

Rollbot: a Spherical Robot Driven by a Single Actuator

Jingxian Wang¹ and Michael Rubenstein¹

Abstract—Spherical robots typically require at least two actuators to achieve controlled 2D planar motion. Here we present Rollbot, the first spherical robot capable of controllably maneuvering on a 2D plane with a single actuator, challenging this assumption. Rollbot rolls on the ground in a circular pattern and controls its motion by changing the trajectory’s curvature by accelerating and decelerating its single motor and the attached mass according to our derived quasi-stable state dynamics and control laws. We present the theoretical analysis, design, and control of Rollbot, and demonstrate its ability to move in a controllable circular pattern and follow waypoints, validating the efficacy of the proposed theoretical framework.

I. INTRODUCTION

Underactuated robots have attracted researchers’ attention in recent years due to their advantages such as energy savings, material savings, and space savings [1]. The simplicity of underactuated robots is also often valued in swarm robotics [2]. While underactuated systems like inverted pendulums [3, 4], underactuated hands [5, 6], and underactuated wrists [7, 8] have been extensively studied, few mobile robots have adopted the idea of extreme underactuation.

Currently, only a handful of mobile robots can maneuver in 2D or 3D space with a single actuator. Robots in [9–11] use compliant mechanisms or one-way bearings to allow a single actuator to achieve multiple functions or modes of operation, effectively ‘multiplexing’ the actuator to move in space. Mugatu in [12] controls its movement by varying the step sizes of its left and right legs. Robots in [13–16] and drones in [2, 17, 18] move by rotating their bodies and accelerating their motors at appropriate times. Among these works, only [16] proposes a wheeled robot, which is also the only ground robot that does not move in discrete steps.

Another key inspiration for our work is spherical robots [19], which are studied for their structural simplicity and the inherent protective nature of their spherical shells that shields the internal mechanisms and electronics from hostile environments. Since the pioneering spherical robot [20], three major types of driven mechanisms for spherical robots have been proposed [19]. These include the most common barycentric type [21, 22], which moves masses attached to pendulums or tracks on the robot; the conservation of angular momentum (CoAM) type [23], which uses reaction wheels; and the shell deformation type [24], which changes the shape of the shell. Interestingly, to the authors’ knowledge, only one spherical robot with a single actuator has been reported [25], and it has not been validated in the real world. Even

¹Jingxian Wang and Michael Rubenstein are with the Center for Robotics and Biosystems, Northwestern University, Evanston, IL 60601, USA jingxianwang2026@u.northwestern.edu, rubenstein@northwestern.edu

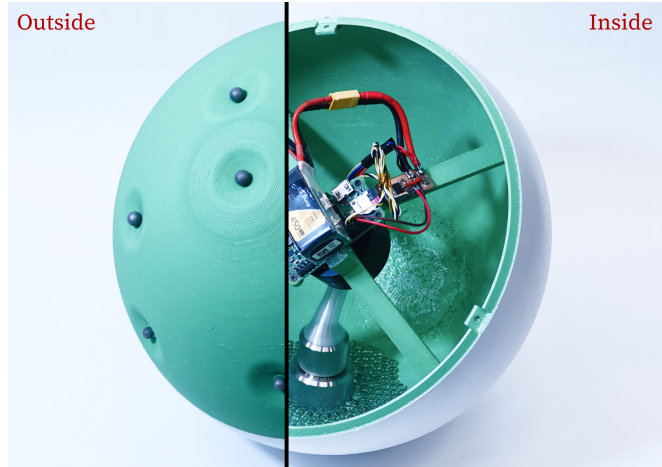


Fig. 1. Photo of inside and outside of Rollbot. Rollbot has an outer diameter of 24 cm and weighs 1.2 kg.

the authors of the review paper [19] claimed that “For a robot to be able to move in more than one direction, at least two (control) DoFs are needed.” Rollbot decisively challenges this assumption, demonstrating for the first time that a spherical robot can achieve precise and controllable 2D motion in the real world using a single actuator.

Furthermore, Rollbot’s design and movement algorithm can be used as a fail-safe for regular barycentric type spherical robots. If properly designed, regular spherical robots can downgrade to the configuration of the proposed robot when only one motor remains active, enabling continued operation despite the limited functionality. This feature is especially helpful as spherical robots typically work in hostile environments like on the Moon or on Mars, where a motor failure could end the mission.

The lack of research on single-actuator spherical robots, despite their potential benefits such as structural simplicity, cost savings, and applications in actuation failure scenarios and swarm robotics, motivates us to develop Rollbot, a barycentric spherical robot driven by a single motor. Rollbot is presented as a proof-of-concept platform demonstrating that precise planar locomotion of a spherical robot can be achieved using a single actuator. While the current implementation prioritizes concept validation over performance, future improvements in parameter selection, structural design, and advanced control can enable single actuator spherical robots to achieve performance comparable to that of more complex conventional spherical robots.

The notation and symbols used in this study are presented in Table. I.

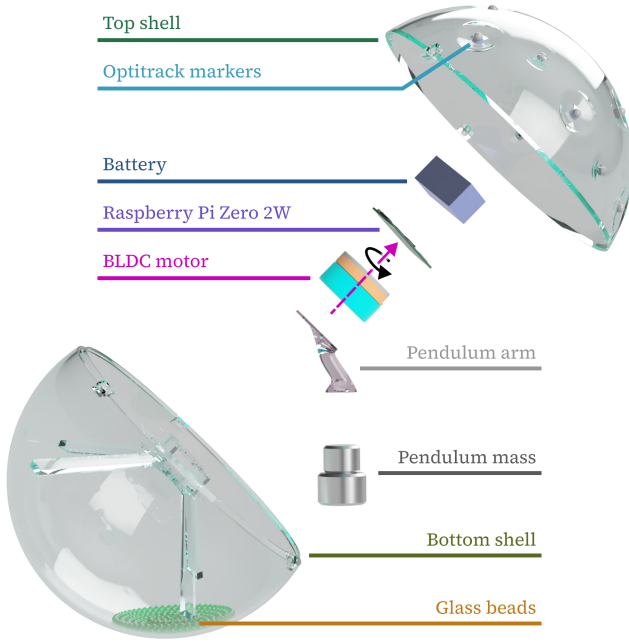


Fig. 2. An exploded view of Rollbot. The only actuator is the BLDC motor at the center, which can rotate around the purple axis.

