5-1-2009

Control of Locomotion with Shape-Changing Wheels

Daniel Mellinger
University of Pennsylvania, dmel@seas.upenn.edu

Vijay Kumar
University of Pennsylvania, kumar@grasp.upenn.edu

Mark Yim
University of Pennsylvania, yim@seas.upenn.edu

Control of Locomotion with Shape-Changing Wheels

Abstract
We present a novel approach to controlling the locomotion of a wheel by changing its shape, leading to applications to the synthesis and closed-loop control of gaits for modular robots. A dynamic model of a planar, continuous deformable ellipse in contact with a ground surface is derived. We present two alternative approaches to controlling this system and a method for mapping the gaits to a discrete rolling polygon. Mathematical models and dynamic simulation of the continuous approximation and the discrete *n*-body system, and experimental results obtained from a physical modular robot system illustrate the accuracy of the dynamic models and the validity of the approach.

Disciplines
Engineering

Comments
Suggested Citation:
Mellinger, D., V. Kumar and M. Yim. (2009). Control of Locomotion with Shape-Changing Wheels. *2009 IEEE International Conference on Robotics and Automation.* Kobe, Japan. May 12-17, 2009.

2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Control of Locomotion with Shape-Changing Wheels

Daniel Mellinger, Vijay Kumar, and Mark Yim
GRASP Laboratory - University of Pennsylvania

***Abstract*— We present a novel approach to controlling the locomotion of a wheel by changing its shape, leading to applications to the synthesis and closed-loop control of gaits for modular robots. A dynamic model of a planar, continuous deformable ellipse in contact with a ground surface is derived. We present two alternative approaches to controlling this system and a method for mapping the gaits to a discrete rolling polygon. Mathematical models and dynamic simulation of the continuous approximation and the discrete n-body system, and experimental results obtained from a physical modular robot system illustrate the accuracy of the dynamic models and the validity of the approach.**

I. Introduction

Closed chain gaits are an efficient way to locomote reconfigure modular robots [9]. The simplest type of closed chain gaits are kinematic gaits that transition through a series of oblong loop shapes which have been designed to effectively rotate the loop to achieve translation [10], [7]. In these kinematic gaits the inertia of the system plays no role and the motion of the system is completely determined by the change in geometry. Kinematic gaits are conceptually similar to statically stable walking gaits. In [8], a flexible rim with spokes made of shape memory alloy was deformed in such a way that it moved in a kinematic roll. In [4], Matsuda and Murata describe a polygonal robot that is controlled to roll by cyclically modifying the stiffness of joints.

Dynamic gaits have also been developed for modular systems. In these types of gaits the inertia of the system plays an important role in locomotion. Dynamic gaits are conceptually similar to dynamic legged gaits like bipedal walking or running. In [3], Lee and Sanderson developed controllers for polygons (and polyhedrons) in which the accelerations of edge lengths were controlled to cause tipping motions over desired vertices (and edges) in order to move in a series of discrete steps. In [6], Sastra, Chitta, and Yim built on this work by using feedback control to cause continuous motion. They implemented controllers for an American football shaped configuration of modular robots that transitioned between shapes based on which module was in contact with the ground. The goal of the controller was to keep the center of mass in front of the contact point.

The design of the controllers for modular robots forming polygonal closed chains has been hampered because of the difficulty in modeling the dynamics of a large number of links and the various contact conditions [4]. Indeed, there has been no systematic approach to exploit the dynamics of these systems. Additionally, the large number of actuated joints creates a high-dimensional input space making the design of gaits for such systems difficult [6].

This difficulty motivates the idea of creating a low-dimensional abstraction of the Discrete Polygonal System (DPS). We derive an Abstract Continuous Model (ACM) from a smooth closed curve that approximates the DPS and use it to derive gaits in the low-dimensional space, before mapping the control inputs onto the high-dimensional DPS.

In this paper, we use a continuous deformable elliptical rim (which we refer to as an ellipse) as an ACM for shape-changing wheels consisting of discrete, rigid modules. We model the dynamics of rolling, deformable ellipses and synthesize shape-changing gaits for the ellipse. We consider two alternative approaches with different shape variables to controlling the locomotion of the deformable ellipse. The first approach involves maintaining the shape of the wheel but controlling the rate at which material (modules) moves along the rim of the wheel as shown in the left panel of Figure 1. We call this abstraction applied to an ellipse the *Tread-Controlled Ellipse*. The second approach involves the control of the shape of the wheel to achieve locomotion as shown in the right panel of Figure 1. We call this abstraction applied to an ellipse the *Shape-Controlled Ellipse*.

