2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

CoM-Jacobian-Based Locomotion Control of an Annular Pneumatic Robot

Ghita Tazeroualt Tlamçani†, Seanan Vilfroy†, Rafael Cisneros-Limón†*, and Hiroshi Kaminaga†

***Abstract*— Inflatable habitat modules are good options for space exploration because they are light and easy to pack. However, controlling their shape and movement is difficult due to their soft materials and complex behavior. This work presents a rigid analog platform designed to capture the dominant dynamic behavior of such modules and allowing to design a model-based control represented by a second-order inverse-Jacobian control combined with null-space optimization. This control strategy allows the robot to maintain a shape leaning toward a circular shape while rolling in a controlled direction towards a target, improving rolling speed compared to traditional sequential control methods without compromising structural integrity. Simulations show significant improvements in speed and demonstrate the importance of maintaining the near-circular shape for enhanced performance. This approach provides a foundation for more advanced control schemes applicable to soft robotic systems.**

***Index Terms*— pneumatic actuation, rolling locomotion, inverse Jacobian control, null-space optimization, modular robots**

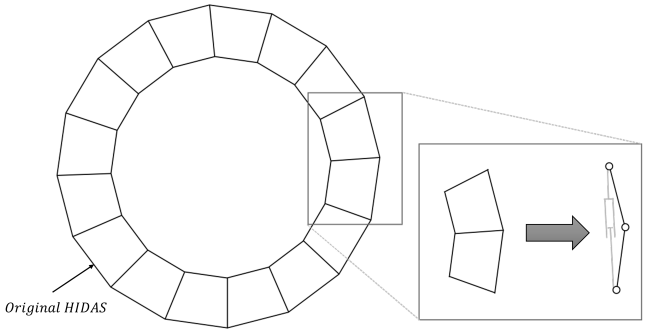

Fig. 1. Design simplification from HIDAS inflatable module to a kinematic chain composed by rigid segments driven by pneumatic actuators (the annular pneumatic robot).

I. Introduction

Space exploration has been of interest for humans for a long time, and the creation of moon bases continues this enduring fascination [1]. This work is conducted as part of the Japanese Science and Technology Agency's Moonshot Research and Development Program, an ambitious initiative aimed at achieving breakthrough innovations for society by 2050. One goal of the program is to enable a sustainable human presence in space using advanced robotic systems. Within the space exploration domain of this program, multiple robotic platforms are being developed to address different aspects of lunar habitat deployment and maintenance[1]. HIDAS (Homeostatic Inflatable Decentralized Autonomous Structure) represents one of such platforms, specifically designed to serve as an autonomous habitat module that can transport itself across lunar terrain and assemble into larger structures. This project brings up several inquiries regarding transportation, since the habitats must be lightweight, compact, and able to assemble autonomously.

To transport these modules efficiently, HIDAS must be equipped with robotic capabilities that allow it to move autonomously and reliably across rugged and uneven ground. Conventional wheeled robots are highly effective for energy-efficient, rapid movement on smooth terrains. Nonetheless, their performance declines sharply on uneven surfaces. To address this limitation, researchers have developed wheel-like segmented robots composed of multiple interconnected segments. These robots are capable of morphing their shape to facilitate traversal over obstacles with greater ease [2]. Some of these systems employ sequential control methods that rotate segments until they make ground contact in the intended direction, allowing them to navigate difficult terrain by altering their overall shape. However, these alterations in shape frequently diminish the benefits of a genuine wheel form, restricting speed and maneuverability [2].

Simultaneously, progress in soft robotics has shown that flexible robots equipped with internal power sources can attain versatile movement by altering their shapes, improving mobility in chaotic surroundings [3]. Moreover, robots featuring deformable outer surfaces have been created to adjust rolling dynamics; nevertheless, these shapes are frequently elliptical and restrict the ability to navigate obstacles [4]. Recent advancements in rolling robots utilizing flexible materials and compact actuators have enhanced shape flexibility for rolling movement. Yet, managing these robots continues to be difficult because of the heightened intricacy of their movements [5], [6].

