 12 modules and no significant difference between 12 and 14 modules.

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

TABLE III

APEX ANGLE (IN DEGREES) VS. POWER (IN WATTS)

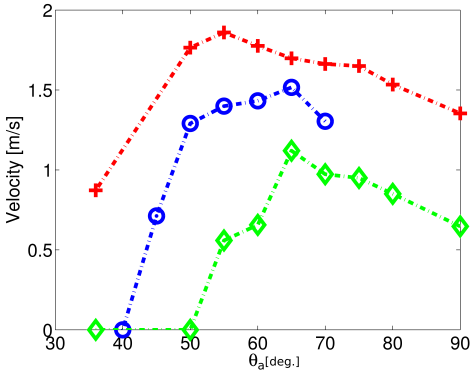


Fig. 13. Experimental results: Terminal Velocity (in m/s) vs. Apex Angle (in degrees) on 5^o (shown with '+'), 0^o (shown with 'o') and -5^o (shown with '$\diamond$') inclines.

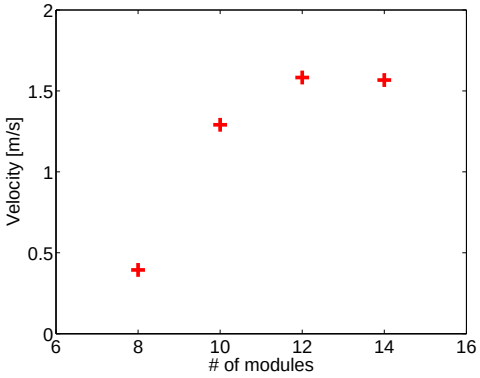


Fig. 14. Terminal velocity vs. number of modules. Apex angle was 50 degrees in all experiments.

The terminal velocity saturates and approaches a limit at 1.6 m/s. Video of the experiments can be found at Extension 3.

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

This number is a measure of speed that accounts for the difference in size of the loops and shows that a loop with more joints has a higher velocity in terms of loop lengths per second. A loop with more joints can more accurately accurately approximates the arc of the shape. Thus, these results show that if the arc is more accurately approximated the faster the gait is.

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

VIII. Discussion

One of the major findings of this work is that elongated desired shapes at touchdown for a rolling loop lead to higher terminal velocities. This is shown through a combination of simulation and experiments. The result makes sense intuitively as more elongated shapes create a larger moment arm due to the center of gravity w.r.t. the ground contact point. Because of this greater moment more acceleration occurs in the falling motion and more energy is put into the system at each step, a result that agrees with our theoretical predictions. It is interesting to note that the acceleration phase of the dynamic gait is similar to the motion of an inverted pendulum which has been shown in the context of walking to be very efficient requiring no work input to move the center of mass [Kuo et al., 2005]. The fastest experimental gait had a speed of 1.6 m/s (roughly 5.4 body lengths per second for the 10 module robot.) Since the robot can reconfigure to different bodies lengths and use different gaits for different applications, a more apt measure for speed may be to normalize to module size. Using this measure the 1.6 m/s translates to 26 module lengths per second. To the authors' knowledge this is the fastest gait for any untethered modular robot.