Symbol	Definition
\vec{a}	Vector quantity
\hat{a}	Unit vector in the direction of \vec{a}
\hat{x}	Unit vector in the positive x -axis direction
$[\vec{a}]$	Skew-symmetric matrix representation of \vec{a}
\mathbf{T}	Matrix quantity
\mathbf{I}_3	3×3 identity matrix
\dot{a}	Time derivative of a
$\vec{a}\vec{c}$	Outer product of vectors \vec{a} and \vec{c}
a^p	Perturbed value of a
δa	Perturbation of a , defined as $\delta a = a^p - a$

TABLE I

NOTATION AND CONVENTIONS USED IN THIS PAPER.

II. ROBOT DESIGN

Rollbot, as shown in Fig. 1 and 2, is designed to test the feasibility of achieving 2D motion using only a single motor actuator. Rollbot is composed of a spherical shell and a pendulum connected to the shell through a motor. As the motor rotates the pendulum, the shell rolls on the ground, and we can control Rollbot's motion by changing the motor's rotation speed. For conciseness, we will refer to the spherical shell as 'the shell', the pendulum mass as 'the mass', and the motor's spinning speed as the 'driving speed' in subsequent discussions.

A. Dynamics of Rollbot

Rollbot can be considered as a spherical robot driven by an internal mass moving along a circular trajectory in the shell's reference frame. In this section, we will derive the dynamics of a general spherical robot driven by the movement of internal point masses and apply the results to Rollbot. We will start with a simple case where a spherically symmetric shell is driven by one internal point mass, as illustrated in Fig. 3. We also assume that the robot maintains contact with the ground and does not slip.

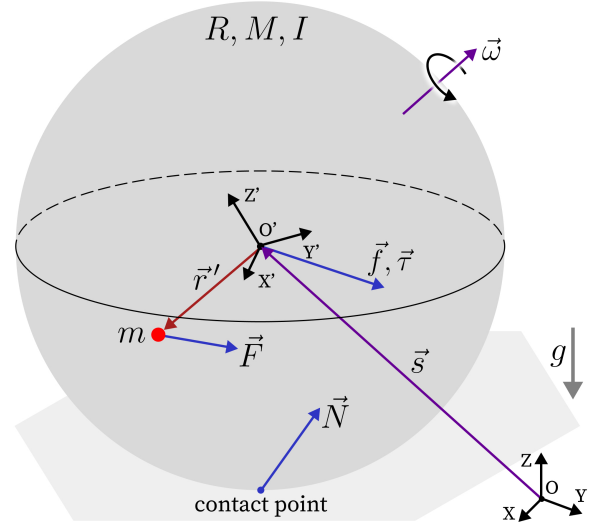


Fig. 3. Illustration of relevant quantities. O-XYZ is the ground reference frame G , O' - $X'Y'Z'$ is the shell's body reference frame B . \vec{s} is the position of the center of the shell in G , $\vec{\omega}$ is the angular velocity of the shell in G , and \vec{r}' is the position of the mass in B . R, M, I are the radius, mass, and the moment of inertia of the shell respectively, and m is the mass of the point mass. $\vec{f}, \vec{\tau}$ are the externally applied force and torque on the shell with respect to O' , \vec{N} is the sum of the normal and friction force the ground applies to the shell, and \vec{F} is the force the shell applies to the point mass.

The vector from the center of shell to the point mass $O'm$ in ground frame, \vec{r} , can be described using the rotation matrix \mathbf{T} from body frame to ground frame as

$$\vec{r} = \mathbf{T} \cdot \vec{r}' \quad (1)$$

We can take its first and second derivatives and obtain

$$\dot{\vec{r}} = \vec{\omega} \times (\mathbf{T} \cdot \vec{r}') + \mathbf{T} \cdot \dot{\vec{r}}' \quad (2)$$

$$\begin{aligned} \ddot{\vec{r}} &= \dot{\vec{\omega}} \times (\mathbf{T} \cdot \vec{r}') \\ &+ (\vec{\omega} \times (\vec{\omega} \times (\mathbf{T} \cdot \vec{r}')) + 2\vec{\omega} \times (\mathbf{T} \cdot \dot{\vec{r}}') + \mathbf{T} \cdot \ddot{\vec{r}}') \\ &= \dot{\vec{\omega}} \times (\mathbf{T} \cdot \vec{r}') + \vec{h}(\vec{\omega}, \mathbf{T}, \vec{r}'(t)) \end{aligned} \quad (3)$$

Notice that the term \vec{h} is the acceleration of point mass in the non-accelerating co-rotating frame, including the centrifugal, Coriolis, and relative acceleration terms but does not include acceleration of the shell.

Newton's second law and the angular momentum theorem yield the equations of motion of the point mass and the shell.

$$m(\ddot{\vec{s}} + \ddot{\vec{r}}) = \vec{F} - mg\hat{z} \quad (4)$$

$$M\ddot{\vec{s}} = -\vec{F} + \vec{N} - Mg\hat{z} + \vec{f} \quad (5)$$

$$I\dot{\vec{\omega}} = -(\vec{r} - \vec{s}) \times \vec{F} + (-R\hat{z}) \times \vec{N} + \vec{\tau} \quad (6)$$

We can eliminate \vec{F} and \vec{N} by combining the three equations

$$\vec{u} \times (\text{Eqn.4}) + R\hat{z} \times (\text{Eqn.5}) + (\text{Eqn.6}) \quad (7)$$

where $\vec{u} = R\hat{z} + \vec{r}$ is the vector from the contact point to the point mass and obtain

$$\begin{aligned} \vec{u} \times (m(\ddot{\vec{s}} + \ddot{\vec{r}})) + (R\hat{z}) \times (M\ddot{\vec{s}}) + I\dot{\vec{\omega}} \\ = \vec{u} \times (-mg\hat{z}) + R\hat{z} \times \vec{f} + \vec{\tau} \end{aligned} \quad (8)$$