Building on these foundations, our work investigates pneumatic actuation in a segmented annular robot, a ring-shaped robot composed of sixteen rigid links connected by joints forming a closed loop, designed to emulate the dynamic behavior of the inflatable version. We aim to achieve rapid

†CNRS-AIST JRL (Joint Robotics Laboratory), IRL3218, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. Emails: ghita.tazeroualt@ensta.fr, seanan.vilfroy@ensam.eu, {rafael.cisneros, hiroshi.kaminaga}@aist.go.jp.

*Corresponding author.

[1]MoonBot: Modular and On-demand Reconfigurable Robot – Moon Base Construction Demo in JAXA [Video]. YouTube. https://www.youtube.com/watch?v=TilwjMgt0ek.

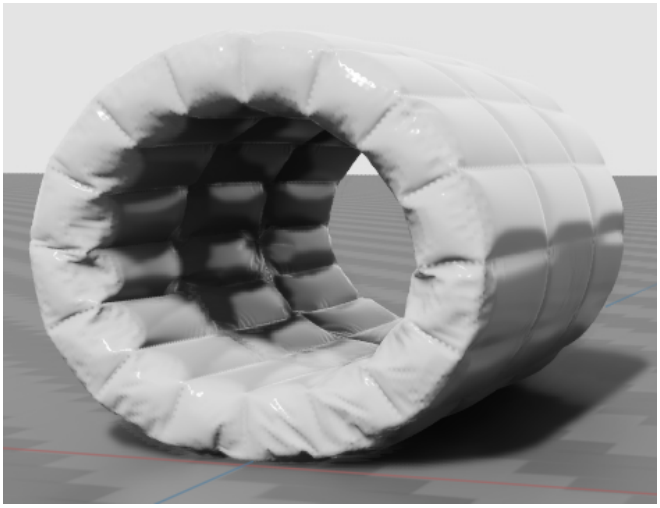

Fig. 2. The original HIDAS robot.

and stable rolling locomotion through a second-order inverse Jacobian-based control combined with null-space optimization. This approach seeks to maintain the robot's circular shape—which facilitates efficient rolling—while applying it to a segmented structure. By preserving shape integrity and enhancing rolling speed, this method addresses prior control challenges, facilitating autonomous alignment and deployment of habitat modules in space environments.

II. Design and Modeling of a Pneumatic Annular Robot

A. Design Strategy and Simplification of HIDAS

HIDAS is an inflatable annular robot composed of three longitudinal rings, each consisting of sixteen modules arranged circumferentially (Fig. 2). It achieves forward locomotion through the sequential inflation and deflation of its modules. However, the complex nonlinear interactions between the soft structure, internal air dynamics, and ground contact make it extremely difficult to derive an accurate analytical model of the system [1].

To address this challenge, we developed a simplified rigid-segment robot that preserves HIDAS's modular actuation concept in the motion plane while allowing straightforward implementation of model-based control. A key design decision was to retain pneumatic actuation in order to reproduce the actuation profile to control HIDAS' inflation/deflation process. We therefore employed springless single-acting pneumatic actuators, which behave analogously to inflatable chambers (Fig. 1): applying pressure causes extension, while releasing pressure through the exhaust results in passive retraction without employing restoring spring-like forces.

B. Mechanical Architecture

The simplified robot is composed of sixteen rigid links connected in a closed loop by rotational joints (Fig. 3) [7]. Each link and joint can change its stiffness using pneumatic actuators [8], allowing the robot to dynamically adjust its shape to maintain stability and improve rolling performance.

