2024 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
July 15-19, 2024. Boston, MA, USA

Dynamic Posture Manipulation During Tumbling for Closed-Loop Heading Angle Control

Adarsh Salagame[1], Eric Sihite[2], Gunar Schirner[1], Alireza Ramezani[1]*

***Abstract*— Passive tumbling uses natural forces like gravity for efficient travel. But without an active means of control, passive tumblers must rely entirely on external forces. Northeastern University's COBRA is a snake robot that can morph into a ring, which employs passive tumbling to traverse down slopes. However, due to its articulated joints, it is also capable of dynamically altering its posture to manipulate the dynamics of the tumbling locomotion for active steering. This paper presents a modelling and control strategy based on collocation optimization for real-time steering of COBRA's tumbling locomotion. We validate our approach using Matlab simulations.**

I. Introduction

Rough terrain locomotion remains a major obstacle for mobile robots, prompting exploration of various design solutions. Of these, legged robots that intermittently interact with their environment to shift the center of mass are the most promising. Unlike wheeled systems with fixed contact points, legged systems can navigate by leveraging their contact-rich locomotion capabilities [1], [2]. However, negotiating steep slopes with bumpy surfaces presents distinct challenges that have yet to be fully explored.

In 2022, NASA's Innovative Advanced Concepts (NIAC) Program invited US academic institutions to participate in NASA's Breakthrough, Innovative, and Game-changing (BIG) Idea competition [3] by proposing novel mobility systems capable of traversing lunar craters. Northeastern University clinched NASA's top accolade, the Artemis Award, by proposing an articulated robot tumbler named COBRA (Crater Observing Bio-inspired Rolling Articulator) [4] shown in Fig. 1. The successful tumbling locomotion demonstrated impressive speed and efficiency over steep slopes in the hills near Pasadena, where the competitions were held. This spurred the formulation of this research question, focusing on active posture manipulation and heading angle steering.

This study specifically aims to elucidate the mathematical model underlying this locomotion challenge and explore control design approaches to address this tracking problem. Before delving into the technical discourse regarding modeling and proposed control design approaches, we briefly survey the literature and previous works pertaining to tumbling robot design.

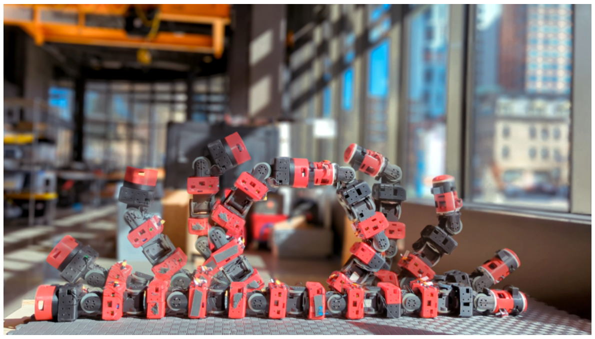

Fig. 1. Shows transformation from snake to tumbling configuration in COBRA platform

A. Tumbling Robots

Tumbling robots have been of interest within the robotic community. Passive tumbling systems, exemplified by the NASA/JPL Mars Tumbleweed Rover [5], offer minimal energy consumption, making them appealing for remote exploration missions where energy efficiency is paramount. On the other hand, active rolling spherical robots like MIT's Kickbot [6] boast a low center of gravity and omnidirectional movement capability, rendering them robust to external disturbances and adept at navigating uneven terrain. The ability of spherical robots to roll in any direction also grants them excellent maneuverability in confined spaces.

However, tumbling locomotion presents its own set of challenges. Passive rolling robots often sacrifice controllability in favor of energy efficiency, relying instead on their inherent morphology or posture for maneuvering. Additionally, rolling robots typically utilize their entire body for locomotion, posing difficulties for sensor placement such as cameras and consequently complicating localization and perception tasks.

Early designs, such as Rollo [7] and the Spherical Mobile Robot (SMR) [8], employed a spherical shell with a diametric spring-mass system to spin a driving wheel, generating movement through a mass imbalance. However, these designs faced reliability issues, particularly in maintaining constant contact between the driving wheel and the sphere. Furthermore, a substantial weight for the central structure was necessary to generate sufficient inertia to propel the system forward.

Subsequent robotic systems, like the University of Pisa's Sphericle [9] and Festo's Spider Rolling Robot (SRR) [10], replaced the diametric driving wheel with a car-driven system inside spherical structures. While these designs were

[1]This author is with the Department of Electrical and Computer Engineering, Northeastern University, Boston MA salagame.a, G.Schirner, *a.ramezani@northeastern.edu

[2]This author is with California Institute of Technology, Pasadena CA esihite@caltech.edu

*Indicates corresponding author.

