Control of Locomotion with Shape Changing Wheels

Daniel Mellinger, Vijay Kumar, and Mark Yim

Abstract—We present a novel approach to controlling the locomotion of a wheel by changing its shape with 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 described. 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 and discrete system, and experimental results obtained from a physical modular robot illustrate the accuracy of the dynamic models and the validity of the approach.

I. INTRODUCTION

The study of the locomotion of reconfigurable modular robots has yielded a range of solutions. Snake-like locomotion, walking, and closed chain gaits have all been developed and are each potentially useful in different situations. Snake-like locomotion is probably best for navigating confined spaces while on flat terrain closed chain gaits are considered to be the most efficient as well as the fastest [1].

The simplest type of closed chain gaits are kinematic gaits that transition through a series of shapes which have been designed such that they move the robot in a desired way. In these kinematic gaits the inertia of the system plays no role and the motion of the system is completely determined by the shape change. Kinematic gaits are conceptually similar to statically stable walking gaits. Several researchers have implemented these kinematic gaits. In [4], a flexible rim with spokes made of shape memory alloy was deformed in such a way that it moved in a kinematic roll. In [3], 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 is necessary for locomotion. Dynamic gaits are conceptually similar to dynamic legged gaits like bipedal walking or running. In [5], 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 [2], Sastra, Chitta, and Yim 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 attempt to keep the center of mass in front of the contact point.

The design of the controllers for all these rolling robots has lacked a complete analysis of the dynamics because of the difficulty in modeling a large number of links and the various contact conditions [3]. Additionally, the large number of actuated joints creates a high-dimensional input space making the design of gaits for such systems difficult [2].

This difficulty motivates the idea of creating a low-dimensional abstract continuous model (ACM) of the discrete polygonal system (DPS) that consists of a smooth, closed curve and using it to derive gaits in the low-dimensional space for the purpose of mapping the control inputs onto high-dimensional DPS. In this paper we use a continuous deformable 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) move along the rim of the wheel. We call this abstraction the Tread-Controlled Ellipse. Note this constant-shape paradigm is similar to that seen in conventional circular wheels with the major difference that the locomotion is derived from the dynamics associated with the non-circular shape. The second approach involves the control of the shape of the wheel, by changing the axis lengths, to achieve locomotion. We call this abstraction the Shape-Controlled Ellipse. Note that these types of control input are not specific to the ellipse, Figure 1 illustrates tread-control and a type of shape-control for a different closed curve.

![Fig. 1. Tread-Control (top) and Shape-Control (bottom) Illustration](image)

Next we develop gaits for these two types of control input to locomote the deformable ellipse. 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 ACM for a rolling polygonal wheel is derived by considering the ellipse shown in 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. The unbarred unit vectors are an orthonormal basis set for the body-fixed reference frame whose origin is point O. The barred unit vectors are an orthonormal basis set for the earth-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 \( \theta \).

![Fig. 2. The ACM for a shape-changing, polygonal wheel.](Image)

We parameterize the curve by a coordinate \( \xi \) so that any generic point Q on the rim of the ellipse can be described in the body-fixed frame in terms of \( \hat{\mathbf{e}}_1 \) and \( \hat{\mathbf{e}}_2 \) by:

\[
\begin{align*}
\bar{x}_{CQ} &= a \cos(\xi) \\
\bar{y}_{CQ} &= b \sin(\xi)
\end{align*}
\]

(1)

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

B. Contact Point

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

\[
\begin{align*}
x_{CD} &= \frac{(b^2 - a^2) \sin \theta \cos \theta}{\sqrt{a^2 \sin^2 \theta + b^2 \cos^2 \theta}} \\
y_{CD} &= -\frac{a^2 \sin^2 \theta + b^2 \cos^2 \theta}{a \tan \theta}
\end{align*}
\]

(2)

(3)

(4)

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.