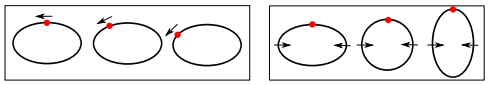

Fig. 1. Tread-Controlled (left) and Shape-Controlled (right) Ellipse

Next we develop gaits for these two types of control inputs to drive the deformable elliptical rim. The gaits are then mapped from the ACM to the DPS. Finally, the gaits are implemented in simulation and on a physical system consisting of 12 single degree-of-freedom modular robots connected in a loop.

II. Problem Formulation

A. Terminology

The Abstract Continuous Model (ACM) for a rolling polygonal wheel is derived by considering the ellipse shown in the left panel of Figure 2 which is constrained to lie in the plane of the page. The ellipse is in contact with the ground surface at point B, fixed to the disk. The contact point which moves along the ground surface and is not fixed to the ellipse is D. $\hat{\mathbf{e}}_\mathbf{i}$ are an orthonormal basis set for the earth-fixed reference frame whose origin is point O. $\hat{\hat{\mathbf{e}}}_\mathbf{i}$ are an orthonormal basis set for the body-fixed reference frame whose origin is the center of the ellipse C. The length of the axes of the ellipse are a and b. The angle the ellipse makes with the horizontal is θ.

We gratefully acknowledge the support of NSF grant IIS-0413138, ARO MURI Grant W911NF-05-1-0219, ONR Grant N00014-07-1-0829, and ARL Grant W911NF-08-2-0004.

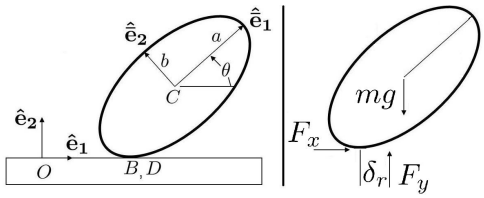

Fig. 2. The ACM for a shape-changing, polygonal wheel (left) and its Free Body Diagram (right)

We parameterize the curve by a coordinate ξ (not shown in figure) so that any generic point Q on the rim of the ellipse can be described in the body-fixed frame in terms of $\hat{\mathbf{e}}_\mathbf{1}$, $\hat{\mathbf{e}}_\mathbf{2}$ by:

$$\bar{x}_{CQ} = acos\left(\xi_Q\right), \quad \bar{y}_{CQ} = bsin\left(\xi_Q\right) \tag{1}$$

In this equation and in the remainder of this paper we will use, for example, CQ, to denote the position vector $\overrightarrow{CQ}$ and $x_{(.)}$ and $y_{(.)}$ to denote components of the vector along $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$, and $\bar{x}_{(.)}$ and $\bar{y}_{(.)}$ to denote components along $\hat{\bar{\mathbf{e}}}_i$.

B. Contact Point and Rolling Assumptions

The contact point is found by finding ξ for which the derivative of the y_{CQ} with respect to ξ goes to zero. Note this equation yields two solutions, but the one for which $y_{CQ} < 0$ is the contact point.

$$x_{CD} = \frac{\left(b^2 - a^2\right) sin\theta cos\theta}{\sqrt{a^2 sin^2\theta + b^2 cos^2\theta}} \tag{2}$$

$$y_{CD} = -\sqrt{a^2 sin^2\theta + b^2 cos^2\theta} \tag{3}$$

Note that points B and D are instantaneously at the same location. Therefore the expressions developed above for the location of point D can be used for the location of point B. However, it is important to realize that their time derivatives differ because point B is fixed to the ellipse while point D is not.

It is assumed that the ellipse remains in contact with the ground surface. This assumption implies that the contact point D moves along the ground surface. Therefore its velocity and acceleration components normal to the surface, $\frac{dy_{OD}}{dt}$ and $\frac{d^2y_{OD}}{dt^2}$, must be zero.

It is also assumed [1] that there is sufficient friction so that the ellipse is rolling on the ground surface without slipping. This pure rolling assumption implies that the velocity and acceleration of the contact point B fixed to the ellipse in the direction tangent to the contact surface, $\frac{dx_{OB}}{dt}$ and $\frac{d^2x_{OB}}{dt^2}$, must be zero [1].