979-8-3503-9154-1/24/$31.00 ©2024 IEEE

promising, they were still unstable from large perturbations on rough terrain.

Another approach to tumbling locomotion involves shifting weights inside a rigid spherical shell to precisely control the 3D position of the center of mass. Examples include the University of Michigan's Spherobot [11] and the University of Tehran's August Robot [12]. However, these systems require additional dead weights to control the center of mass, making them less energy-efficient.

A more energy-efficient method of positioning the center of gravity for tumbling involves using deformable structures. Successful examples include Ritsumeikan University's Deformable Robot [13] and Ourobot [14] with Articulated Closed Loop. COBRA also adopts the concept of structural deformation to initiate and control tumbling. However, what sets COBRA apart from these examples are its multi-modal locomotion abilities (sidewinding, slithering, and tumbling), field-tested capabilities for fast and dynamic tumbling locomotion on bumpy surfaces, a head-tail locking mechanism to form rugged structures for tumbling, and the ability to actively control posture for steering in two dimensions. This paper primarily focuses on tumbling locomotion and provides an overview of the head-tail locking concept.

B. Overview of COBRA Hardware

As seen in Fig. 1, the COBRA system consists of eleven actuated joints. The front of the robot consists of a head module containing the onboard computing of the system, a radio antenna for communicating with a lunar orbiter, and an inertial measurement unit (IMU) for navigation. At the tail end, there is an interchangeable payload module. The rest of the system consists of identical joint modules each containing a joint actuator and a battery. The robot has a latch mechanism in its head and tail that allows the formation of tumbling structures actively and based on commands telecommunicated to it wirelessly. Due to the large number of body joints, COBRA can actively adjust its posture to regulate its tumbling. This capability was put to the test in open-loop fashion in NASA BIG Idea competitions in 2022.

II. Reduced-Order Model (ROM) Derivations

In the tumbling configuration, COBRA is reduced to an elliptical ring, as depicted in Fig. 2. Throughout these derivations, we make the following assumptions:

- The ring has a negligible cross-section area (see Fig. 2).
- Mass distribution is uniform.
- The shape of the ring is parameterized by the lengths of its principal axes, u_1 and u_2, controlled by the actuation of joints in the robot (see Fig. 2).
- All achieved postures are symmetric, ensuring that the center of the ellipse aligns with the center of mass (CoM).

With these assumptions, we determine the inertia tensor of the ring $\mathcal{I}(u_i)$, manipulated by u_i. Subsequently, we utilize the inertia tensor to derive the nonholonomic equations of motion of the rolling ring, as the control actions manipulate its posture.

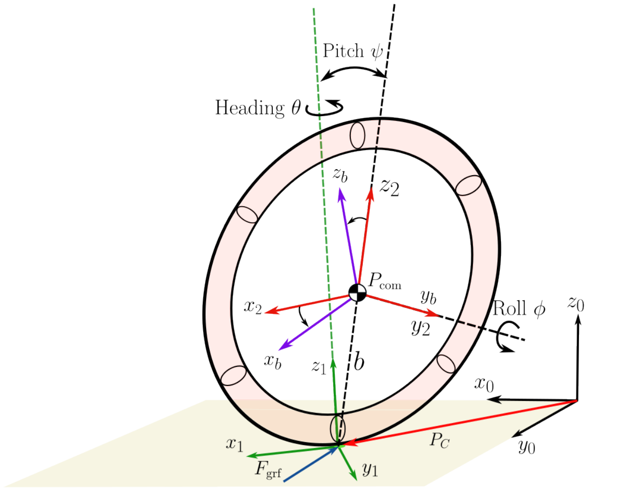

Fig. 2. Illustrates model parameters, coordinate frames, etc., used to derive Eqs. 1, 7, 12, and 13.

A. Cascade Model

In our approach to mathematically formulating tumbling servoing using shape manipulations in the ring, we present a cascade nonlinear model of the following form:

$$\begin{aligned} \Sigma_{tbl} &: \dot{x} = f(x, y) \\ \Sigma_{pos} &: \begin{cases} \dot{\xi} = f_\xi(\xi, u) \\ y = h_\xi(\xi) \end{cases} \end{aligned} \tag{1}$$

Here, x and y represent the state vector and output function, encapsulating the ring's orientation (Euler angles, CoM position) and mass moment of inertia about its body coordinate axes (x-y-z). The function $f(.)$ governs the dynamics for the states x.

Additionally, f_ξ, and h_ξ represent the dynamics governing the internal states ξ (see Section II). u denotes the actuation effort along the principal axes of the ring as shown in Figs. 2 and 3. The cascade model assumes separate governing dynamics for tumbling Σ_{tbl} and posture manipulation Σ_{pos}, considering a specialized case of tumbling that uses internal posture manipulation to maintain the shape of the robot with respect to the inertial frame. In the subsequent subsections, we derive the equations for each model.