C. Rolling Assumptions

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 must be zero.

\[
\frac{dy_{OD}}{dt} = \frac{d^2 y_{OD}}{dt^2} = 0
\]

(5)

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 must be zero in the direction tangent to the contact surface. These assumptions are referred to as the zero sliding velocity assumption and the zero sliding acceleration assumption [6] and are written as

\[
\begin{align*}
\frac{dx_{OB}}{dt} &= \frac{d^2 x_{OB}}{dt^2} = 0
\end{align*}
\]

(6)

D. Equations of Motion

The free body diagram for the ellipse is moving in the positive \( \hat{\mathbf{e}}_1 \) direction is shown in Figure 3. The idealized rolling model does not incorporate any mechanisms for energy dissipation due to rolling resistance. Here we account for rolling resistance by letting the normal force \( (F_y) \) be offset from the actual contact point by a distance of \( \delta_r \) 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

\[
\begin{align*}
m \frac{d^2 x_{OC}}{dt^2} &= F_x \\
m \frac{d^2 y_{OC}}{dt^2} &= F_y - mg
\end{align*}
\]

(7)

(8)

\[
\frac{dH_C}{dt} = (x_{CD} + \delta_r \text{sign}(\frac{dx_{OC}}{dt})) F_y - y_{CD} F_x
\]

(9)

where \( m \) is the mass, \( g \) is the acceleration due to gravity, \( F \) is the contact force, and \( H_C \) is the angular momentum of the ellipse about the center of mass \( C \). The rolling assumptions

\(^1\)This assumption is validated later via simulations and experiments.
allow the contact forces to be eliminated and yield one equation of motion:

\[
\frac{dH_c}{dt} = (x_{CD} + \delta_r \text{sign}(\frac{dx_{OC}}{dt})(m \frac{d^2y_{OC}}{dt^2} + mg) - y_{CD}m \frac{d^2y_{OC}}{dt^2})
\]  

(10)

E. 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 4. 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 \(\frac{a_{mid}}{a} = \sigma \approx a_{mid} + \frac{1}{2}\) and \(\frac{b_{mid}}{b} = b \approx b_{mid} + \frac{1}{2}\). It assumed that the outer border of this system shown in Figure 4 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.

To find the total angular momentum of this system the angular momentum component due to a differential element represented by a rectangle as shown in Figure 4 is found. The position, \(r\), of this differential element’s center of mass with respect to the center of mass of the ellipse is

\[
r = a_{mid} \cos(\xi) \hat{e}_1 + b_{mid} \sin(\xi) \hat{e}_2
\]  

(11)

The angle, \(\gamma\), of a differential element satisfies the relationship

\[
tan(\gamma) = \frac{dy}{dx} = -\frac{a_{mid} \tan(\xi)}{b_{mid}}
\]  

(12)

The velocity, \(v\), of a differential element is

\[
v = \sigma_v t \hat{t} + \dot{\theta} \hat{e}_3 \times r + v_C,
\]  

(13)

where \(\hat{t}\) is the unit tangent vector at the element and \(\sigma_v t\) is the speed of the tread or the rate at which the material of the wheel rim (the center of mass of the differential element) moves in the body-fixed coordinate system. Notice that this tread speed is scaled by \(\sigma\). This is because \(v_t\) is the speed at the outer edge while the center of mass of the differential element moves along a different curve (shown solid in Figure 4). The angular velocity of a differential element is a sum of the components due to local rotation, \(\gamma\), (which is a consequence of tread velocity) and global rotation of the ellipse, \(\dot{\theta}\).

\[
H_C = \int r \times \mathbf{v} \, dm + \int (\gamma + \dot{\theta}) \, dI
\]

\[
= \int r \times (\sigma_v t \hat{t} + \dot{\theta} \hat{e}_3 \times r + v_C) \frac{m}{P} \, ds
\]

\[
+ \int (\dot{\theta} + \gamma) \frac{ml^2}{12P} \, ds
\]  

(14)