C. Equations of Motion

The free body diagram for the ellipse is moving in the positive $\hat{\mathbf{e}}_\mathbf{1}$ direction is shown in the right panel of Figure 2. Here we account for rolling resistance by letting the normal force (F_y) be offset from the actual contact point by a distance of δ_r in the direction of rolling. This offset creates a moment which opposes the motion of the system and causes energy loss. The equations of motion corresponding to this free body diagram are

[1]This assumption is validated later via simulations and experiments.

$$m\frac{d^2x_{OC}}{dt^2} = F_x \tag{4}$$

$$m\frac{d^2y_{OC}}{dt^2} = F_y - mg \tag{5}$$

$$\frac{dH_c}{dt} = \left(x_{CD} + \delta_r sign\left(\frac{dx_{OC}}{dt}\right)\right) F_y - y_{CD}F_x \tag{6}$$

where m is the mass, g is the acceleration due to gravity, F_x is the friction force, F_y is the normal force, and H_C is the angular momentum of the ellipse about the center of mass C. The rolling assumptions allow the contact forces to be eliminated and yield one equation of motion:

$$\begin{aligned}\frac{dH_c}{dt} = \quad & \left(x_{CD} + \delta_r sign\left(\tfrac{dx_{OC}}{dt}\right)\right)\left(m\tfrac{d^2y_{OC}}{dt^2} + mg\right) \\ & -y_{CD}m\tfrac{d^2x_{OC}}{dt^2}\end{aligned} \tag{7}$$

D. Internal Dynamics

The internal system is modeled as an elliptical rim of mass m with axis lengths a_{mid} and b_{mid} and thickness l as shown in Figure 3. The mass is assumed to be concentrated in infinitesimal rectangles of dimension l by ds which are always normal to the surface of the ellipse. A scaling parameter, $\sigma \leq 1$, is defined so that $\frac{a_{mid}}{\sigma} = a \approx a_{mid} + \frac{l}{2}$ and $\frac{b_{mid}}{\sigma} = b \approx b_{mid} + \frac{l}{2}$. It assumed that the outer border of this system shown in Figure 3 is close to the ellipse with axis lengths a and b that was previously analyzed. This assumption is good if $\frac{b}{a}$ is close to 1 or l is small.

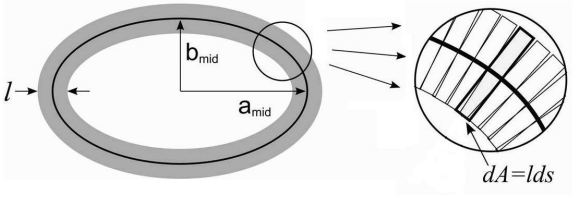

Fig. 3. Internal Dynamics Model

Integrating the angular momentum contribution of each differential element over the entire ellipse gives the total angular momentum:

$$H_C = \left(I_{mid} + \frac{ml^2}{12}\right)\dot{\theta} + \left(\frac{2\pi m a_{mid} b_{mid}}{P} + \frac{\pi l^2 m}{6P}\right) v_t \tag{8}$$

where I_{mid} is the moment inertia about C as if the mass was uniformly concentrated along the line of the ellipse with axis lengths a_{mid} and b_{mid}. For the tread-controlled system where a and b are fixed the derivative of H_C is

$$\frac{dH_c}{dt} = \frac{\partial H_c}{\partial \dot{\theta}}\ddot{\theta} + \frac{\partial H_c}{\partial v_t}\dot{v}_t \tag{9}$$

For the shape-controlled ellipse where the tread is fixed the derivative is

$$\frac{dH_c}{dt} = \frac{\partial H_c}{\partial \dot{\theta}}\ddot{\theta} + \frac{\partial H_c}{\partial a}\dot{a} + \frac{\partial H_c}{\partial b}\dot{b} \tag{10}$$

E. Kinematic Relationships

In this section, we develop expressions for the first and second derivatives of y_{OC} and x_{OC} for both the tread-controlled and shape-controlled ellipse so that they may be substituted into the equation of motion of the system (7). First we write expressions for y_{OC} and x_{OC} and for simplicity introduce the terms $Y = -y_{CD}$ and $X = -x_{CB}$:

$$y_{OC} = y_{OD} - y_{CD} = y_{OD} + Y \tag{11}$$

$$x_{OC} = x_{OB} - x_{CB} = x_{OB} + X \tag{12}$$

1) Tread-Controlled Ellipse: We take derivatives of Equation (11) and recall from the rolling assumptions that derivatives of y_{OD} go to zero:

$$\frac{dy_{OC}}{dt} = Y_\theta \dot{\theta} \tag{13}$$

$$\frac{d^2 y_{OC}}{dt^2} = Y_{\theta\theta}\dot{\theta}^2 + Y_\theta \ddot{\theta} \tag{14}$$

Recall that the position of a generic point on the perimeter is parameterized by ξ. The tread velocity or the rate of material flow around the wheel is the rate of change of the arc length from a reference point to the generic point and is related to $\dot{\xi}$. To find this relationship let us consider point B with the coordinate ξ_B as our generic point and the reference point on the rim to be denoted by ξ_B^*. The arc length is given by:

$$p_t = \int_{\xi_B^*}^{\xi_B} \sqrt{a^2 sin^2\xi' + b^2 cos^2\xi'} d\xi' \tag{15}$$