B. Governing Dynamics for Posture Manipulation Σ_{pos}

As illustrated in Fig. 2, we consider the control actions $u = [u_1, u_2]^\top$ to adjust the length of the principal axes for the ring. Therefore, the objective of this section is to derive the governing equations for posture dynamics which is motivated by posture control from legged robotics [15]–[19].

While there is no closed-form equation for the perimeter of an ellipse, several approximations exist that are sufficiently accurate. One such approximation is employed in this work to maintain the perimeter at a fixed value as the length of the axes changes. However, for calculating the inertia tensor for this line mass, the following approach utilizing integrals provides a mathematically simpler method that can subsequently be computed through numerical integration.

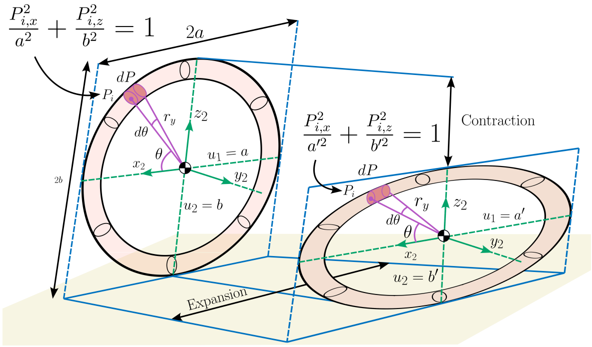

Fig. 3. Illustrates posture manipulation by considering two imaginary actuators, denoted as u_1 and u_2, which act along the principal axes of the ring to induce planar deformations.

Consider the general equation of the center line in the ring depicted in Fig. 3 in the x-z plane of the body frame with principal axes of length a and b,

$$p_{i,x}^2/a^2 + p_{i,z}^2/b^2 = 1 \tag{2}$$

where $p_i = [p_{i,x}, 0, p_{i,z}]^\top$ denotes the body-frame coordinates of a point on the ring. Consider the following change of variables:

$$\xi_1 = r_y C_\theta, \qquad \xi_2 = r_y S_\theta \tag{3}$$

where r_y and θ are polar coordinates and are shown in Fig. 3. S_θ and C_θ denote $\sin\theta$ and $\cos\theta$. We take the time-derivative of the equation above and Eq. 2, which yields

$$\begin{gathered}\dot\xi_1 = \dot r_y C_\theta - \xi_2\dot\theta, \qquad \dot\xi_2 = \dot r_y S_\theta + \xi_1\dot\theta \\ \frac{\xi_1\dot\xi_1}{a^2} - \frac{2\xi_1^2 u_1}{a^3} + \frac{\xi_2\dot\xi_2}{b^2} - \frac{2\xi_2^2 u_2}{b^3} = 0\end{gathered} \tag{4}$$

where $\dot a = u_1$ and $\dot b = u_2$. The perimeter of the ring is fixed and given by the following equation

$$P = \int_0^{2\pi} \sqrt{a^2 C_\theta^2 + b^2 S_\theta^2}\; d\theta \tag{5}$$

therefore, we can write the following relationship between $\dot P$ and $\dot\theta$

$$\dot P = \left(\sqrt{a^2 C_\theta^2 + b^2 S_\theta^2}\right)\dot\theta = 0 \tag{6}$$

This equation constitutes the remaining ordinary differential equations necessary to establish the state-space model for the posture dynamics. By defining $\xi_3 = r_y$, $\xi_4 = \theta$, $\xi_5 = a$, and $\xi_6 = b$, and considering Eqs. 2, 3, 4, 6, the state-space model governing the state vector $\xi = [\xi_1, \dots, \xi_6]^\top$ is given by

$$\begin{bmatrix} 1 & 0 & -C_{\xi_4} & \xi_2 & 0 & 0 \\ 1 & 0 & -S_{\xi_4} & \xi_1 & 0 & 0 \\ \frac{\xi_1}{\xi_5^2} & \frac{\xi_2}{\xi_6^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma(\xi) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot\xi_1 \\ \dot\xi_2 \\ \dot\xi_3 \\ \dot\xi_4 \\ \dot\xi_5 \\ \dot\xi_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{2\xi_1^2}{\xi_5^3} & \frac{2\xi_2^2}{\xi_6^3} \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{7}$$

where $\gamma(\xi) = \sqrt{\xi_5^2 C_{\xi_4}^2 + \xi_6^2 S_{\xi_4}^2}$. The matrix in the left-hand side of Eq. 7 is invertible, and therefore, the normal form $\dot\xi = f_\xi(\xi, u)$ can be obtained, which is skipped here. Now, it is possible to show that the mass moments of inertia about the body-frame x, y, and z axes, denoted by $\mathcal{I}_{xx}$, $\mathcal{I}_{yy}$, and $\mathcal{I}_{zz}$, are functions of the hidden state vector ξ.