Then we can apply the no slipping constraint

$$\dot{\vec{s}} = \vec{\omega} \times (R\hat{z}) \quad (9)$$

to (8) and substitute $\ddot{\vec{r}}$ using (3) to obtain the complete equation of motion of the shell as

$$\begin{aligned} & (m(\vec{u} \cdot \vec{u} \mathbf{I}_3 - \vec{u}\vec{u}) + MR^2(\mathbf{I}_3 - \hat{z}\hat{z}) + I\mathbf{I}_3) \cdot \dot{\vec{\omega}} \\ & = m\vec{u} \times (-\vec{h}(\vec{\omega}, \mathbf{T}, \vec{r}'(t)) - g\hat{z}) + R\hat{z} \times \vec{f} + \vec{\tau} \end{aligned} \quad (10)$$

The physical meaning of (10) is clear: the left hand side is the moment of inertia matrix of the whole robot with respect to the contact point applied on the angular acceleration. The right hand side is the torque applied on the robot with respect to the contact point contributed by the point mass's inertia in the non-accelerating co-rotating frame, the point mass's gravity, and the external force and torque applied on the shell.

We can also write down the sum of supporting and friction force

$$\vec{N} = (M + m)\ddot{\vec{s}} + m\ddot{\vec{r}} + (M + m)g\hat{z} - \vec{f} \quad (11)$$

The condition for the robot to not jump is the supporting force being positive, i.e.

$$\vec{N} \cdot \hat{z} > 0 \quad (12)$$

The condition for the robot to not slide is the friction force being no larger than the friction coefficient, μ , times the normal force, i.e.

$$\frac{\vec{N} \cdot \hat{z}}{\|\vec{N}\|} \geq \frac{1}{\sqrt{1 + \mu^2}} \quad (13)$$

The derived equation of motion can be extended to handle more general cases. Adding a summation over all terms related to the point mass extends the formulation to multiple point masses

$$\begin{aligned} & \left(\sum_q m_q (\vec{u}_q \cdot \vec{u}_q \mathbf{I}_3 - \vec{u}_q \vec{u}_q) + MR^2(\mathbf{I}_3 - \hat{z}\hat{z}) + I\mathbf{I}_3 \right) \cdot \dot{\vec{\omega}} \\ & = \sum_q m_q \vec{u}_q \times (-\vec{h}(\vec{\omega}, \mathbf{T}, \vec{r}'_q(t)) - g\hat{z}) + R\hat{z} \times \vec{f} + \vec{\tau} \end{aligned} \quad (14)$$

where m_q is the mass of q th point mass, \vec{r}'_q is the position of q th point mass in B , and \vec{u}_q is the vector from the contact point to q th point mass. Furthermore, by distributing extra point masses fixed to the shell or other point masses, (14) can also handle cases involving non-symmetrical mass distribution of the shell or general solid bodies moving inside the robot.

The result we obtained is equivalent to that proposed in [26], but the derivation is more straightforward and provides clearer physical insights.

When applying the model to Rollbot's case, a good approximation is to use (10) and let

$$\vec{r}' = r_0(\sin \phi \cos \theta(t) \hat{x} + \sin \phi \sin \theta(t) \hat{y} - \cos \phi \hat{z}) \quad (15)$$

$$\vec{\tau} = -k_0 \vec{\omega} \quad (16)$$

$$\vec{f} = \mathbf{0} \quad (17)$$

$$M = M_s + m_b \quad (18)$$

where $\theta(t)$ is the rotating angle of the pendulum mass around Z' axis, and k_0 is an experimentally determined damping constant. Such simplification is valid when:

- The shell has a moment of inertia matrix in the form of $I\mathbf{I}_3$.
- The pendulum mass's geometric size is small compared to both its distance from the center $|\vec{r}'|$ and the shell's radius R , so we can simplify the pendulum mass to a point mass.
- The damping on the shell's motion is small and has small impact on the behavior of Rollbot.

and we will show that our system satisfies these requirements in section II-D.

B. Quasi-stable State of Rollbot

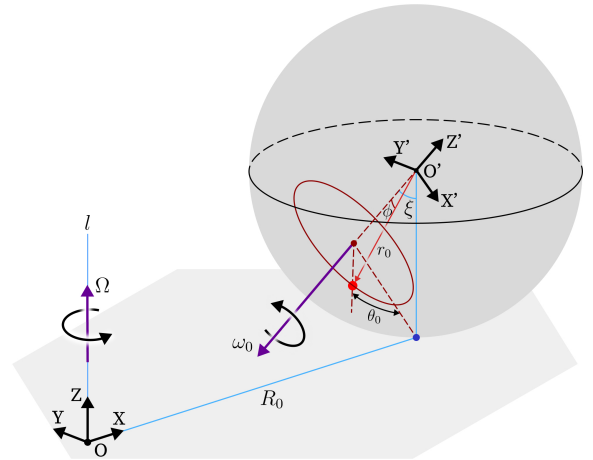


Fig. 4. Illustration of the quasi-stable state at $t = 0$. O-XYZ is the stationary ground reference frame G with its origin at the intersection of the revolving axis l and the ground plane, and X axis pointing toward the contact point. O'-X'Y'Z' is the shell's body reference frame B at this moment. Without loss of generality, we place Y' axis parallel to Y axis and Z' axis aligned with the rotation axis of the motor and on X-Z plane, and the angle between Z and Z' is ξ . The shell revolves around l at angular velocity $\Omega\hat{z}$, while also spinning at angular velocity $\vec{\omega}_s = -\omega_0\hat{z}'$ in the quasi-stable state. The shell's revolving radius is R_0 .

In our experiments, we slowly accelerate and decelerate the pendulum mass, keeping Rollbot close to the quasi-stable state where the driving speed $\theta'(t) = \omega_0$ is constant. In the quasi-stable state, Rollbot revolves around a vertical axis l . In the frame revolving together with the Rollbot, its state remains constant, as shown in Fig. 4.