The magnitude of the tread or rim velocity, v_t, is given by:

$$v_t = \frac{dp_t}{dt} = \sqrt{a^2 sin^2\xi_B + b^2 cos^2\xi_B}\dot{\xi_B} = -X_{\xi_B}\dot{\xi_B} \tag{16}$$

We then take derivatives of Equation (12) and recall from the pure rolling assumption that derivatives of x_{OB} go to 0.

$$\frac{dx_{OC}}{dt} = X_\theta \dot{\theta} + X_{\xi_B}\dot{\xi_B} = X_\theta \dot{\theta} - v_t \tag{17}$$

$$\frac{d^2 x_{OC}}{dt^2} = X_{\theta\theta}\dot{\theta}^2 + X_\theta \ddot{\theta} - \dot{v_t} \tag{18}$$

Substituting expressions in Equations (9), (14), (17-18) into the equation of motion of the system (7) yields the acceleration of the wheel ($\ddot{\theta}$) in terms of the control input ($\dot{v_t}$):

$$\ddot{\theta} = f\left(\theta, \dot{\theta}, a, b, v_t\right) + g\left(\theta, \dot{\theta}, a, b, v_t\right)\dot{v_t} \tag{19}$$

where f and g are nonlinear functions. This equation will be used to develop gaits for the system in Section III.

2) Shape-Controlled Ellipse: In the shape-controlled ellipse a and b can be changed while the perimeter remains constant. The integral for the perimeter of an ellipse has no closed-form solution but we can still set its first and second derivatives to zero in order to find the relationship between the derivatives of the shape variables a and b. Note that this yields an affine relationship between $\ddot{a}$ and $\ddot{b}$. We consider the control input to be $\ddot{a}$ and let b change accordingly.

We now differentiate Equation (11) as before. Once again the derivatives of y_{CD} go to zero but there are additional terms because a and b are not fixed. Taking derivatives yields:

$$\frac{dy_{OC}}{dt} = Y_a \dot{a} + Y_b \dot{b} + Y_\theta \dot{\theta} \tag{20}$$

$$\begin{aligned}\frac{d^2 y_{OC}}{dt^2} &= Y_{aa}\dot{a}^2 + Y_{bb}\dot{b}^2 + Y_{\theta\theta}\dot{\theta}^2 + Y_a\ddot{a} + Y_b\ddot{b} \\ &\quad + Y_\theta\ddot{\theta} + 2Y_{ab}\dot{a}\dot{b} + 2Y_{a\theta}\dot{a}\dot{\theta} + 2Y_{b\theta}\dot{b}\dot{\theta}\end{aligned} \tag{21}$$

In order to differentiate x_{OC} we first note that the arc length between a point on the ellipse and any of the four points at $\xi = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$ is fixed. Let L be the constant arc length between $\xi = 0$ and point B. Taking derivatives of the constant L yields an affine relationship between $\dot{\xi}$ and $\dot{a}$ and an affine relationship between $\ddot{\xi}$ and $\ddot{a}$. The first and second time derivative of Equation (12) for the shape-controlled system are

$$\frac{dx_{OC}}{dt} = X_a \dot{a} + X_b \dot{b} + X_{\xi_B}\dot{\xi_B} + X_\theta \dot{\theta} \tag{22}$$

$$\begin{aligned}\frac{d^2 x_{OC}}{dt^2} &= X_{aa}\dot{a}^2 + X_{bb}\dot{b}^2 + X_{\xi_B\xi_B}\dot{\xi_B}^2 + X_{\theta\theta}\dot{\theta}^2 + X_a\ddot{a} \\ &\quad + X_b\ddot{b} + X_{\xi_B}\ddot{\xi_B} + X_\theta\ddot{\theta} + 2X_{ab}\dot{a}\dot{b} + 2X_{a\xi_B}\dot{a}\dot{\xi_B} \\ &\quad + 2X_{a\theta}\dot{a}\dot{\theta} + 2X_{b\xi_B}\dot{b}\dot{\xi_B} + 2X_{b\theta}\dot{b}\dot{\theta} + 2X_{\xi_B\theta}\dot{\xi_B}\dot{\theta}\end{aligned} \tag{23}$$

We can then substitute the expressions in Equations (10), (21), (22-23) into the equation of motion of the system (7) to get an expression of the form:

$$\ddot{\theta} = \bar{f}\left(\theta, \dot{\theta}, a, b, \dot{a}\right) + \bar{g}\left(\theta, \dot{\theta}, a, b, \dot{a}\right)\ddot{a} \tag{24}$$

We develop gaits for the shape-controlled ellipse using this equation in Section IV.