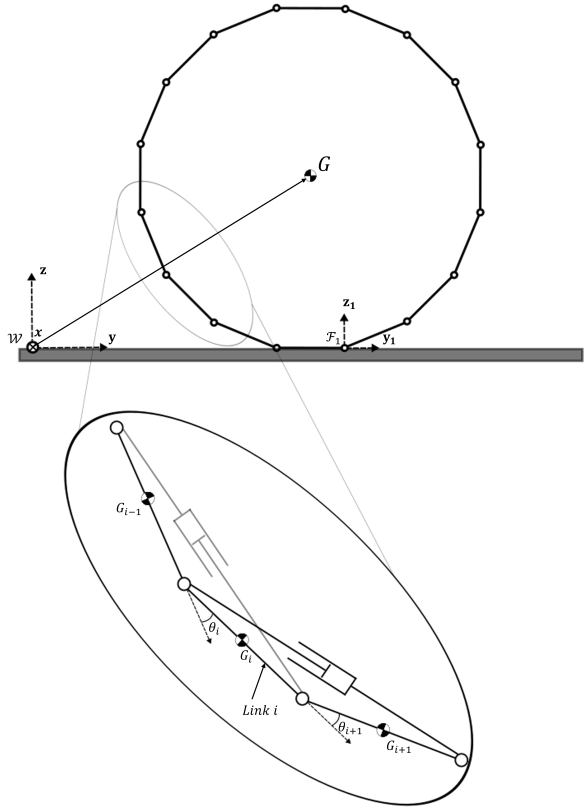

Fig. 3. Kinematic model of the annular robot.

The actuators are placed in an alternating pattern, switching direction at each joint to create a balanced layout around the loop and preventing uneven rolling performance. By changing the actuator lengths, the angles between adjacent links vary, enabling the robot to bend its circular structure, shift its center of mass, and roll forward smoothly.

The rigid links are arced on the outside so that they form a perfect circle when all joints attain equal angles of $\pi/8$ rad. The overall dimensions were constrained by the actuator geometry, ensuring that when each actuator is fully extended, the corresponding joint angle is 0 rad. This configuration helps maintain a compact and continuous rolling profile, simplifying both mechanical integration and control. In this fully circular configuration, the robot reaches a total height of 69.18 cm.

Building on previous work [9] that demonstrated dynamic rolling in a ten-module loop robot, our design maintains a similar principle: controlling shape and stiffness in coordination allows the robot to lift and move its center of mass effectively while preserving structural stability. This previous work showed that maintaining a near-circular shape minimizes resistance and increases rolling speed, a strategy we adopt here to enhance mobility and efficiency.

C. Notation and Kinematic Modelling

To describe the motion of the annular robot, we assume the plane of motion to be oriented such that the horizontal axis is denoted by Y and the vertical axis by Z [7]. A set of notations and reference frames is first defined.

Vectors are expressed with their associated reference frame indicated as a left superscript. For simplicity, when a vector is expressed in the world frame, the superscript is omitted.

Frame $\mathcal{W}$ is the global coordinate system fixed to the ground and used as the main reference frame. Each physical link $\hat{i}$ is associated with a local frame $\mathcal{F}_{\hat{i}}$, placed at the joint connecting it to link $\hat{i}+1$ and aligned with the longitudinal link axis.

For control computations, we introduce a *logical indexing system*: the link currently in contact with the ground is labeled $i = 1$, and the remaining links are numbered sequentially around the loop. The joint angle θ_i denotes the relative orientation between links $i-1$ and i in this logical frame, measured counterclockwise (Fig. 3).

The center of mass (CoM) of link i, expressed in its local frame $\mathcal{F}_i$, is defined as:

$$ {}^{i}\mathbf{G}_i = \begin{bmatrix} -\alpha L \\ 0 \end{bmatrix}, \tag{1} $$

where the left superscript i indicates that the coordinates are expressed in $\mathcal{F}_i$, L is the link length, and α is the normalized position of the CoM along the link ($0 \leq \alpha \leq 1$). In our simulation, we assume $\alpha = 1/2$ for simplicity.

We also assume that the contact link ($i = 1$) is perfectly flat on the ground, which defines the orientation of the moving base frame $\mathcal{F}_1$. As the robot rolls and another physical link $\hat{i}$ makes ground contact, the logical indexing resets so that this new link becomes $i = 1$, while the physical numbering ($\hat{i} = 1, 2, \ldots, N$) remains unchanged. This dynamic re-indexing ensures consistent mathematical treatment throughout motion.

Under this assumption, the position of the CoM of any link $i \in [2, N]$ in frame $\mathcal{F}_1$ is given by:

$$ {}^{1}\mathbf{G}_i = \begin{bmatrix} \sum_{k=2}^{i} \cos(\beta_k) - \alpha \cos(\beta_i) \\ \sum_{k=2}^{i} \sin(\beta_k) - \alpha \sin(\beta_i) \end{bmatrix}, \tag{2} $$

where β_k represents the absolute orientation of link k relative to the horizontal axis, which can be expressed as:

$$ \beta_k = \sum_{j=2}^{k} \theta_j. \tag{3} $$

This allows us to get the CoM of the whole module as follows:

$$ {}^{1}\mathbf{G} = \frac{\sum_{i=1}^{N} m\, {}^{1}\mathbf{G}_i}{\sum_{i=1}^{N} m}, \tag{4} $$

where m is the mass of each link, considered constant and equal for all links. This leads us to the following expression for ${}^{1}\mathbf{G}$:

$$ {}^{1}\mathbf{G} = \frac{L}{N} \begin{bmatrix} -\frac{1}{2} + \sum_{i=2}^{N} \cos(\beta_i)\big(N - i + 1 - \alpha\big) \\ -\frac{1}{2} + \sum_{i=2}^{N} \sin(\beta_i)\big(N - i + 1 - \alpha\big) \end{bmatrix}. \tag{5} $$

From [4], we know that shifting the center of mass (CoM) can initiate a tipping motion which, combined with an approximately circular shape, enables their robot — with a diameter of about $0.28\,\mathrm{m}$ — to achieve significant speeds of up to $1.6\,\mathrm{m/s}$ [10]. The robots studied in these works are continuous, soft, or deformable structures that can smoothly redistribute their mass to generate rolling locomotion. In contrast, our focus is on segmented designs composed of a discrete number of rigid links connected by joints. We believe that the CoM-shifting strategy demonstrated for continuous robots can be applied to such segmented architectures, preserving both the adaptability to various terrains and the potential for high-speed locomotion characteristic of these systems.

III. Model-Based Control Using Center of Mass Jacobian

A. Center of Mass Jacobian

The Jacobian matrix links the speeds of the joints to the movement speed of the robot's center of mass (CoM). In our model, the CoM's position depends only on the joint angles, which we control by changing the lengths of the pistons between each consecutive segments. In this way, the Jacobian shows how changing these joint angles moves the CoM in space.

In this paper, the CoM Jacobian matrix is defined as:

$$ J_G(\mathbf{q}) = \frac{\partial\, {}^{1}\mathbf{G}}{\partial \mathbf{q}}(\mathbf{q}) \tag{6} $$

where the joint angle vector $\mathbf{q}$ is given by:

$$ \mathbf{q} = \big[\theta_1, \theta_2, \ldots, \theta_N\big]^{\top}. \tag{7} $$

By applying (6) to our CoM vector (5), we get the following equation:

$$ J_G = [\mathbf{J}_1 \quad \ldots \quad \mathbf{J}_k \quad \ldots \quad \mathbf{J}_N] \in \mathbb{R}^{2\times N}, \tag{8} $$

with each column J_k given by:

$$ \mathbf{J}_k = \frac{L}{N} \begin{bmatrix} -\sum_{i=k}^{N} \sin(\beta_i)(N - i + 1 - \alpha) \\ \sum_{i=k}^{N} \cos(\beta_i)(N - i + 1 - \alpha) \end{bmatrix}. \tag{9} $$

B. Control Design Based on CoM Jacobian

To achieve controlled rolling with a segmented annular robot, we need to shift its CoM to cause tipping and forward movement. Our control method uses a simple kinematic model and the CoM Jacobian to convert desired CoM motions into joint commands. This model-based approach allows the pneumatic cylinders to work together efficiently [11], improving on basic sequential control methods [2], while keeping the robot's nearly circular shape during rolling.

Specifically, we employ a second-order kinematic approach to regulate the CoM acceleration and convert it into corresponding joint accelerations $\ddot{\mathbf{q}}$. This method facilitates smooth rolling motion while accounting for the coupled dynamics of the segments. By utilizing the CoM Jacobian J_G and its time derivative $\dot{J}_G$, the control law establishes a direct relationship between the desired CoM acceleration and the joint-space acceleration commands, ensuring coordinated actuation of all pneumatic cylinders for efficient rolling.

The desired CoM acceleration $\ddot{\mathbf{x}}$ itself is computed using a proportional-derivative (PD) controller, defined as:

$$\ddot{\mathbf{x}} = K_p\mathbf{e} + K_d\dot{\mathbf{e}}, \tag{10}$$

where $\mathbf{e}$ is the position error and $\dot{\mathbf{e}}$ its time derivative. The gain matrices $K_p, K_d \in \mathbb{R}^{2\times N}$ are positive definite diagonal matrices, with $K_d = 2\sqrt{K_p}$.