Because the entire system revolves around l at constant angular velocity Ω , we have

$$\vec{u}(t) = \mathbf{T}_r(t) \cdot \vec{u}(0) \quad (19)$$

$$\vec{\omega}(t) = \mathbf{T}_r(t) \cdot \vec{\omega}(0) \quad (20)$$

where $\mathbf{T}_r(t)$ is the rotation matrix defined as

$$\mathbf{T}_r(t) = \begin{bmatrix} \cos(\Omega t) & -\sin(\Omega t) & 0 \\ \sin(\Omega t) & \cos(\Omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (21)$$

Using this property, we can then express (10) using only variables at $t = 0$, and subsequently reduce it into three equations about Ω , θ_0 , and ξ .

$$m\vec{u} \times (-g\vec{z} + \Omega^2(R_0\hat{x} + \vec{u} - (\vec{u} \cdot \hat{z})\hat{z}) + M\Omega^2 R_0 R \hat{y} - k_0(-\omega_0 \hat{z}' + \Omega \hat{z}) + I\Omega\omega_0 \sin \xi \hat{y} = \mathbf{0} \quad (22)$$

where $R_0 = \omega_0 R \sin \xi / \Omega$ and $\hat{z}' = \sin \xi \hat{x} + \cos \xi \hat{z}$. The quasi-stable state of Rollbot at different driving speed ω_0 can then be obtained by numerically solving (22). The solution with Rollbot's actual physical parameters is shown in Fig. 5.

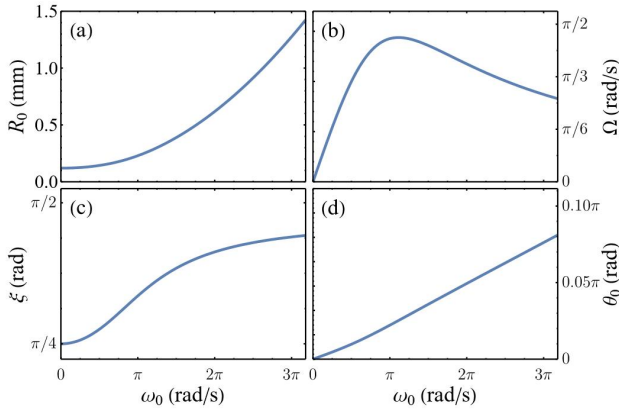


Fig. 5. Plots of key parameters of quasi-stable state versus driving speed ω_0 . (a) to (d) show the trends of revolving radius R_0 , revolving angular velocity Ω , tilting angle of rotating axis of the motor ξ and rotating angle of pendulum mass θ_0 respectively. These results are calculated using the specific physical parameters of the physical Rollbot prototype as detailed in Table II.

Fig. 5 shows that the revolving radius R_0 starts at $R \tan \phi$ when $\omega_0 = 0$, and gradually increases with ω_0 , suggesting that we can control the trajectory of Rollbot by changing the driving speed. ξ starts at ϕ when $\omega_0 = 0$ and gradually approaches $\pi/2$ rad/s, meaning that Rollbot moves in a nearly straight line when rolling at high speeds. θ_0 remains close to 0, meaning that the pendulum mass stays near the bottom of the shell when driving speed is smaller than 3π rad/s.

Furthermore, we also discovered that as long as damping coefficient k_0 is not excessively large, it has little impact on the results. This means that we can make assumption (16) for convenience of computation without significantly affecting the results.

Aside from the state shown above, there is also another set of 'fast revolving state', where θ_0 is close to π and Rollbot revolves in the opposite direction at a much smaller radius. However, because Rollbot moves quasi-stably in our experiments, this set of states will not occur.

C. Perturbation around Quasi-stable State

Perturbation around the quasi-stable state tells us how stable the quasi-stable state is and the response of Rollbot if we deviate from the state, either during acceleration

and deceleration or due to external disturbances. We will only consider the first order perturbation in the following discussion.

We can apply a perturbation to the rotation matrix \mathbf{T} and have

$$\mathbf{T}^p(t) = (1 + [\mathbf{T}_r(t) \cdot \vec{\alpha}(t)]) \cdot \mathbf{T}(t) \quad (23)$$

where $\vec{\alpha}$ is a small perturbation vector. According to the definition, we have

$$\begin{aligned} \delta\vec{u} &= \delta\vec{r}' = (\mathbf{T}_r \cdot \vec{\alpha}) \times \vec{r} \\ &= \mathbf{T}_r \cdot (\vec{\alpha} \times \vec{r}(0)) \end{aligned} \quad (24)$$

$$\begin{aligned} \delta\vec{\omega} &= (\mathbf{T}_r \cdot \vec{\alpha}) \times \vec{\omega} + d_t(\mathbf{T}_r \cdot \vec{\alpha}) \\ &= \mathbf{T}_r \cdot (\vec{\alpha} \times \vec{\omega}_s(0) + \dot{\vec{\alpha}}) \end{aligned} \quad (25)$$

Put (24) and (25) into (9) and we have

$$\delta\dot{\vec{s}} = \mathbf{T}_r \cdot ((\vec{\alpha} \times \vec{\omega}_s(0) + \dot{\vec{\alpha}}) \times (R\hat{z})) \quad (26)$$

Take perturbation of (8), left multiply by \mathbf{T}_r^{-1} and use (26) then we have

$$\begin{aligned} &m(\vec{\alpha} \times \vec{r}(0)) \times (\ddot{\vec{u}}(0) + \ddot{\vec{s}}(0)) \\ &+ m\vec{u}(0) \times \Omega^2(\vec{\alpha} \times \vec{r}(0) - ((\vec{\alpha} \times \vec{r}(0)) \cdot \hat{z})\hat{z}) \\ &+ m\vec{u}(0) \times (2\Omega\hat{z} \times (\vec{\alpha} \times \vec{r}(0)) + \vec{\alpha} \times \vec{r}(0)) \\ &+ (m\vec{u}(0) + MR\hat{z}) \times ((\vec{\alpha} \times \vec{\omega}_s(0) + \dot{\vec{\alpha}}) \times (R\hat{z})) \\ &+ I(\dot{\vec{\alpha}} \times \vec{\omega}_s(0) + \ddot{\vec{\alpha}}) \\ &+ mg(\vec{\alpha} \times \vec{r}(0)) \times \hat{z} \\ &+ k_0(\vec{\alpha} \times \vec{\omega}_s(0) + \dot{\vec{\alpha}}) = 0 \end{aligned} \quad (27)$$

Notice that this is a second order ordinary differential equation group of $\vec{\alpha}(t)$ and the parameters are not dependent on time, we can plug in the values obtained in section II-B and obtain an equation in the following form

$$\begin{pmatrix} \dot{\vec{\alpha}} \\ \ddot{\vec{\alpha}} \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{I}_3 \\ \mathbf{A}_{3 \times 3} & \mathbf{B}_{3 \times 3} \end{pmatrix} \cdot \begin{pmatrix} \vec{\alpha} \\ \dot{\vec{\alpha}} \end{pmatrix} \quad (28)$$

The recovery behavior of Rollbot under perturbation is characterized by eigenvalues of the matrix in (28).