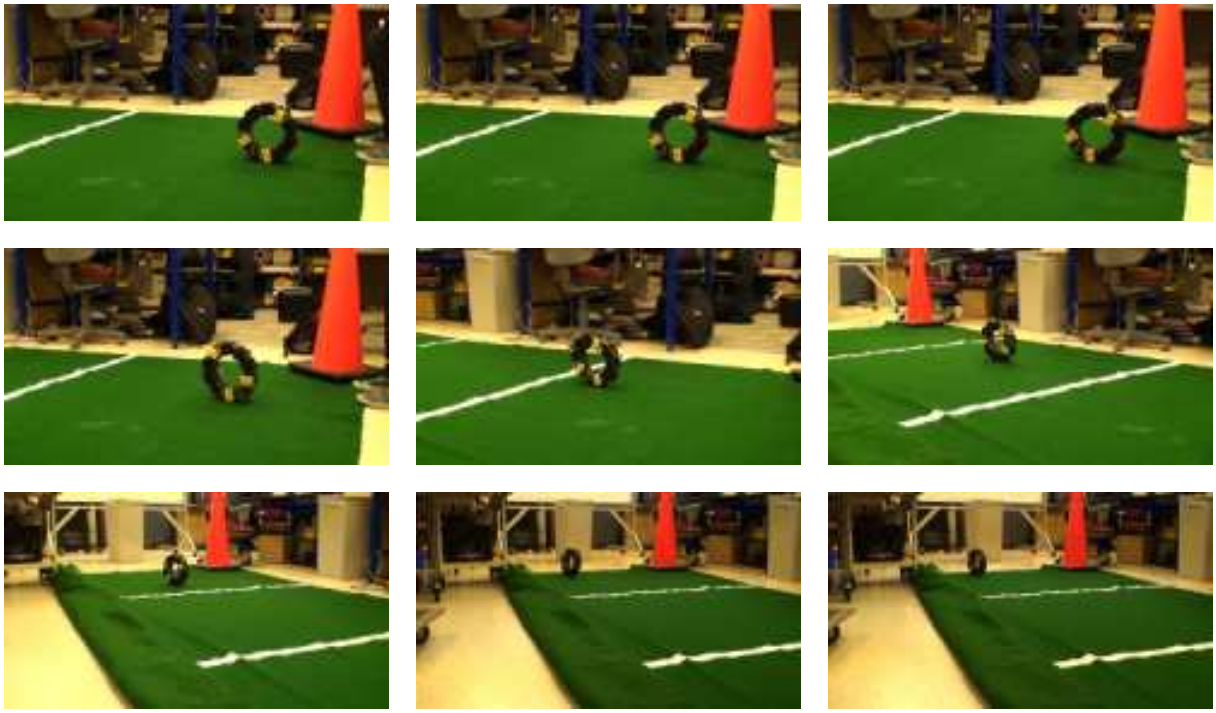


Fig. 15. Snapshots of the rolling motion. Speed was controlled by a joystick in real time. Frames are in chronological order from left to right, then top to bottom. The motion of the robot is from right to left.

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

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

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

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

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

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

Figure 16 shows an interesting comparison between the specific resistance of the dynamic rolling gait of CKBOT, other gaits for CKBOT (including a kinematic rolling gait, a crawling gait and an inchworm gait) and other robotic systems with non-wheeled modes of locomotion like walking. Here, all quantities have been plotted on a logarithmic scale. Ideally, we

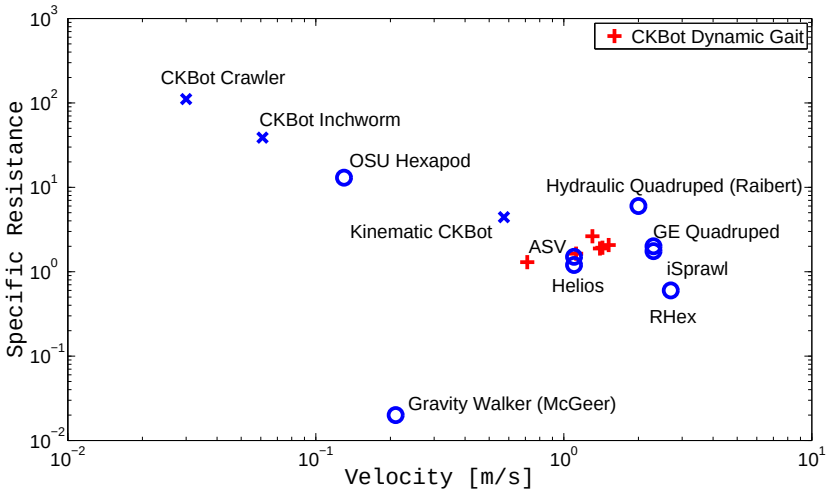


Fig. 16. Specific resistance vs. velocity for several robotic systems. The data for walking systems included here is from [Gregorio et al., 1997].

would like to have as low a specific resistance as possible at as high a speed as possible. This corresponds to being on the lower right corner of the graph. Robots shown in Figure 16 range from very energy efficient robots such as the Gravity Walker by McGeer [McGeer, 1990] to very fast robots like RHex [Weingarten et al., 2004] and iSprawl [Kim, 2006].

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

Figure 16 also compares the dynamic rolling gait with other modular robot gaits. In Section VII, we saw that the dynamic rolling gait improved on the kinematic rolling gait. From Figure 16 it can be seen that it is a dramatic improvement on the inchworm gait and crawler gait. The inchworm gait was implemented with 10 CKBOT modules as well whereas the crawling gait was implemented with only 2 modules. These gaits are very slow and energy inefficient. The kinematic rolling gait is shown to have greater performance in terms of velocity and specific resistance, but it is the dynamic rolling gait that has pushed modular robots into the same range as walking systems.

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

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

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

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

IX. FUTURE WORK

This work is part of a research effort to develop modular robots with a large number of modules and controllers that scale. In this context, we aim to develop controllers that can be easily adapted to a wide variety of modular robot configurations. The rolling loop configuration is the first step in this effort since it provides an easy and efficient mode of locomotion to cover large distances. Future efforts in this direction will include finding the maximum incline it can traverse , the ability to turn and the exploration of optimization techniques to find optimal gaits. We also aim to examine dynamic controllers that allow the robot to adapt to rough terrain and locomote over unstructured environments using a conformal gait where the shape of the robot conforms to the profile of the ground in a dynamic fashion.

Figure 16 includes data for several robots that have improved on their specific resistance or terminal velocity using novel control and design techniques. iSprawl has a compliant leg design and achieves very fast speeds velocities and low specific resistance. By incorporating similar compliance into CKBOT the robot could store energy of the impacts and return it into the system to improve its speed. Learning algorithms were used to tune the gaits for RHex [Weingarten et al., 2004] and resulted in almost a threefold increase in speed and a halving of specific resistance. These techniques could be adapted to rolling in order to achieve similar gains in both maximum speed and specific resistance. One strategy could be to use elongated shapes initially to achieve a high speed, then switch to a more energy efficient mode by using rounder shapes to maintain this speed.