The mass of the differential element on the ring can be calculated assuming uniform distribution as follows:

$$dm = \frac{m}{P}dP = \frac{m}{P}\gamma(\xi)d\xi_4 \tag{8}$$

where m is the total mass of the elliptical ring. Thus, the mass moment of inertia around each body frame axis can be obtained by:

$$\begin{gathered}\mathcal{I}_{kk} = \frac{m}{P}\int_{\xi_4} r_k^2\gamma(\xi)d\xi_4, \quad k \in \{x, y, z\} \\ r_x = \xi_3 C_{\xi_4}, \quad r_y = \xi_3, \quad r_z = \xi_3 S_{\xi_4}\end{gathered} \tag{9}$$

In the equation above, the output function $y = h_\xi(\xi) = [\mathcal{I}_{xx}, \mathcal{I}_{yy}, \mathcal{I}_{zz}]^\top$ encapsulates the mass moments of inertia. Next, we will derive the equations of motion for the tumbling ring using these posture dynamics as follows.

C. Governing dynamics for Tumbling Σ_{tbl}

Consider the ring shown in Fig. 2 with imaginary actuators u_i along its two principal axes for posture manipulation. We define the following coordinate frames of reference to describe the motion of the ring:

- world frame x_0-y_0-z_0,
- contact frame x_1-y_1-z_1 placed at the contact point p_c; its z-axis is perpendicular to the ground surface,
- gimbal frame x_2-y_2-z_2 placed at the CoM p_{cm}; it does not spin with the body,
- body frame x_b-y_b-z_b placed at the CoM p_{cm}; it spins with the ring.

We incline the contact frame from the world frame at an angle α to create an infinite inclined plane. The orientation R_b^0 of the ring (body coordinate frame x_b-y_b-z_b) is parameterized using roll, pitch, and yaw angles θ, ψ, and ϕ expressed as:

$$R_b^0 = R_z(\theta)R_y(\phi)R_x(\psi) \tag{10}$$

The body frame angular velocity vector $\boldsymbol{\omega_b} = [\omega_{b,x}, \omega_{b,y}, \omega_{b,z}]^\top$ expressed in terms of $\dot\theta$, $\dot\psi$, and $\dot\phi$ is given by:

$$\begin{aligned}\omega_{b,x} &= \dot\psi\sin(\theta)\sin(\phi) + \dot\theta\cos(\phi) \\ \omega_{b,y} &= \dot\psi\sin(\theta)\cos(\phi) - \dot\theta\sin(\phi), \\ \omega_{b,z} &= \dot\psi\cos(\theta) + \dot\phi\end{aligned} \tag{11}$$

From Σ_{pos}, the rings principal moment of inertia are given by y_1, y_2 and y_3. The angular momentum of the ring about p_{cm} is given by $\boldsymbol{H_b} = y^\top\boldsymbol{\omega_b}$. We define the radius of rotation as the vector from p_{cm} to p_c. Since the ring is pure rolling at the contact point p_c, there will be three constraints to consider, including a holonomic ($v_{c,z} = 0$) and two nonholonomic ($v_{c,x} = 0$) and ($v_{c,y} = 0$) constraints, where $\boldsymbol{v_c}$ denotes the contact velocity.

We can formulate the equations of motion by resolving the linear and angular momentum balance concerning the ring's CoM. The resulting equations of motion, derived from applying the balance laws alongside the non-integrable constraints, constitute a set of differential equations describing the ring's orientation and the lateral translation of its CoM over time. We can express this system of equations in the first-order form $\dot{x} = f(x,y) = M^{-1}(x,y)N(x,y)$ for numerical integration in MATLAB, given by:

$$M(x,y) = \left[\begin{array}{ccc|c} y_1 S_\theta & 0 & 0 & \\ 0 & y_2 + mr^2 & 0 & \mathbf{0} \\ \left(y_3 + mr^2\right) C_\theta & 0 & y_3 + mr^2 & \\ \hline & \mathbf{0} & & \mathbf{I} \end{array}\right] \tag{12}$$

where $\mathbf{0}$ and $\mathbf{I}$ are the zero and identity block matrices, respectively. The state vector is $x = [\theta, \psi, \phi, \dot{\theta}, \dot{\psi}, \dot{\phi}, p_{c,x}, p_{c,y}]^\top$ and r denotes the distance to the contact point. The nonlinear vector N is given by

$$N(x,y) = \begin{bmatrix} (y_3 - 2y_1)\,\dot{\psi}\dot{\theta}C_\theta + y_1\dot{\theta}\dot{\phi} \\ \beta_1 \\ \left(y_3 + 2mr^2\right)\dot{\psi}\dot{\theta}S_\theta \\ \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \\ \beta_2 \\ \beta_3 \end{bmatrix} \tag{13}$$

where β_i are defined in the appendix. This concludes the cascade model derivation. Next, we propose a controller based on the collocation method to utilize u in order to steer the heading angle of the tumbling ring.