Substituting \(ds = \sqrt{a^2 \sin^2 \xi + b^2 \cos^2 \xi} \, d\xi\) and integrating over 0 to 2\(\pi\) yields:

\[
H_C = (I_{mid} + \frac{ml^2}{12}) \dot{\theta} + (\frac{2\pi ma_{mid} b_{mid}}{P} + \frac{\pi l^2 m}{6P}) v_t
\]  

(15)

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 we take derivatives of \(H_C\) is

\[
\frac{dH_c}{dt} = \frac{\partial H_c}{\partial t} \dot{t} + \frac{\partial H_c}{\partial \theta} \dot{\theta} + \frac{\partial H_c}{\partial v} \dot{v}_t
\]  

(16)

For the shape-controlled ellipse where \(p_t\) is fixed the derivative is

\[
\frac{dH_c}{dt} = \frac{\partial H_c}{\partial \theta} \dot{\theta} + \frac{\partial H_c}{\partial a} \dot{a} + \frac{\partial H_c}{\partial b} \dot{b}
\]  

(17)

F. 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 (10). For simplicity, we introduce the terms \(Y = -y_{CD}\) and \(X = -x_{CB}\). First we write expressions for \(y_{OC}\) and \(x_{OC}\):

\[
y_{OC} = y_{OD} - y_{CD}
\]

\[
= y_{OD} + Y
\]  

(18)

\[
x_{OC} = x_{OB} - x_{CB}
\]

\[
= x_{OB} - (acos(\xi_B)cos \theta - bsin(\xi_B)sin \theta)
\]

\[
= x_{OB} + X
\]  

(19)

1) Tread-Controlled Ellipse: We take derivatives of Equation 18 and recall from the rolling assumptions that derivatives of \(y_{OD}\) go to zero:

\[
\frac{dy_{OC}}{dt} = Y_{\theta} \ddot{\theta}
\]  

(20)

\[
\frac{d^2y_{OC}}{dt^2} = Y_{\theta \theta} \dot{\theta}^2 + Y_{\theta} \ddot{\theta}
\]  

(21)

Recall that the position of a generic point on the perimeter is parameterized by \(\xi\). 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 related to \(\dot{\xi}\). To find this relationship let us consider point \(B\) with the coordinate \(\xi_B\) as our generic point and the reference point on the rim to be denoted by \(\xi_B^*\). The arc length is given by:

\[
p_t = \int_{\xi_B}^{\xi_B^*} \sqrt{a^2 \sin^2 \xi + b^2 \cos^2 \xi} \, d\xi'
\]  

(22)
The magnitude of the tread or rim velocity, \( v_t \), is given by:

\[
v_t = \frac{d \rho}{dt} = \sqrt{a^2 \sin^2 \xi_B + b^2 \cos^2 \xi_B} = -X_{\xi_B} \dot{\xi}_B
\]  

(23)

Taking the second derivative yields the tread acceleration.

\[
\ddot{v}_t = -X_{\xi_B} \ddot{\xi}_B - X_{\xi_B} \dot{\xi}_B
\]  

(24)

We then take derivatives of Equation 22 and recall from the pure rolling assumption that derivatives of \( \dot{\xi}_B \) go to 0. Taking the derivative of \( x_{OC} \) and substituting the previously found relationship yields

\[
\frac{dx_{OC}}{dt} = X_{\theta} \dot{\theta} + X_{\xi_B} \dot{\xi}_B
\]  

\[
= X_{\theta} \dot{\theta} - v_t
\]  

(25)

\[
\frac{d^2 x_{OC}}{dt^2} = X_{\theta} \ddot{\theta}^2 + X_{\theta} \ddot{\theta} + X_{\xi_B} \ddot{\xi}_B + X_{\xi_B} \ddot{\xi}_B
\]  

\[
= X_{\theta} \ddot{\theta}^2 + X_{\theta} \ddot{\theta} - \dot{v}_t
\]  

(26)