The dynamic gait implemented in this work exploits the *passive falling dynamics* of the modular loop robot. Significant work has been performed in this area for walking robots [Collins et al., 2005], [Collins et al., 2001] where controllers are developed to take advantage of the passive dynamics of the robots to reduce torque requirements on the actuators. Indeed, McGeer's gravity walker [McGeer, 1990] in Figure 16 has the lowest specific resistance amongst robots included in that Figure. Since modular robots have multiple actuated degrees of freedom, controllers that can reduce the requirements on the actuators would present significant benefits in extending the range and duration of operation of these systems.

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

REFERENCES

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

[Buehler et al., 2005] Buehler, M., Iagnemma, K., and Singh, S. (2005). *The 2005 DARPA Grand Challenge: The Great Robot Race*. Springer.

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

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

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

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

[Chitta et al., 2007] Chitta, S., Vemaza, P., Geykhman, R., and Lee, D. D. (2007). Proprioceptive localilzatilon for a quadrupedal robot on known terrain. In *ICRA*, pages 4582–4587.

[Collins et al., 2005] Collins, S. H., Ruina, A., Tedrake, R., and Wisse, M. (2005). Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085.

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

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

[Gregorio et al., 1997] Gregorio, P., Ahmadi, M., and Buehler, M. (1997). Design, control and energetics of an electrically actuated legged robot. *IEEE Transactions on Systems, Man and Cybernetics*, 27(4):626–634.

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

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

[Kamimura et al., 2001] Kamimura, A., Murata, S., Yoshida, E., Kurokawa, H., Tomita, K., and Kokaji, S. (2001). Self-reconfigurable modular robot - experiments on reconfiguration and locomotion. In *Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS200)*, pages 606–612, Maui, Hawaii.

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

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

[Lee and Sanderson, 2000] Lee, W. H. and Sanderson, A. (2000). Dynamic rolling of modular robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco.

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

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

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

[Park et al., 2006] Park, M., Chitta, S., Teichman, A., and Yim, M. (2006). Automatic configuration recognition methods in modular robots. Submitted to IJRR.

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

[Playter et al., 2006] Playter, R., Buehler, M., and Raibert, M. (2006). Bigdog. In Gerhart, G. R., Shoemaker, C. M., and Gage, D. W., editors, *Unmanned Systems Technology VIII*, volume 6230(1), page 62302O. SPIE.

[Shen et al., 2006] Shen, W.-M., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M., and Venkatesh, J. (2006). Multimode locomotion via superbot robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2552–2557, Orlando, Florida.

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

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

[Weingarten et al., 2004] Weingarten, J. D., Lopes, G. D., Buehler, M., Groff, R. E., and Koditschek, D. E. (2004). Automated gait adaptation for legged robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, New Orleans, LA.

[Yamauchi, 2004] Yamauchi, B. M. (2004). Packbot: a versatile platform for military robotics. In Gerhart, G. R., Shoemaker, C. M., and Gage, D. W., editors, *Unmanned Ground Vehicle Technology VI*, volume 5422(1), pages 228–237. SPIE.

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

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

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

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

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

APPENDIX

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

$$f_1 = -(l(10\cos(\theta_g) + 11\cos(\theta_s + \theta_g) + 20\cos(\frac{\theta_s}{2})\cos((5\theta_s)/2 + \theta_g)))/(4(4I_0 + 9ml^2 + 2ml^2(6\cos(\theta_s) + 3\cos(2\theta_s) + \cos(3\theta_s))));$$

$$f_2 = -2/(4(4I_0 + 9ml^2 + 2ml^2(6\cos(\theta_s) + 3\cos(2\theta_s) + \cos(3\theta_s))));$$

$$\begin{aligned}\ddot{\theta}_a = (mgl((4I_0 + 9l^2m)(10\cos(\theta_g) + 11\cos(\theta_s + \theta_1) + 20\cos(\theta_s/2)\cos((5\theta_s)/2 + \theta_g)) \\ - 40(4I_0 + 9l^2m + 2l^2m(6\cos(\theta_s) + 3\cos(2\theta_s) + \cos(3\theta_s)))\cos(3\theta_s + \theta_g + \theta_a)) + \\ 2(5(4I_0 + 9l^2m) + 8l^2m(6\cos(\theta_s) + 3\cos(2\theta_s) + \cos(3\theta_s)))\tau)/(4(4I_0 + 9l^2m)(4I_0 + 9l^2m + 2l^2m(6\cos(\theta_s) + 3\cos(2\theta_s) + \cos(3\theta_s))));\end{aligned}$$

$$\ddot{\theta}_s = 0.$$

TABLE IV
MULTI MEDIA EXTENSIONS

Extension	Type	Description
1	Video	This movie shows three modes of rolling, 1) kinematic rolling 2) failed kinematic rolling 3) dynamic rolling
2	Video	This movie shows dynamic rolling under user teleoperated control moving forward and backward.
3	Video	This movie shows dynamic rolling gaits with 4 different sizes: 8 modules, 10 modules, 12 modules and 14 modules.
4	Video	This movie shows dynamic rolling up and down a 5 degree incline.