In addition, notice that the system will be identical if we make a perturbation and let Rollbot revolve around l by a certain angle, we should remove the eigenvalue and eigenvector corresponding to this mode of movement when analyzing the stability. Finally, we can conclude that if all but one eigenvalues have real part smaller than 0, then Rollbot is stable under perturbation and the characteristic time of recovery τ is determined by the eigenvalue λ_2 with second smallest real part.

$$\tau \approx -\frac{1}{\text{Re}(\lambda_2)} \quad (29)$$

Plot of eigenvalues' distribution under different ω_0 with Rollbot's actual physical parameters is shown in Fig. 6, showing that with our choice of physical parameters given in Table. II, Rollbot is able to recover from perturbations

with a characteristic time of about 7 s. This is consistent with the result in direct numerical simulation as well as our experiments.

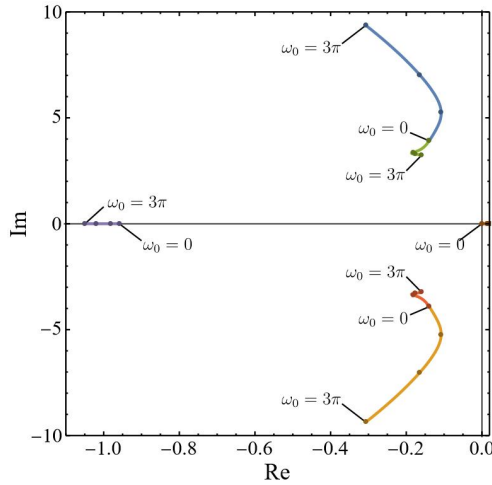


Fig. 6. Plot of eigenvalues' trajectory for $\omega_0 \in [0, 3\pi]$. Only one eigenvalue corresponding to revolution symmetry around l has positive real part, meaning such motion is stable. The second largest real part of all eigenvalues, $Re(\lambda_2)$, is within $[-0.1, -0.2]$, meaning the characteristic recovery time from transient behavior is around $\tau = 7$ s. These results are calculated using the specific physical parameters of the physical Rollbot prototype as detailed in Table II.

As far as the authors know, this is the first time quasi-stable state and perturbation behavior are analyzed for spherical robots.

D. Hardware Design

When designing the hardware, we want it to satisfy the following criterion:

- Rollbot should be as close to our theoretical model as possible, i.e. satisfying the criteria given in section II-A.
- Rollbot should be close to quasi-stable state even when accelerating/decelerating.
- Rollbot's trajectory should change significantly as we accelerate/decelerate the motor.

The first criteria require us to design the Rollbot shell's center of mass near the shell's geometric center, moment of inertia in the form of $I_0 \mathbf{I}_3$, and pendulum mass small enough to be considered as a point mass, and the second requires Rollbot to recover quickly from transient state, and the third requires revolving radius R_0 to change significantly when we change the driving speed ω_0 .

We designed the diameter of Rollbot's shell to be 24 cm, which is a few times larger than the scale of the battery and the motor, and concentrated the mass at the spherical surface of the shell. This makes sure that the moment of inertia is dominated by the surface of the shell instead of its internal structures, keeping the moment of inertia in the form of $I_0 \mathbf{I}_3$. Furthermore, we also carefully positioned the motor and the battery to keep the center of mass near the shell's geometric center. The pendulum mass is made of stainless steel and is small compared to the distance between it and the center of the shell as well as the size of the shell. We put $m_b = 40$ g of glass beads in the shell so the collision

between beads can introduce energy dissipation and create damping, making sure that the Rollbot can quickly recover from transient behaviors. Because the total mass of the beads is much smaller than the pendulum mass or the shell, and beads remain near the bottom of the shell, we can simplify the beads' effect to an addition of mass to the shell plus a linear damping given by (16).

With the analysis in section II-C, we found out that the system is more stable when r_0 , m , ϕ , and k_0 are larger. As such, we placed the pendulum mass as close to the shell as possible, and introduced the beads inside the shell. Experimentally, comparing to the case where no beads are added, Rollbot's quasi-stable dynamics is almost the same while exhibiting much less oscillation and much faster recovery after acceleration and deceleration.

With the analysis in section II-B, we can notice that the revolving radius R_0 changes more significantly with ω_0 when ϕ and m are smaller. This factor is conflicting with the stability factor discussed in the previous paragraph. We chose a balanced set of parameters shown in Table II, which allows Rollbot to vary its revolving radius from 0.12 m to 1.28 m when $\omega_0 \in [0, 3\pi]$ rad/s and recover from transients within a few seconds.

Quantity	Value	Explanation
R	0.12 m	Outer radius of shell
M	0.840 kg	Mass of shell
I	$0.0053 \text{ kg} \cdot \text{m}^2$	Moment of inertia of shell
ϕ	45°	Angle between the pendulum mass, center of shell and the axis of motor
r	0.093 m	Distance between pendulum mass's center of mass and center of shell
m	0.306 kg	Pendulum mass
m_b	0.040 kg	Total mass of beads
k_0	$(0.4 \text{ s}^{-1})MR^2$	Friction coefficient
M_{max}	$0.28 \text{ N} \cdot \text{m}$	Maximum rolling torque, computed using $M_{max} = r m g$

TABLE II

PHYSICAL PARAMETERS OF ROLLBOT.

E. Control Design

We already have a robot that can roll on the ground in circular pattern and theory derived in section II-B tells us that the revolving radius R_0 can be modulated by changing the driving speed ω_0 . In this section, we will derive a way of controlling Rollbot's 2D motion by changing R_0 at different times. An illustration of this process and relevant parameters are shown in Fig.7.