III. Gait for Tread-Controlled Ellipse

We use feedback linearization starting with Equation (19) by creating virtual control input u so that $\dot{v_t} = \frac{1}{g}(-f + u)$ to yield $\ddot{\theta} = u$. A PD controller can now be used to make θ track a desired trajectory. Consider the idea of having θ track a constant value between 0 and $\frac{\pi}{2}$. In this position the center of mass of the ellipse is to the right of the contact point with the ground. For this angle to be maintained the ellipse must be experiencing a constant acceleration to the right. In a similar way if θ is between $-\frac{\pi}{2}$ and 0 then the ellipse must accelerating to the left. Therefore we can control the position of the ellipse by controlling its angle. We call this gait the "Road Runner Gait" because of the similarity to the way the legs of the cartoon character move as it runs. Note that the principle behind this gait is similar to that of the Segway Personal Transporter where riders lean forward to accelerate and backward to decelerate.

IV. Gaits for Shape-Controlled Ellipse using Energy Based Heuristics

Feedback linearization control does not work on the shape-controlled ellipse because of the complexity caused by constraints on a and $\ddot{a}$. The control input, $\ddot{a}$, must be constrained as a function of the state so that the ellipse does not leave

the ground, $F_y > 0$. Additionally, the axis length a must stay between some minimum and maximum value because of the fixed perimeter constraint. However, a control input obtained by feedback linearization and the PD controller may not satisfy this constraint on a. We instead develop two heuristics for gaits that allow the ellipse to move in any direction.

A. Start-up Phase

The heuristic gaits described later require that there be some initial motion or displacement in order to work. This state of motion is achieved by an initial start-up phase illustrated in Figure 4. If the resting ellipse shown in Figure 4a is made taller it will become unstable and tip over at some point in one direction or the other as shown in Figure 4c. After this instant a heuristic gait can take over and drive the ellipse in either desired direction.

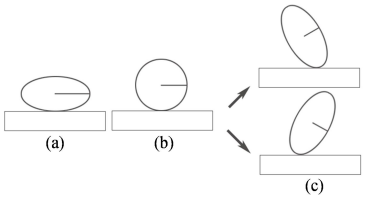

Fig. 4. Start-up Phase for Heuristic Gaits

B. Pump Gait

Consider a rigid ellipse oscillating with small amplitude that does not have enough energy to rock over its long axis. Energy can be added to this system by controlling the ellipse to become more circular near $\theta = 0$, which raises its center of mass, and then returning to its elliptical shape at other times. If energy is continually added to this system the ellipse will eventually gain enough to displace itself in either direction. However, if energy is only added when the ellipse is traveling in the desired direction then it is guaranteed that the ellipse will roll over the desired side. After rolling over the desired side the ellipse will continue to move in the desired direction as more energy is added to the system whenever $\theta = n\pi$.

This gait is shown graphically in Figure 5. In 5a, the ellipse starts at rest at an initial angle θ_0 with some initial $\frac{b}{a}$. The ellipse then rolls to the right through $\theta = 0$ in 5b and reaches an angle $-\theta_0$ in 5c, assuming no energy loss. The ellipse then rolls to the left but this time the ellipse is controlled to become circular around θ_0 (5d) which adds energy to the system. Now the ellipse oscillates back to the left to some greater angle as shown in 5e. The process is repeated until the ellipse gains enough energy to tip over the left side and maintain a continuous roll.

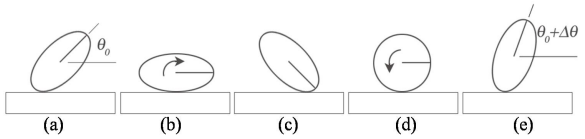

Fig. 5. Graphical Description of Pump Gait to Move Left

The control input for this system is $\ddot{a}$ and is set from the desired a, the current a, and the current $\dot{a}$ according to

$$\ddot{a} = k_p\left(a_{des} - a\right) - k_d\dot{a} \tag{25}$$

where k_p and k_d are gains that can be tuned.

In this gait a is always being controlled to one of two values: the value that makes the ellipse circular ($\frac{b}{a} = 1$) or the value that achieves some desired $\frac{b}{a} < 1$ ratio β. To move the ellipse to the left the ellipse is controlled to be circular only in the range $-\psi < \theta < \psi$ when $\dot{\theta} > 0$. These controlled states are shown in Figure 6. Note that this gait is periodic in π.