III. Controls

To solve this controls problem, i.e., the ring's posture dynamics Σ_{pos} is recruited to regulate the heading angle in the tumbling dynamics Σ_{tbl}, we consider the following cost function given by

$$J = \|\theta - \theta_d\|_2 \tag{14}$$

where θ_d is the desired heading angle. The cost function J is governed by the cascade model given by Eq. 1.

We perform temporal (i.e., $t_i, \ i = 1, \ldots, n, \quad 0 \leq t_i \leq t_f$) discretization of Eq. 1 to obtain the following system of equations

$$\dot{x}_i = f_i(x_i, y_i), \quad i = 1, \ldots, n, \quad 0 \leq t_i \leq t_f \tag{15}$$

where x_i embodies the values of the state vector x at i-th discrete time. And, y_i embodies the principal mass moment of inertia of the ring at i-th discrete time. f_i denotes the governing dynamics of tumbling at i-th discrete time.

We stack all of the discrete values from x_i and y_i in the vectors $X = \left[x_1^\top(t_1), \ldots, x_n^\top(t_n)\right]^\top$ and $Y = \left[y_1^\top(t_1), \ldots, y_n^\top(t_n)\right]^\top$.

We consider $2n$ boundary conditions at the boundaries of n discrete periods to ensure continuity, given by

$$r_i\left(x(0), x\left(t_f\right), t_f\right) = 0, \quad i = 1, \ldots, 2n \tag{16}$$

Since we have 3 entries in y, we consider 3 inequality constraints to ensure the principal mass moment of inertia remains bounded. g_i is given by

$$g_i(x(t_i), y(t_i), t_i) \geq 0, \quad i = 1, 2, 3, \quad 0 \leq t_i \leq t_f \tag{17}$$

To approximate the nonlinear dynamics from the tumbling ring, we employ a method based on polynomial interpolations. This method extremely simplifies the computation efforts.

Consider the n time intervals, as defined previously and given by

$$0 = t_1 < t_2 < \ldots < t_n = t_f \tag{18}$$

We stack the states x_i and principal mass moment of inertia terms y_i from the ring at these discrete times into a single vector denoted by $\mathcal{Y}$ and form a decision parameter vector that the optimizer finds at once. Additionally, we append the final discrete time t_f as the last entry of $\mathcal{Y}$ so that tumbling speed too is determined by the optimizer.

$$\mathcal{Y} = \left[x_1^\top, \ldots, x_n^\top, y_1^\top, \ldots, y_n^\top, t_f\right]^\top \tag{19}$$

We approximate the output function $y_i(t_i)$ at time $t_i \leq t < t_{i+1}$ as the linear interpolation function $\tilde{y}$ between $y_i(t_i)$ and $y_{i+1}(t_{i+1})$ given by

$$\tilde{y} = y_i\left(t_i\right) + \frac{t - t_i}{t_{i+1} - t_i}\left(y_{i+1}\left(t_{i+1}\right) - y_i\left(t_i\right)\right) \tag{20}$$

We interpolate the states $x_i(t_i)$ and $x_{i+1}(t_{i+1})$ as well. However, we use a nonlinear cubic interpolation, which is continuously differentiable with $\dot{\tilde{x}}(s) = f(x(s), y(s), s)$ at $s = t_i$ and $s = t_{i+1}$.

To obtain $\tilde{x}$, we formulate the following system of equations:

$$\begin{aligned} \tilde{x}(t) &= \sum_{k=0}^{3} c_k^j \left(\frac{t - t_j}{h_j}\right)^k, \quad t_j \leq t < t_{j+1}, \\ c_0^j &= x\left(t_j\right), \qquad c_1^j = h_j f_j, \\ c_2^j &= -3x\left(t_j\right) - 2h_j f_j + 3x\left(t_{j+1}\right) - h_j f_{j+1}, \\ c_3^j &= 2x\left(t_j\right) + h_j f_j - 2x\left(t_{j+1}\right) + h_j f_{j+1}, \\ \text{where } f_j &:= f\left(x\left(t_j\right), y\left(t_j\right)\right), \quad h_j := t_{j+1} - t_j. \end{aligned} \tag{21}$$

The interpolation function $\tilde{x}$ utilized for x needs to fulfill the continuity at discrete points and at the midpoint of sample times. By examining Eq. 21, it is evident that the derivative terms at the boundaries t_i and t_{i+1} are satisfied. Hence, the only remaining constraints in the nonlinear programming problem are the collocation constraints at the midpoint of $t_i - t_{i+1}$ time intervals, the inequality constraints at t_i, and the constraints at t_1 and t_f, all of which are included in the optimization process. We address this optimization problem using MATLAB's fmincon function.