Instead of directly controlling the position \vec{s} of Rollbot, we control the center of curvature \vec{o} .

$$\vec{o} = \vec{s} - R_0(\cos \gamma \hat{x} + \sin \gamma \hat{y}) \quad (30)$$

When Rollbot is rolling with a constant driving speed ω_0 , it will move in a circle, and the center of curvature will be stationary. However, if we accelerate the driving speed at a rate β at $t = t_0$ when $\vec{s} - \vec{o}$ is at angle γ to \hat{x} , then the rate of change of curvature will be

$$|\dot{\vec{o}}| = v_{R_0} = \left. \frac{dR_0(\omega)}{d\omega} \right|_{\omega=\omega_0} \beta \quad (31)$$

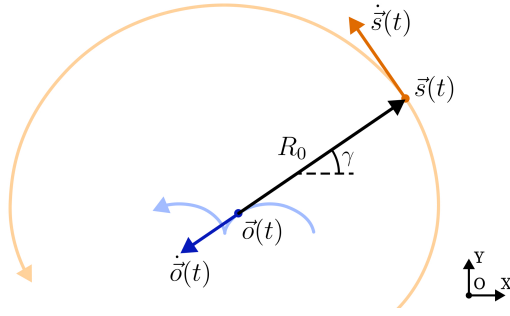


Fig. 7. Illustration of relevant parameters in the control algorithm. The orange and blue curves are the trajectory of the center of the shell and the corresponding center of curvature respectively. At time t , the center of shell is at $\vec{s}(t)$, the center of curvature is at $\vec{o}(t)$ and the curvature radius is R_0 . The center of shell is moving at velocity $\dot{\vec{s}}(t)$ along the curve perpendicular to $\vec{s} - \vec{o}$, and if Rollbot is accelerating, then the center of curvature will also move along $\vec{s} - \vec{o}$ at a rate of $\ddot{\vec{o}}(t)$

and subsequently the velocity of center of curvature will be

$$\dot{\vec{o}} = -v_{R_0}(\cos \gamma \hat{x} + \sin \gamma \hat{y}) \quad (32)$$

To move \vec{o} towards a certain target position \vec{o}_g , a simple solution can be

$$v_R = -k_p(\vec{o}_g - \vec{o}) \cdot (\cos \gamma \hat{x} + \sin \gamma \hat{y}) \quad (33)$$

where $k_p > 0$ is a constant. It is worth noting that in experiments, \vec{o} will not be the same as the actual center of curvature due to the transient behavior of Rollbot.

In reality, we need to consider the capability of the motor and various non-idealities. So, instead of using (33), we add a limit to the revolving radius to limit the driving speed, a limit to the maximum acceleration so Rollbot is always near quasi-stable state, and use PID instead of simple proportional control to compensate for external interference like the slope of the ground and to keep the system stable. The control diagram is shown in Fig. 8.

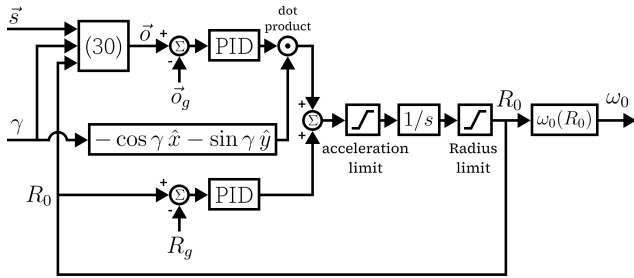


Fig. 8. Control diagram used in experiments. The input of the system are the current position \vec{s} and orientation γ of the shell provided by the Optitrack system and the output is the target driving speed ω_0 .

So far we have been talking about how to control \vec{o} instead of \vec{s} . In case we want the robot to move across a certain point \vec{s}_g at velocity $\dot{\vec{s}}_g$, we only need to compute the corresponding target R_g using equations obtained in section II-B and target \vec{o}_g using (30), and command Rollbot to roll around \vec{o}_g at revolving radius R_g . If we want to stop the Rollbot at \vec{s}_g , we will first let it revolve around \vec{o}_g and gradually stop the motor when Rollbot is approaching \vec{s}_g .

III. EXPERIMENTS

We conducted several experiments to demonstrate the ability of Rollbot. We used Optitrack system to provide Rollbot with its real-time position and orientation, but all other computations are executed on-board.

A. Open-loop Motion

In the open-loop motion experiments, Rollbot will maintain a constant driving speed from 0 to 7 rad/s. We collect the revolving radius R_0 , the rotating axis' tilting angle ξ , and the revolving angular velocity Ω of the Rollbot under different driving speeds ω_0 and compare it with the theoretical result obtained in section II-B to verify that our theory and approximation are consistent with reality. The results are shown in Fig. 9.

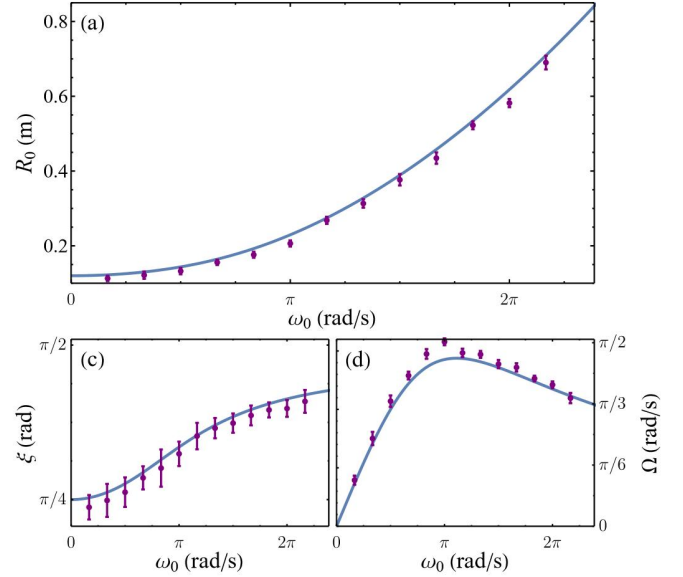


Fig. 9. Comparison between the theoretical and experimental results. Theoretical curves are drawn in blue lines and experimental results are drawn in purple points. The slight systematic difference between experimental data and theoretical prediction of R_0 and ξ (the difference is 6 mm and 1° respectively) are likely caused by the slight imbalance of the shell. The exact values of ξ were not measured since the pendulum is inside the opaque sphere. However, we confirmed qualitatively that it always remain near the bottom of the shell during operation.