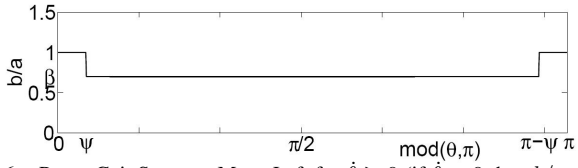

Fig. 6. Pump Gait States to Move Left for $\dot{\theta} \geq 0$ (if $\dot{\theta} < 0$ then $b/a = \beta$)

C. Double Pump Gait

The Double Pump gait is illustrated graphically in Figure 7. The first part of the Double Pump gait is equivalent to that of the Pump gait. Notice that Figures 7a-d are the same as Figures 5a-d. As before, the ellipse is controlled to a circle around $\theta = 0$ in order to raise the center of mass of the system. However, after leaving the region around $\theta = 0$ instead of being controlled back to the original $\frac{a}{b}$ ratio the system is controlled to the reciprocal of the original $\frac{a}{b}$ ratio as shown in 7e. In this way the center of mass is raised again and the ellipse continues to roll in the desired direction. Around $\theta = \frac{\pi}{2}$ the ellipse is controlled to a circle to raise the center of mass (7f) and then finally back to the original $\frac{a}{b}$ ratio after leaving the region around $\theta = \frac{\pi}{2}$ (7g).

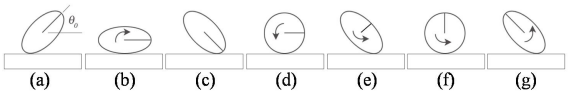

Fig. 7. Graphical Description of Double Pump Gait to Move Left

As with the Pump gait, the control input for this system is computed according to (25). However, for the Double Pump gait, a is always being controlled to one of three values: the value that makes the ellipse circular ($\frac{b}{a} = 1$), the value that achieves some desired axis length ratio $\beta < 1$, or its reciprocal $\frac{1}{\beta}$. Exactly when these values are controlled to is determined by the chosen parameter χ as shown in Figure 8. Note that this gait is also periodic in π.

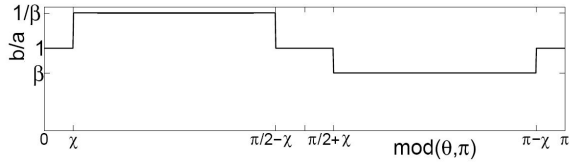

Fig. 8. Double Pump Gait States to Move Left for $\dot{\theta} \geq 0$ (if $\dot{\theta} < 0$ then $b/a = \beta$)

V. Mapping the Continuous to the Discrete System

To implement previously described gaits on the modular robots we develop a method for mapping the control inputs of the abstract continuous model (ACM) to the control inputs of the discrete polygonal system (DPS). In general we consider the problem of taking a parametric curve defined by $x(\xi)$

and $y(\xi)$ for some range of ξ from ξ_i to ξ_f and finding n fixed length connected line segments that approximate it. Note that the fixed length of the line segments, L, is not known a priori.

Our algorithm starts with a guess at the length of the line segments, L, and the constraint that the first endpoint of the first discrete segment lies at beginning of the continuous curve and finds a solution for which all endpoints of line segments lie on the continuous curve. It then iterates on the length of the line segments, L, until the last endpoint of the last line segment lies on the last point of the continuous curve. Note that for a closed curve the starting point and ending point of the curve are equivalent but arbitrary which implies that an infinite number of discrete approximations can be found by changing the starting point of the algorithm.

A. Joint Angles as Functions of Control Input

The tread parameter, p_t, is the integral of the tread velocity or the distance material has traveled along the perimeter of the ellipse. To create a map from p_t to the joint angles of the discrete system the starting point, ξ_i, for the described algorithm is iteratively cycled completely around the ellipse from 0 to 2π. The tread parameter is measured for each solution and the joint angles are recorded. A polynomial curve is then fit to this data so that the n joint angles ψ_i can be written as functions of the form $\psi_i = \psi_i(p_t)$.

For the shape-control input case the starting point of the algorithm is always $\xi_i = 0$ and a multiple of four links is used so that the discrete system is symmetric along both the x and y axis. The algorithm described previously is used to find the joint angles for a range of b/a ratios. A polynomial curve is fit to this data so that we can write $\psi_i = \psi_i\left(\frac{b}{a}\right)$.

VI. Simulations and Experiments

The three gaits described above were simulated on both the ACM and a DPS and implemented on a physical DPS. Videos of these nine experiments can be seen in the video submission accompanying the paper.

The ACM simulation is simply a numerical integration of the equations of motion in Matlab. The DPS simulation was performed in Gazebo [2]. Gazebo is a popular open source simulator that can accurately simulate multiple robots, sensors, and objects in a three-dimensional world. The Gazebo model of the ellipse was built out of 12 rectangular prisms connected with hinge joints which were controlled with a PD controller running at 100 Hz.