Substituting expressions in Equations (16), (21), (25-26) into the equation of motion of the system (10) yields the acceleration of the wheel (\( \ddot{\theta} \)) in terms of the control input (\( \dot{v}_t \)):

\[
\ddot{\theta} = f(\theta, \dot{\theta}, a, b, \dot{v}_t) + g(\theta, \dot{\theta}, a, b, v_t) \dot{v}_t
\]  

(27)

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 keeping the perimeter constant. The perimeter, \( P \), is given by:

\[
P = \int_0^{2\pi} \sqrt{a^2 \sin^2 \xi + b^2 \cos^2 \xi} d\xi
\]  

(28)

By taking the first and second derivatives of this constant we can find a relationship between the rates of change of the shape variables \( a \) and \( b \):

\[
P_a \dot{a} + P_b \dot{b} = 0
\]  

(29)

\[
P_a a^2 + P_b b^2 + P_a \dot{a} + P_b \dot{b} + 2P_{ab} \dot{a} \dot{b} = 0
\]  

(30)

Note that there is an affine relationship between \( \ddot{a} \) and \( \ddot{b} \). We consider the control input to be \( \ddot{a} \) and let \( \ddot{b} \) change accordingly.

We now differentiate Equation (18) as before. Once again the derivatives of \( \mathcal{V}_{CD} \) go to zero but there are additional terms because \( a \) and \( b \) are not fixed. Taking derivatives yields:

\[
\frac{d x_{OC}}{dt} = Y_a \dot{a} + Y_b \dot{b} + Y_\theta \dot{\theta}
\]  

(31)

\[
\frac{d^2 x_{OC}}{dt^2} = Y_a \ddot{a}^2 + Y_b \ddot{b}^2 + Y_\theta \ddot{\theta}^2 + Y_a \ddot{a} + Y_b \ddot{b}
\]  

\[
+ Y_\theta \ddot{\theta} + 2Y_{ab} \ddot{a} \ddot{b} + 2Y_{ab} \ddot{b} \ddot{\theta}
\]  

(32)

In the \( \hat{e}_1 \) direction, the relationship between \( \dot{\xi}_B \) and \( \ddot{\xi}_B \) and the first and second derivatives of \( a \) must be found. To do this we 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 arc length between \( \xi = 0 \) and point \( B \).

\[
L = \int_0^{\xi_B} \sqrt{a^2 \sin^2 \xi + b^2 \cos^2 \xi} d\xi = \text{constant}
\]  

(33)

Taking derivatives of \( L \) yields an affine relationship between \( \dot{\xi} \) and \( \ddot{a} \) and an affine relationship between \( \dot{\xi} \) and \( \ddot{b} \). The first and second time derivative of Equation 19 for the shape-controlled system are

\[
\frac{dx_{OC}}{dt} = X_a \ddot{a} + X_b \ddot{b} + X_{\xi_B} \ddot{\xi}_B + X_\theta \dot{\theta}
\]  

(34)

\[
\frac{d^2 x_{OC}}{dt^2} = X_a \ddot{a} + X_b \ddot{b}^2 + X_{\xi_B} \ddot{\xi}_B^2 + X_\theta \ddot{\theta}^2 + X_a \ddot{a}
\]  

\[
+ X_b \ddot{b} + X_{\xi_B} \dddot{\xi}_B + X_\theta \dddot{\theta} + 2X_{ab} \dddot{a} + 2X_{ab} \dddot{b} X_{ab} \dddot{\xi}_B
\]  

\[
+ 2X_{aa} \dddot{a} \dddot{b} + 2X_{bc} \dddot{b} \dddot{\xi}_B + 2X_{bb} \dddot{b} \dddot{\theta} + 2X_{ab} \dddot{b} \dot{\theta}
\]  

(35)

We can then substitute the expressions in Equations (17), (21), (34-35) into the equation of motion of the system (10) to get an expression of the form:

\[
\ddot{\theta} = f(\theta, \dot{\theta}, a, b, \ddot{a}) + g(\theta, \dot{\theta}, a, b, \ddot{a}) \ddot{\theta}
\]  