B. Stable Circular Motion

In the stable circular motion experiments, Rollbot will try to circle around a set point with a given radius ranging from 20 cm to 65 cm. Fig. 10 shows how Rollbot can start from a point far away from the setpoint and move towards the goal, and Fig. 11 shows the trajectory of Rollbot when it is stably revolving around the set point. It is worth noting that the ground in our experiments was tilted by approximately 1° , which will cause Rollbot to drift rapidly from the target trajectories without feedback control. With the proposed feedback policy, however, Rollbot can successfully compensate for this bias.

From Fig. 10 we can see that Rollbot moves towards the set point by having higher driving speed when moving towards the set point and vice versa. In this example, Rollbot moves at an average speed of approximately 2 cm/s when

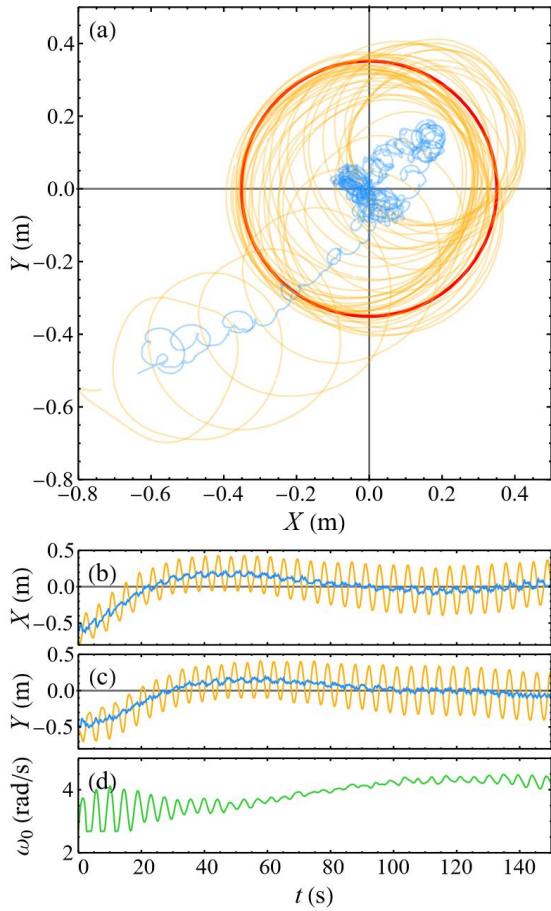


Fig. 10. Trajectory of Rollbot when moving towards the set point. (a) shows the trajectory, where the red circle shows the target radius $R_g = 0.35$ m, and the orange and blue lines are the trajectory of Rollbot and the $\vec{\sigma}$ respectively. (b), (c), and (d) shows the change of variables over time, where the orange and blue lines represent the position of Rollbot and the $\vec{\sigma}$ respectively and the green line represent the driving speed.



Fig. 11. Trajectory of Rollbot when stably revolving around the set point at $\{0,0\}$. Target revolving radius R_g is 0.20, 0.35, 0.50, 0.65 m for (a) to (d) respectively. The ground tilts slightly upward toward the top left.

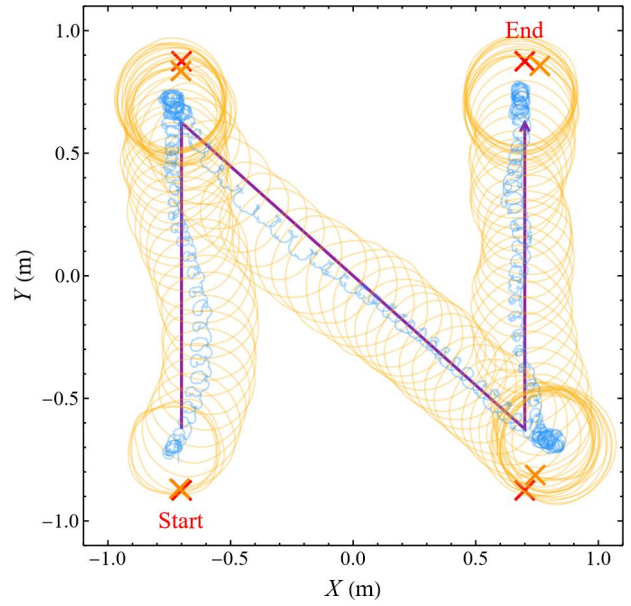


Fig. 12. Experimental results for waypoint movement experiment. Purple line shows the target trajectory of $\vec{\sigma}$, red and orange crosses are the target and actual stopping points.

moving towards the set point. Once it reached the stably revolving state, it will still modulate the driving speed slightly to compensate for the slope of the ground.

As shown in Fig. 11, Rollbot can revolve around the set point at the specified radius. During these tests, Rollbot is within 15 cm away from the expected trajectory, with an average deviation of 3.4 cm.

C. Moving between Waypoints

We would also like to demonstrate the Rollbot's ability to maneuver between waypoints, so we programmed Rollbot to move in a 'N' shape and stop at the four vertices. The results in Fig. 12 shows that Rollbot can successfully move towards waypoints and stop within 7 cm of the targets.

IV. DISCUSSION

The presented Rollbot prototype is primarily a proof-of-concept, focusing on demonstrating that precise planar locomotion is possible with a single actuator. In our experiments, we used an OptiTrack system for high-speed (100Hz sampling rate) global localization to simplify testing; in the future, we plan to switch to IMU-based sensing for fully onboard control. At present, Rollbot travels at about 2 cm/s, much slower than typical rolling robots, which can reach 10 cm/s to 1 m/s. This is largely due to our quasi-stable motion assumption and the resulting conservative control strategy. Future work will focus on drastically improving performance through better parameter tuning and advanced controllers that can take advantage of the robot's full dynamic capabilities. Another promising direction is adapting Rollbot for more challenging environments, such as sand, gravel, or even water surfaces, which will further expand the range of scenarios where single-actuator spherical robots can be useful. Aside from these possible movement enhancements, sensing and telemetry can also be equipped on Rollbot,

allowing it to execute a wider range of tasks autonomously. With these improvements, Rollbot can become a minimalist yet still powerful spherical robot, making it a great testbed for underactuated robotics and related fields.