IV. Results and Discussions

We implemented the presented dynamic model in MATLAB and utilized the *ode45* solver to simulate the robot's behavior. The simulation began with the robot positioned

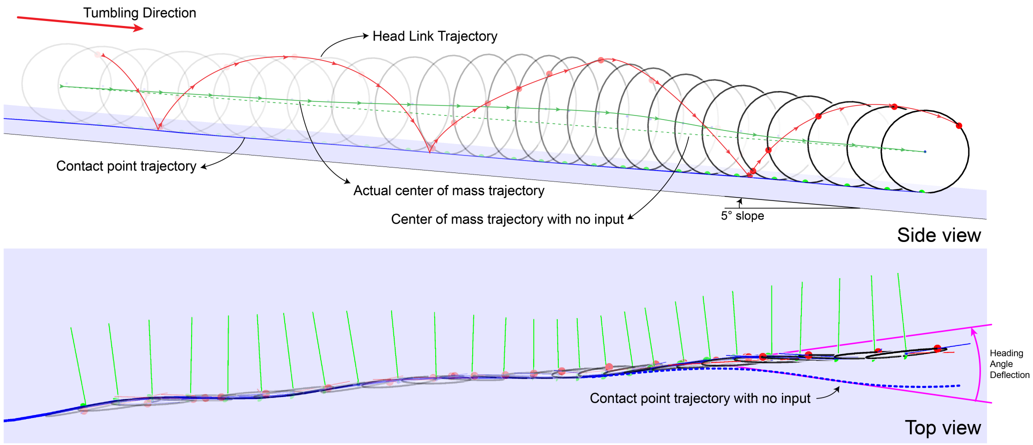

Fig. 4. Top figure illustrates the MATLAB simulation of the reduced order model of the robot rolling down a 5° incline with control input applied to change the length of the principle axes of the elliptical ring. The bottom figure shows the change in the path of tumbling as an impulse is applied to the axes lengths.

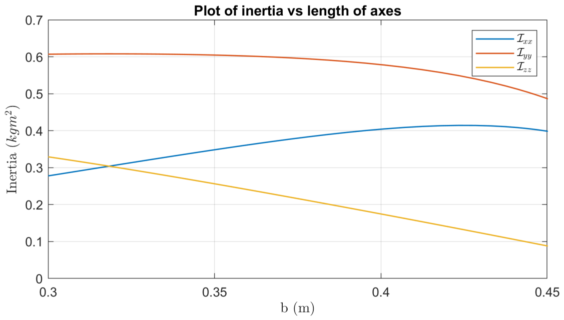

Fig. 5. Shows the change in inertia along the principle axes as the lengths of axes are manipulated.

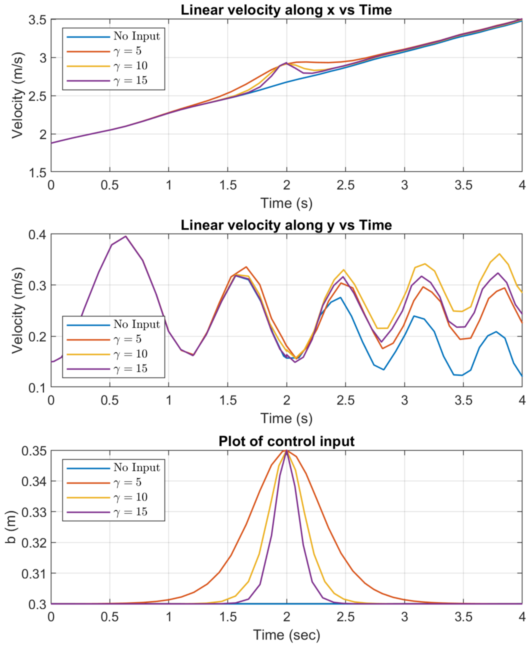

Fig. 6. Shows the impact of various impulse inputs on the velocity forward (along x) and lateral (along y) velocity of the tumbling structure