Finally, the joint accelerations $\ddot{\mathbf{q}}$ required to produce the desired CoM acceleration are computed as:

$$\ddot{\mathbf{q}} = J_G^{\#}\big(\ddot{\mathbf{x}} - \dot{J}_G\dot{\mathbf{q}}\big), \tag{11}$$

where $J_G^{\#}$ denotes the Moore-Penrose pseudoinverse of the CoM Jacobian.

In this context, we also require the time derivative of the CoM Jacobian, expressed as:

$$\dot{J}_G = [\dot{\mathbf{J}}_1 \ \dots \ \dot{\mathbf{J}}_k \ \dots \ \dot{\mathbf{J}}_N], \tag{12}$$

where each column $\dot{J}_k$ is given by:

$$\dot{\mathbf{J}}_k = -\frac{L}{N}\begin{bmatrix}\sum_{i=k}^{N}\dot{\beta}_i\cos(\beta_i)\big(N-i+1-\alpha\big)\\ \sum_{i=k}^{N}\dot{\beta}_i\sin(\beta_i)\big(N-i+1-\alpha\big)\end{bmatrix}. \tag{13}$$

C. Secondary Task Optimization in Null-Space

Our system exhibits a high degree of kinematic redundancy, meaning that multiple joint configurations can achieve the same desired CoM trajectory. This redundancy can be exploited through a null-space projection [12], which allows secondary objectives to be fulfilled without interfering with the primary CoM control task. In particular, maintaining the robot's circular shape is important for stable and efficient rolling locomotion [4]. To accomplish this, we project an additional corrective action into the null space of the CoM Jacobian. The projection matrix P_{null} is defined as:

$$P_{\text{null}} = I_N - J_G^{\#}J_G, \tag{14}$$

where I_N is the identity matrix of size N and $J_G^{\#}$ denotes the Moore-Penrose pseudoinverse of the CoM Jacobian J_G.

The secondary task is formulated as a joint correction vector $\mathbf{z}$ that attracts the robot's configuration $\mathbf{q}$ towards a reference circular configuration $\mathbf{q}_{\text{ref}}$:

$$\mathbf{z} = K_{\text{null}}(\mathbf{q}_{\text{ref}} - \mathbf{q}), \tag{15}$$

where K_{null} is a positive scalar gain. Since our robot is composed of N joints, to form a circle, each joint angle should be $\frac{\pi}{8}$ rad. Therefore, $\mathbf{q}_{\text{ref}} \in \mathbb{R}^N$ is the vector

$$\mathbf{q}_{\text{ref}} = \left[\frac{\pi}{8}, \frac{\pi}{8}, \dots, \frac{\pi}{8}\right]^T.$$

This null-space contribution is combined with the primary inverse-Jacobian control law to obtain the following final joint acceleration command:

$$\ddot{\mathbf{q}} = J_G^{\#}\big(\ddot{\mathbf{x}} - \dot{J}_G\dot{\mathbf{q}}\big) + P_{\text{null}}\mathbf{z}. \tag{16}$$

D. Simplified Torque-to-Pressure Conversion

In this work, the control input for each joint ultimately takes the form of a pressure command sent through proportional actuator valves to the air cylinders. Ideally, the desired joint angular accelerations would be mapped into actuator torques by accounting for the robot's full dynamic behavior. This dynamic model includes the effects of inertia, Coriolis and centrifugal forces, as well as gravity, all of which influence the joint torques required for precise control.

However, accurately identifying and implementing this dynamic model is particularly challenging due to the high number of joints in the segmented annular robot. Each joint's dynamics are coupled with those of the others, making the full dynamic characterization and control complex.

In our prototype, the links are lightweight and the robot primarily moves in the horizontal plane. As a result, the gravitational torques at the joints are relatively small compared to the actuator-generated torques. Neglecting the gravity component simplifies the control design without significantly impacting performance. This assumption is valid for the planned motions of our prototype, but may need to be reconsidered for future applications involving vertical motion or heavier links.

Therefore, as a practical simplification, we approximate the required actuator torque $\boldsymbol{\tau}$ using:

$$\boldsymbol{\tau} = I\ddot{\mathbf{q}}, \tag{17}$$

where I is the moment of inertia of the link relative to the joint axis. Since all links are identical, I is treated as a constant scalar. This formulation assumes that inertial effects dominate and neglects Coriolis, centrifugal, and gravity components within the control timescale of interest.