Comparing with the other single-actuator spherical robot proposed in [25], Rollbot differs in two fundamental aspects. First, Rollbot has a more symmetric mass distribution, which significantly reduces the complexity of its dynamics. This simplicity allows for the derivation of closed-form equations of its quasi-stable state, which serve as the foundation for the described control law. Second, while the mechanism in [25] was demonstrated primarily in simulation, Rollbot is designed for operation in real world, which can greatly benefit from robustness. By operating within a quasi-stable regime, Rollbot trades peak locomotion speed for increased stability and resilience against environmental perturbations, such as surface friction variations and minor ground tilt.

V. CONCLUSION

In this paper, we presented Rollbot, a single actuator spherical robot capable of controlled rolling motion on the ground. We provided a general mechanical analysis of spherical robots driven by internal masses and the quasi-stable state and stability analysis of Rollbot. Based on our theory, we designed Rollbot's hardware and control algorithm and successfully achieved stable circular motion and waypoint following. Through experiments, we also verified our theory's accuracy.

We hope the design and analysis of Rollbot will inspire new minimalist robot designs, and the proposed control strategy can be used as a fail-safe for conventional spherical robots when they experience partial motor failure.

REFERENCES

- [1] Bin He et.al. "Underactuated Robotics: a Review". In: *IJARS* 16.4 (2019).
- [2] Jingxian Wang et.al. "A Single Motor Nano Aerial Vehicle with Novel Peer-to-Peer Communication and Sensing Mechanism". In: *RSS*. 2024.
- [3] Olfa Boubaker. "The Inverted Pendulum Benchmark in Nonlinear Control Theory: a Survey". In: *IJARS* 10.5 (2013).
- [4] Hongxing Li et.al. "Hardware Implementation of the Quadruple Inverted Pendulum with Single Motor". In: *Progress in Natural Science* 14.9 (2004).
- [5] Clement et.al. Gosselin. "An Anthropomorphic Underactuated Robotic Hand with 15 DOFs and a Single Actuator". In: *ICRA*. 2008.
- [6] Tadaaki et.al. Hasegawa. "Powerful and Dexterous Multi-finger Hand Using Dynamical Pulley Mechanism". In: *ICRA*. 2022.
- [7] Wu et.al. Licheng. "A Fully Rotational Joint Underactuated Finger Mechanism and Its Kinematics Analysis". In: *IJARS* 13.5 (2016).
- [8] Raffaele Di Gregorio. "Kinematic Analysis of the (nS)-2SPU Underactuated Parallel Wrist". In: *Journal of Mechanisms and Robotics* 4.3 (2012).
- [9] David Zarrouk and Ronald S Fearing. "Compliance-based Dynamic Steering for Hexapods". In: *IROS*. 2012.
- [10] Evandro et.al. Bernardes. "Modelling, Control and Simulation of a Single Rotor UAV with Swashplate-less Torque Modulation". In: *Aerospace Science and Technology* (2023).
- [11] Jianguo et.al. Zhao. "MSU Jumper: A Single-motor-actuated Miniature Steerable Jumping Robot". In: *IEEE T-RO* 29.3 (2013).
- [12] James et.al. Kyle. "The Simplest Walking Robot: A Bipedal Robot with One Actuator and Two Rigid Bodies". In: *IEEE Humanoids*. 2023.
- [13] Gedaliah Knizhnik and Mark Yim. "Thrust Direction Control of an Underactuated Oscillating Swimming Robot". In: *IROS*. 2021.
- [14] Andrew SaLoutos and Michael Rubenstein. "SpinBot: An Autonomous, Externally Actuated Robot for Swarm Applications". In: *IEEE DARS*. 2019.
- [15] Jingxian Wang and Michael Rubenstein. "PCBot: a Minimalist Robot Designed for Swarm Applications". In: *IROS*. 2022.
- [16] Satoshi et.al. Ito. "An Autonomous Mobile Robot with Passive Wheels Propelled by a Single Motor". In: 122 (2019).
- [17] Weixuan et.al. Zhang. "A Controllable Flying Vehicle with a Single Moving Part". In: *ICRA*. 2016.
- [18] Andrew G et.al. Curtis. "Autonomous 3D Position Control for a Safe Single Motor Micro Aerial Vehicle". In: *IEEE R-AL* (2023).
- [19] Aminata et.al. Diouf. "Spherical Rolling Robots - Design, Modeling, and Control: a Systematic Literature Review". In: *Robotics and Autonomous Systems* (2024).
- [20] Aarne et.al. Halme. "Motion Control of a Spherical Mobile Robot". In: *4th IEEE International Workshop on Advanced Motion Control*. Vol. 1. 1996.
- [21] Amir Homayoun Javadi A et.al. "Introducing August: a Novel Strategy for an Omnidirectional Spherical Rolling Robot". In: *ICRA*. Vol. 4. 2002.
- [22] Jiazhen et.al. Chen. "Design and Motion Control of a Spherical Robot with Control Moment Gyroscope". In: *3rd International Conference on Systems and Informatics*. 2016.
- [23] Shourov et.al. Bhattacharya. "Spherical Rolling Robot: A Design and Motion Planning Studies". In: *IEEE T-RO* 16.6 (2000).
- [24] Keith W et.al. Wait. "Self Locomotion of a Spherical Rolling Robot Using a Novel Deformable Pneumatic Method". In: *ICRA*. 2010.
- [25] Giacomo et.al. Spigler. "Intuitive Engineering as a Proxy Facilitates the Evolutionary Optimization of an Underactuated Robot Ball". In: *Workshop on Embodiment-Aware Robot Learning @ RSS 24*. 2024.
- [26] Vakhtang Putkaradze and Stuart Rogers. "On the Dynamics of a Rolling Ball Actuated by Internal Point Masses". In: *Meccanica* 53 (2018).