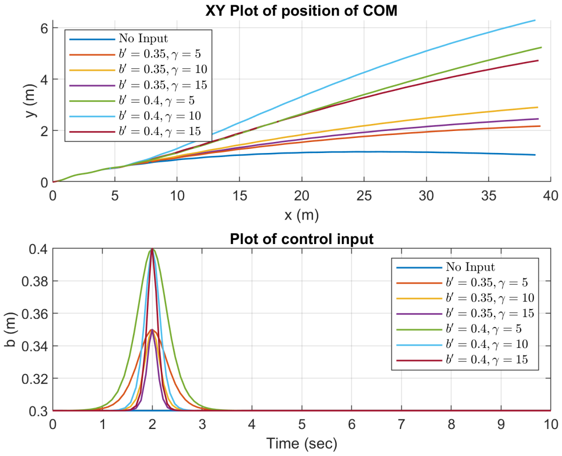

Fig. 7. Shows the deflection in the path of the tumbling ellipse from the no control input case upon application of various impulse inputs

on an infinite plane inclined at 5° about the inertial y axis with zero heading and pitch angles such that the direction of rolling is along the inertial x axis. Initial rolling and pitching velocities were set to 2π rad/s and 0.5 rad/s, respectively. After 2 seconds, an impulse was applied to alter the tumbling trajectory. Figure 4 depicts the simulation setup and showcases snapshots of the robot changing direction upon receiving the input.

The applied input is an impulse signal to the length of the ellipse's vertical principle axis b. The length of the horizontal principle axis a is dynamically calculated based on a fixed perimeter of 2 m, matching the length of the actual robot. The impulse signal is of the form of the form:

$$b(t) = 4b'\sigma(\gamma(t - t_0))(1 - \sigma(\gamma(t - t_0))) + b_0$$

where σ is the Sigmoid function parameterized by variables b', t_0 and γ representing the amplitude, time and sharpness of the impulse peak respectively, and b_0 refers to the zero input length of the axis b. Figure 5 shows how the inertia

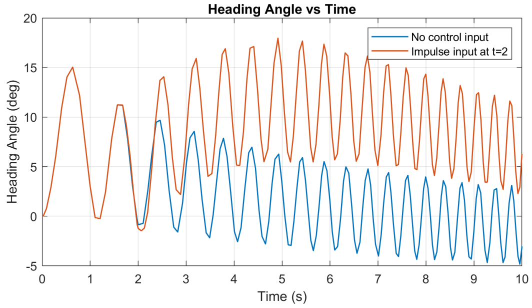

Fig. 8. Shows the change in heading angle from the no input case upon application of an impulse input with characteristics: $b' = 0.4\ m,\ t_0 = 2\ s,\ \gamma = 10$

of the tumbling structure varies as the axes lengths are changed, which contributes to the change in rolling dynamics during tumbling. Figure 6 shows the response of the tumbling dynamics to various impulse inputs. The linear velocity along x increases and decreases with the input, and lateral velocity along y increases as the heading is changed by the impulse. Figure 7 shows the xy plot of the center of mass of the robot as it tumbles, with the path changing as impulses of various amplitudes and sharpness are applied. The graph shows sharper impulses with higher amplitudes producing a larger deviation from the nominal trajectory with no control inputs applied. Figure 8 shows the corresponding change in heading angle for an input impulse at $t_0 = 2$, with $b' = 0.35$ and $\gamma = 10$.

V. Concluding Remarks

This study has established a control framework to steer a simplified model of COBRA during tumbling locomotion by dynamically manipulating its posture. We introduced a cascade model consisting of tumbling and posture dynamics and applied collocation to determine inputs to the cascade model. The simulation results presented herein demonstrate the effectiveness of our approach in adjusting COBRA's heading angle.

Future development will concentrate on integrating this control strategy into a closed-loop system, facilitating real-time steering of the physical robot. Hardware implementation may involve discretizing the simplified model and utilizing COBRA's 11 articulated joints to achieve desired postures. However, this necessitates reliable continuous ground contact estimation to effectively execute posture control. Successful contact estimation and application of the proposed control approach outlined in this paper will enable agile and controllable tumbling locomotion of COBRA in desired directions in real-world scenarios.

VI. Appendix

The nonlinear terms β_i from Eq. 13 are given by

$$\begin{aligned}
\beta_1 &= \left(y_1 - y_3 - mr^2\right)\dot{\psi}^2 S_\theta C_\theta \\
&\quad - \left(y_3 + mr^2\right)\dot{\psi}\dot{\phi} S_\theta - mgrC_\theta \\
\beta_2 &= -r\dot{\psi}C_\psi C_\theta + r\dot{\theta}S_\psi S_\theta - r\dot{\phi}C_\psi \\
\beta_3 &= -r\dot{\psi}S_\psi C_\theta - r\dot{\theta}C_\psi S_\theta - r\dot{\phi}S_\psi
\end{aligned}$$

References

[1] H.-W. Park, A. Ramezani, and J. W. Grizzle, "A Finite-State Machine for Accommodating Unexpected Large Ground-Height Variations in Bipedal Robot Walking," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 331–345, Apr. 2013.