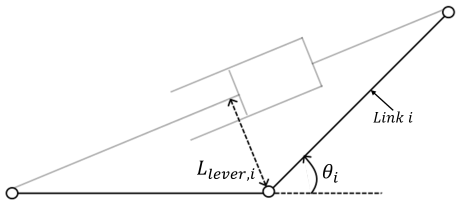

Fig. 4. Schematic of the lever arm geometry.

The torque vector is converted into the corresponding actuator force vector using element-wise division by the lever-arm lengths:

$$\mathbf{F} = \boldsymbol{\tau} \oslash \mathbf{L}_{\text{lever}}, \qquad \boldsymbol{\tau},\, \mathbf{L}_{\text{lever}} \in \mathbb{R}^N, \tag{18}$$

where $\oslash$ denotes element-wise division, i.e., $F_i = \tau_i / L_{\text{lever,i}}$ for $i = 1, \ldots, N$.

As illustrated in Fig. 4, L_{lever} is the perpendicular distance from the joint axis to the actuator's line of action. It can be obtained from the right triangle defined by the link of length L (hypotenuse) and the angle $\frac{\pi-\theta_i}{2}$, leading to:

$$\cos\left(\frac{\pi-\theta_i}{2}\right) = \frac{L_{\text{lever},i}}{L}, \tag{19}$$

where $L_{\text{lever},i}$ is the lever arm corresponding to link i. Using the identity $\cos\left(\frac{\pi-\theta_i}{2}\right) = \sin\left(\frac{\theta_i}{2}\right)$, we obtain:

$$L_{\text{lever},i} = L\left|\sin\left(\frac{\theta_i}{2}\right)\right|, \tag{20}$$

which represents the effective moment arm of the actuator force.

Finally, the corresponding pressure command $\mathbf{P}$ to be sent to the proportional valves is computed by dividing the actuator force vector by the piston cross-sectional area S which is a constant scalar:

$$\mathbf{P} = \frac{\mathbf{F}}{S}. \tag{21}$$

This simplified approach reduces model complexity and enables straightforward real-time implementation while maintaining acceptable control performance. Simulation results indicate that this approximation is sufficient to achieve the desired rolling motion under the considered operating conditions. The computed pressures are converted into a proportion of a maximum pressure $P_{\max}$ and then saturated, constraining the values between -100% and 100%. The resulting signal, denoted as $\mathbf{u}$, is obtained by combining (17), (18), and (21):

$$\mathbf{u}_i = \frac{I}{L_{\text{lever},i} S}\ddot{\mathbf{q}}_i, \tag{22}$$

where $\ddot{\mathbf{q}}_i$ is the angular acceleration of link i from (16).

IV. Simulation

A. Simulink® Model Architecture

To validate the proposed control strategy and assess the robot's rolling performance, a simulation environment was developed using MATLAB® Simulink® and Simscape®, as shown in Fig. 5. This setup replicates the kinematic and simplified dynamic behavior of the segmented annular robot, allowing rapid testing and tuning of parameters before proceeding with the hardware implementation.

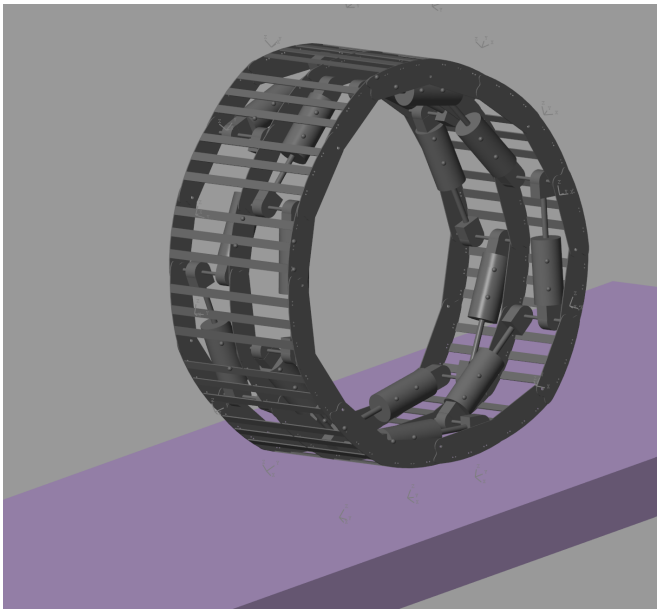

Fig. 5. Visualization of the annular robot in the MATLAB® Simulink® simulation environment. The robot has a height of $h \approx 0.7\,\text{m}$ and a width of $w = 0.3\,\text{m}$.