(36)

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

III. GAIT FOR TREAD-CONTROLLED ELLIPSE

We use feedback linearization on Equation 27 by substituting the virtual control input \( u \) as \( \dot{v}_t = \frac{1}{2}(-f + u) \) to yield \( \ddot{\theta} = u \). A PD controller can now be used to make \( \theta \) track a desired trajectory, \( \theta_{des}(t) \):

\[
u = k \theta(\theta_{des} - \theta) + k \dot{\theta}(\dot{\theta}_{des} - \dot{\theta})
\]  

(37)

Consider the idea of having \( \theta \) 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 accelerating to the right. In a similar way if \( \theta \) is between \( -\frac{\pi}{2} \) and 0 then the ellipse must accelerating to the left. This simple idea of controlling to a constant angle is a good way to move the ellipse in either desired direction. 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 as shown in Figure 5. 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 complexity caused from 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 which necessitates some type of constraint on the control input, \( \ddot{a} \), so that \( a \) stays within limits. We discuss 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. A start-up phase is needed if this requirement is not met. Consider the resting ellipse shown in Figure 6a, note that this state can always be reached if there is some energy dissipation in the system. From this position the ellipse can be controlled to the reciprocal relationship. If the ellipse is made taller it will become unstable and tip over at some point in one direction or the other as shown in Figure 6c. After this instant a heuristic gait can take over and drive the ellipse in either desired direction.

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 7. In 7a, the ellipse starts at rest at an initial angle \( \theta_0 \) with some initial \( \frac{b}{a} \). The ellipse then rolls to the right through \( \theta = 0 \) in 7b and reaches an angle \( -\theta_0 \) in 7c, assuming no energy loss. The ellipse then rolls to the left but this time the ellipse is controlled to become circular around \( \theta_0 \) (7d) which adds energy to the system. Now the ellipse oscillates back to the left to some greater angle as shown in 7e. The process is repeated until the ellipse gains enough energy to tip over the left side and maintain a continuous roll.

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

\[
\dot{a} = k_p(a_{des} - a) - k_d\dot{a}
\]

(38)

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 \( \beta \). 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 8. Note that this gait is periodic in \( \pi \).

C. Double Pump Gait

The Double Pump gait is illustrated graphically in Figure 9. The first part of the Double Pump gait is equivalent to that of the Pump gait. As before, the ellipse is controlled to a circle around \( \theta = 0 \) in order to raise the center of mass of the system. Notice that Figures 9a-d are the same as Figures 7a-d. However, after leaving the region around \( \theta = 0 \) instead of being controlled back to the original \( \frac{b}{a} \) ratio the system is controlled to the reciprocal of the original \( \frac{b}{a} \) ratio as shown in 9e. In this way the center of mass is raised again and the ellipse continues to roll in the desired direction. Around \( \theta = \pi \) the ellipse is controlled to a circle to raise the center of mass (9f) and then finally back to the original \( \frac{b}{a} \) ratio after leaving the region around \( \theta = \pi \) (9g).

As with the Pump gait, the control input for this system is computed according (38). 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 $\frac{1}{\beta}$. These ranges for which these values are controlled by the chosen parameter $\chi$ and are shown in Figure 10. Note that this gait is also periodic in $\pi$.

V. MAPPING THE CONTINUOUS SYSTEM TO THE DISCRETE SYSTEM

To implement previously described gaits on the modular robots we develop a method for mapping the control inputs of the ACM to the control inputs of the DPS. In this section we first describe an algorithm for approximating a continuous curve with $n$ line segments and then its application to the mapping problem.

A. Algorithm

In general we consider the problem of taking a parametric curve defined by $x(\xi)$ and $y(\xi)$ for some range of $\xi$ from $\xi_i$ to $\xi_f$ 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. This implies that there are an infinite number of discrete approximations that can be found using this method by changing the starting point of the algorithm. This feature is utilized to create a map from the tread parameter to joint angles.