[2] A. Ramezani and J. Grizzle, "Atrias 2.0, a new 3d bipedal robotic walker and runner," in *Adaptive Mobile Robotics*, WORLD SCIENTIFIC, May 2012, pp. 467–474.

[3] *Big Idea — NASA's Breakthrough, Innovative, and Game-changing (BIG) Idea Challenge*, en-US. [Online]. Available: https://bigidea.nianet.org/.

[4] A. Salagame, N. Bhattachan, A. Caetano, *et al.*, *How Strong a Kick Should be to Topple Northeastern's Tumbling Robot?* arXiv:2311.14878 [cs, eess], Nov. 2023. DOI: 10.48550/arXiv.2311.14878. [Online]. Available: http://arxiv.org/abs/2311.14878 (visited on 12/02/2023).

[5] A. Behar, J. Matthews, F. Carsey, and J. Jones, "NASA/JPL Tumbleweed polar rover," in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, vol. 1, Mar. 2004, 395 Vol.1.

[6] C. Batten and D. Wentzla, "Kickbot: A Spherical Autonomous Robot," en,

[7] A. Halme, T. Schonberg, and Y. Wang, "Motion control of a spherical mobile robot," in *Proceedings of 4th IEEE International Workshop on Advanced Motion Control - AMC '96 - MIE*, vol. 1, Mar. 1996, 259–264 vol.1.

[8] G. Reina, M. Foglia, A. Milella, and A. Gentile, "Rough-terrain traversability for a cylindrical shaped mobile robot," in *Proceedings of the IEEE International Conference on Mechatronics, 2004. ICM '04.*, Jun. 2004, pp. 148–153.

[9] A. Bicchi, A. Balluchi, D. Prattichizzo, and A. Gorelli, "Introducing the "SPHERICLE": An experimental testbed for research and teaching in nonholonomy," in *Proceedings of International Conference on Robotics and Automation*, vol. 3, Apr. 1997, 2620–2625 vol.3.

[10] A. Western, M. Haghshenas-Jaryani, and M. Hassanalian, "Golden wheel spider-inspired rolling robots for planetary exploration," en, *Acta Astronautica*, vol. 204, pp. 34–48, Mar. 2023.

[11] R. Mukherjee, M. Minor, and J. Pukrushpan, "Simple motion planning strategies for spherobot: A spherical mobile robot," in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, vol. 3, Dec. 1999, 2132–2137 vol.3.

[12] A. H. Javadi A. and P. Mojabi, "Introducing Glory: A Novel Strategy for an Omnidirectional Spherical Rolling Robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, no. 3, pp. 678–683, Dec. 2004.

[13] Y. Sugiyama and S. Hirai, "Crawling and Jumping by a Deformable Robot," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 603–620, May 2006.

[14] J. Paskarbeit, S. Beyer, M. Engel, A. Gucze, J. Schröder, and A. Schneider, "Ourobot—A sensorized closed-kinematic-chain robot for shape-adaptive rolling in rough terrain," en, *Robotics and Autonomous Systems*, vol. 140, p. 103 715, Jun. 2021.

[15] P. Dangol, A. Ramezani, and N. Jalili, "Performance satisfaction in Midget, a thruster-assisted bipedal robot," in *2020 American Control Conference (ACC)*, Jul. 2020, pp. 3217–3223.

[16] P. Dangol and A. Ramezani, "Feedback design for Harpy: A test bed to inspect thruster-assisted legged locomotion," in *Unmanned Systems Technology XXII*, vol. 11425, SPIE, May 2020, pp. 49–55.

[17] E. Sihite, A. Kalantari, R. Nemovi, A. Ramezani, and M. Gharib, "Multi-Modal Mobility Morphobot (M4) with appendage repurposing for locomotion plasticity enhancement," en, *Nature Communications*, vol. 14, no. 1, p. 3323, Jun. 2023.

[18] E. Sihite, P. Dangol, and A. Ramezani, "Optimization-free Ground Contact Force Constraint Satisfaction in Quadrupedal Locomotion," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Dec. 2021, pp. 713–719.

[19] P. Dangol and A. Ramezani, *Reduced-Order-Model-Based Feedback Design for Thruster-Assisted Legged Locomotion*, arXiv:2105.10082 [cs], May 2021. DOI: 10.48550/arXiv.2105.10082. [Online]. Available: http://arxiv.org/abs/2105.10082 (visited on 12/14/2023).