The gaits were also implemented on a modular robot called the CKBot [5] shown in Figures 9. Each module has a single rotational degree of freedom actuated by a servo motor which is controlled to track a desired angle with an on board microprocessor running a PD controller. Modules were attached end-to-end to form a loop of 12 robots. A VICON motion capture system consisting of six cameras captures the three-dimensional position of reflective markers placed on the modules. A computer receives the marker data from the VICON system from which θ and $\dot{\theta}$ of the ellipse are calculated. The computer then calculates the desired control input and sends the corresponding desired angles to each of the twelve modules via a CAN messaging system. This closed-loop control system runs at 80 Hz.

The parameters for the modular robot system are $m = 1.83kg$, $l = 0.060m$, $P = 0.869m$, and $\delta_r = 0.009m$. The rolling resistance coefficient, δ_r, was determined by controlling the physical robot to be a circle and rolling it on the test surface by giving it an initial push. The angle of the system as a function of time was recorded. The equivalent situation was then simulated on the continuous model with a range of δ_r and the measured m, l, and P parameters and the best δ_r was chosen. The parameters for the simulated models were matched to the parameters for the physical system.

A. Road Runner Gait

The Road Runner gait was implemented with $k_p = 90\frac{m}{s^2}$ and $k_d = 28.5\frac{m}{s}$ with initial conditions close to $\theta_0 = 0$ and $\dot{\theta}_0 = 0$. The desired tracking angle was ramped up to 0.6 radians over 1 second after which it was held constant which caused the system to accelerate in the positive x direction. Snapshots from the implementation of this gait on the CKBot are shown in the left panel of Figure 9.

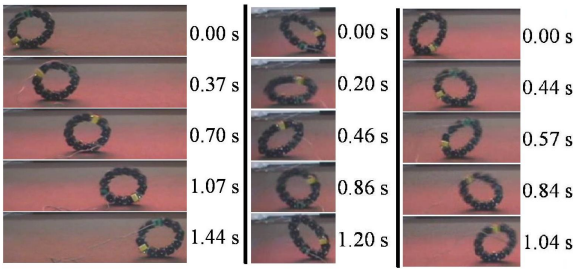

Fig. 9. Road Runner (left), Pump (middle), and Double Pump Gait (right) Snapshots

Data from the experimental and simulation implementations of the Road Runner gait are shown in Figure 10. Notice that the ACM simulation is smooth and the angle is maintained exactly, because the surface of the ellipse is perfectly smooth and there is perfect state feedback and actuation. There is more noise in the other two systems because the discrete segments impact the ground surface as the ellipse rolls which adds a disturbance to the system. Despite this unmodeled disturbance, the two discrete systems are still able to track the desired angle. However, notice that with the discrete systems the tracking worsens with time because the system is moving faster with increasing time which corresponds to a greater frequency of impacts.

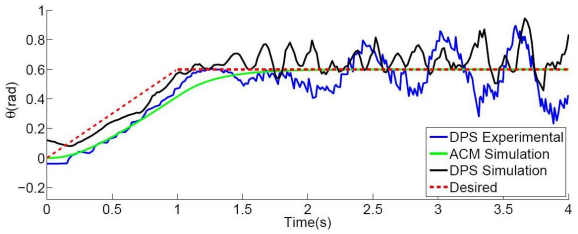

Fig. 10. Road Runner Gait Data Comparison

B. Pump Gait

The Pump gait was also implemented with $\psi = \frac{\pi}{5}$, $\beta = 0.5$, $k_p = 100\frac{1}{s^2}$, and $k_d = 20\frac{1}{s}$ with initial conditions close

to $\theta_0 = 0.8rad$ and $\dot{\theta}_0 = 0$. Snapshots from the first part of the implementation of the Pump gait on the CKBot are shown in the middle panel Figure 9. Note that in this figure the ellipse is controlled to roll to the right. It is clear that energy is added in this cycle because of the increase in angle between the top and bottom images.

Data from the experimental and simulation implementations of the Pump gait are shown in Figure 11. In all three plots the ellipse oscillates with increasing amplitude until the it gains enough energy to maintain a continuous roll. The performance of this gait is strongly based on the how the ellipse stores energy. The joints in the discrete model cause energy loss. Exactly how this energy is lost depends on the characteristics of the 12 actuated hinge joints which differ between the experimental modular robot and the DPS simulation. Notice how the ACM simulation builds up enough energy to maintain a continuous roll faster than the other two. This is likely due to the fact that the surface of the continuous ellipse is completely smooth so there is no energy loss due to impacts.