B. Joint Angles for the Tread-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, $\xi_i$, for the described algorithm is iteratively cycled completely around the ellipse from 0 to $2\pi$, shown in part in Figure 11. Note that there is some ambiguity as to how to measure $p_t$ for a given discrete approximation because each vertex travels a different distance. To alleviate this issue the distance each node has traveled along the ellipse is calculated and the average is taken in order to measure $p_t$ for a given discrete solution. After cycling through a range of $\xi_i$’s from 0 to $2\pi$ 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 $\psi_i$ can be written as functions of the form $\psi_i = \psi_i(p_t)$.

C. Joint Angles for the Shape-Control Input

The mapping from the axis lengths to joint angles is simpler than in the tread-controlled case. The starting point for 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 as shown in Figure 12. A polynomial curve can be fit to this data so that we can write $\psi_i = \psi_i \left( \frac{2}{a} \right)$.

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 attached file.
The ACM simulation is simply a numerical integration of the equations of motion in Matlab. The DPS simulation was performed in Gazebo [7]. 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 [8] shown in Figure 13. 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 was used for position sensing of the robot. This system consists of six cameras that shine LEDs onto the workspace who’s light is reflected back to the cameras by reflective markers fixed to the robot. The image location of the markers in each of the six cameras is combined to calculate a three dimensional world location of each marker. A computer then interfaces with the VICON system to receive this marker data from which \( \theta \) 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 shown in the Table I. The rolling resistance coefficient, \( \delta_r \), 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 \( \delta_r \) and the measured \( m \), \( l \), and \( P \) parameters and the best \( \delta_r \) was chosen. The parameters for the simulated models were matched to the parameters for the physical system.

### Table I

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>1.83 kg</td>
</tr>
<tr>
<td>( l )</td>
<td>0.060 m</td>
</tr>
<tr>
<td>( P )</td>
<td>0.869 m</td>
</tr>
<tr>
<td>( \delta_r )</td>
<td>0.009 m</td>
</tr>
</tbody>
</table>

#### 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. Note that maintaining this angle causes the system to accelerate in the positive x direction. Snapshots from the implementation of this gait on the CKBot are shown in Figure 14.

![Fig. 14. Road Runner Gait Snapshots](image)

Data from the experimental and simulation implementations of the Road Runner gait are shown in Figure 15. Notice that the ACM simulation is smooth and the angle is maintained exactly. This is 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. This is because the system is moving faster with increasing time which corresponds to a greater number of impacts which are also more forceful.

#### B. Pump Gait

The Pump gait was also implemented with \( \psi = \frac{\pi}{5} \), \( \beta = 0.5 \), \( k_p = 100 \frac{m}{s^2} \), and \( k_d = 20 \frac{m}{s} \) with initial conditions close to \( \theta_0 = 0.8 \text{ rad} \) and \( \dot{\theta}_0 = 0 \). Snapshots from the first part of the implementation of the Pump gait on the CKBot are shown in Figure 16. 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 first and last images of Figure 16.

![Fig. 13. Modular Loop Robot](image)
Data from the experimental and simulation implementations of the Pump gait are shown in Figure 17. The behavior of each of these systems is qualitatively equivalent and quantitatively similar. 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 are no impacts and also because there is no energy loss at the joints.

C. Double Pump Gait

The Double Pump gait was implemented with $\chi = \frac{\pi}{4}$, $\beta = 0.6$, $k_p = 100 \frac{1}{s^2}$, and $k_d = 20 \frac{1}{s}$. Snapshots from the initial part of the implementation of the Double Pump gait are shown in Figure 18. 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 19. Notice that the time for one complete revolution is close to 1.5 seconds in all three systems which corresponds to a speed of 0.58 m/s.

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 control input (tread-control and shape-control) and explicitly obtained the equations of motion of this shape-changing wheel. 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.

REFERENCES