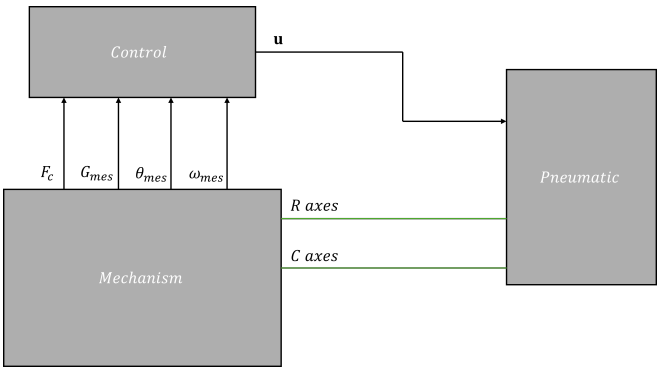

Fig. 6. Main subsystems of the Simulink® model. C denotes the cylinders, and R denotes the rods.

The Simulink® model is divided into three main subsystems representing the physical components and control loops of the robot. A block diagram of the overall structure is shown in Fig. 6, highlighting the main signal flow and interactions between these subsystems. Figs. 7, 8, and 9 are simplified diagrams of the corresponding Simulink® blocks.

The *mechanism subsystem* models the physical interactions among the rigid components (Fig. 7). It consists of sixteen interconnected units joined by revolute joints. The entire assembly is "connected" to the ground via a 6-DOF joint. Every second unit features rods and cylinders attached to right and left axes that enable rotational movement of these units. These rods and cylinders serve as inputs to the

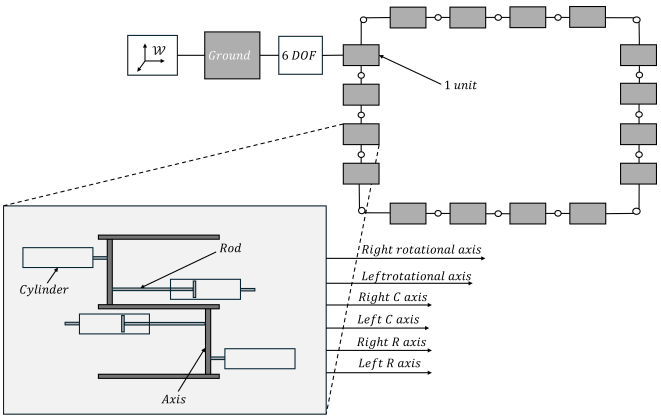

Fig. 7. Details of the mechanism subsystem in the Simulink® model. The zoomed-in section shows a top view of a single unit. C denotes the cylinders, and R denotes the rods. Black arrows represent Simulink® signals (vectors), green lines denote axes used for physical connections, and purple dashed lines indicate air pressure signals.

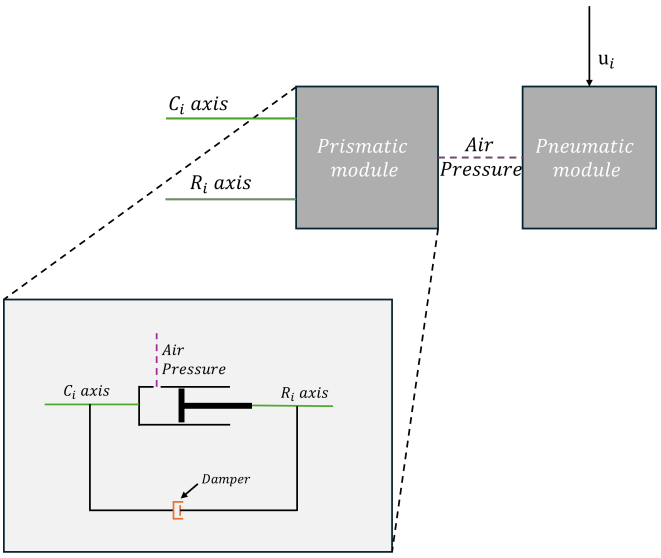

Fig. 8. Details of the pneumatic subsystem in the Simulink® model. C_i denotes the i-th cylinder, and R_i denotes the i-th rod. The color code is the same as Fig. 7.