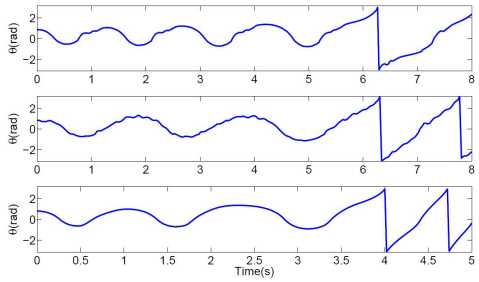

Fig. 11. Pump Gait Data Comparison - DPS Experimental (top), DPS Simulation (middle), and ACM Simulation (bottom)

C. Double Pump Gait

The Double Pump gait was implemented with $\chi = \frac{\pi}{8}$, $\beta = 0.6$, $k_p = 100\frac{1}{s^2}$, and $k_d = 20\frac{1}{s}$. Snapshots from the initial part of the implementation of the Double Pump gait are shown in the right panel of Figure 9. Notice how in the Double Pump gait the ellipse is controlled to $\frac{b}{a} = \beta$ (in the top and bottom image) and $\frac{b}{a} = \frac{1}{\beta}$ (in the middle image.)

After an initial start-up phase the behavior of this gait is primarily determined by the PD gains. Steady-state data from these experiments are shown in Figure 12. Notice that the time for one complete revolution is close to 1.5 seconds for all three data sets which corresponds to a speed of 0.58 m/s.

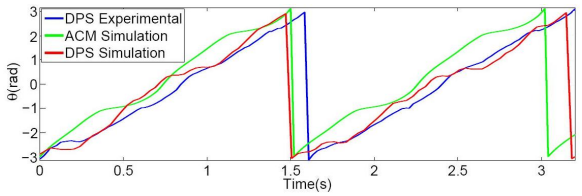

Fig. 12. Double Pump Gait Data Comparison - Steady-State

D. Discussion

Of the three gaits, the Road Runner gait is the fastest and most practical for locomoting the system. However, since the control input is calculated directly from the model and the state feedback it is sensitive to errors in both. The Pump and Double Pump gaits are most interesting because they use a novel control input where it is not obvious that controlled locomotion can be achieved at all. Additionally, the control input is not calculated directly from θ and $\dot{\theta}$ but rather the range in which θ resides so they are not as sensitive to the accuracy of the state feedback. Between these two, the Double Pump gait has the advantage over the Pump gait because it does not require the initial energy buildup phase.

VII. Concluding Remarks

We presented a low-dimensional abstract continuous model (ACM) of a discrete polygonal system (DPS) in the form of a continuous deformable ellipse. We modeled two different types of gaits (tread-controlled and shape-controlled) and obtained a dynamic model for the shape-changing wheel that explicitly reflects the role of the control inputs. The simplified model provided insight into the dynamics of the DPS and eliminated the complexity of dealing with a multi-link system and its various contact conditions. The ACM allowed for the creation of gaits in a low-dimensional space which were then mapped to the high-dimensional DPS. The gaits were implemented on a physical modular robotic system called the CKBot and the data was shown to match well with both that of the simulation of the ACM and the DPS.

The methodology for modeling and applying an ACM to any discrete closed chain robot can be extended to arbitrary continuous curves. Future work will include exploring the optimality of such shapes as well as the theoretical and experimental characterization of the performance and efficiency of this approach.

References

[1] C. Cai and B. Roth. On the planar motion of rigid bodies with point contact. *Mechanism and Machine Theory*, 21:453–466, 1986.

[2] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. *Proc. Int. Conf. on Intelligent Robots and Systems*, 3:2149–2154, 2004.

[3] W. Lee and A. Sanderson. Dynamic rolling locomotion and control of modular robots. *IEEE Transactions on Robotics and Automation*, pages 32–41, 2002.

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

[5] M. Park, S. Chitta, A. Teichman, and M. Yim. Automatic configuration recognition methods in modular robots. *Int. Journal of Robotics Research*, 27:403–421, 2008.

[6] J. Sastra, S. Chitta, and M. Yim. Dynamic rolling for a modular loop robot. *Proc. Int. Symp. on Experimental Robotics*, 2006.

[7] W.M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh. Multimode locomotion via SuperBot reconfigurable robots. *Autonomous Robots*, 20(2):165–177, 2006.

[8] Y. Sugiyama, A. Shiotsu, M. Yamanaka, and S. Hirai. Circular/spherical robots for crawling and jumping. *Proc. IEEE Int. Conf. Robotics and Automation*, 2005.

[9] M. Yim. *Locomotion with a Unit-Modular Reconfigurable Robot*. PhD thesis, Stanford University, 1994.

[10] M. Yim, DG Duff, and KD Roufas. PolyBot: a modular reconfigurable robot. *IEEE International Conference on Robotics and Automation*, 1, 2000.