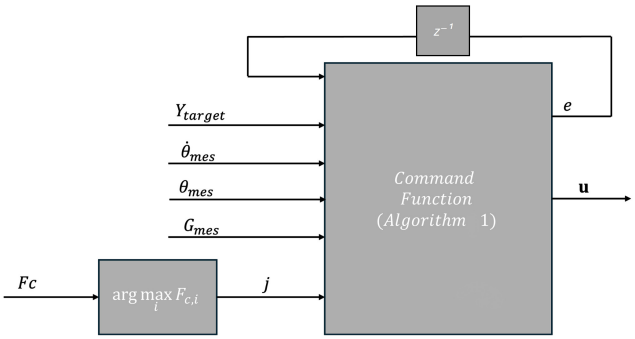

Fig. 9. Details of the control subsystem in the Simulink® model. The color code is the same as Fig. 7.

pneumatic subsystem, where they constitute the pneumatic actuators. The remaining units lack these rods and cylinders but still provide two rotational axes for connecting adjacent units through revolute joints. This subsystem also includes sensors such as the Inertia sensor from the Simscape® library, which enables obtaining the position of the center of mass (CoM) for groups of connected elements—either directly or through rigid transforms—allowing us to determine the CoM of the entire robot, $\mathbf{G}_{\text{mes}}$. This information is crucial for our control loop. Additionally, the normal force between the ground and each module is measured, forming a vector $\mathbf{F}_c \in \mathbb{R}^N$ that is sent to the *control subsystem*. Relative angles between units, $\boldsymbol{\theta}_{\text{mes}}$ and their velocities, $\boldsymbol{\omega}_{\text{mes}}$, are also directly measured. All these signals are fed into the *control subsystem*.

The *pneumatic subsystem* consists of sixteen blocks modeling each of the pneumatic actuators. As shown in Fig. 8, each block represents a springless, single-acting pneumatic cylinder, as mentioned in Sec. II. Within each block, as illustrated in Fig. 8, the prismatic module connects the rod and cylinder axes using a prismatic joint (from the Simscape® library), which allows translational motion along the axis. This joint is coupled with a translational mechanical converter and a damper to model the dynamic behavior and resistance within the actuator. The single-acting pneumatic cylinder admits air into one chamber only, and the pneumatic module controls this airflow through proportional valves. When the control signal is positive, the valve opens the supply path to pressurize the chamber, extending the actuator; when negative, the valve opens the exhaust path to release air, allowing retraction.

The *control subsystem* consists of two MATLAB® functions (Fig. 9).

The first function computes the index j of the unit experiencing the largest normal contact force, expressed as

$$j = \arg \max_{i \in [\![1,N]\!]} F_{c,i},$$

where $F_{c,i}$ denotes the normal contact force measured at the i^{th} unit. The operator is defined by

$$\arg \max_{i \in [\![1,N]\!]} F_{c,i} = \{\, j \in [\![1,N]\!] \mid F_{c,j} \geq F_{c,i} \ \forall i \in [\![1,N]\!] \}\,.$$

The second function, called the *Command Function*, takes as inputs the measurements $\boldsymbol{\theta}_{\text{mes}}$, $\boldsymbol{\omega}_{\text{mes}}$, and $\mathbf{G}_{\text{mes}}$ provided by the *mechanism subsystem*, as well as Y_{target}, which represents the horizontal coordinate of the target position. It also computes the position error e, which is returned to facilitate PD control. To prevent algebraic loops that arise during the calculation of the derivative of the error $\dot{e}$, a delay block is incorporated in the feedback loop. Finally, the function outputs the control command $\mathbf{u}$.

Algorithm 1 implements the core of the control strategy. It begins by shifting the measured joint angles and angular velocities based on the index `j` of the unit currently in contact with the ground. This index shifting simplifies the computation of the Jacobian matrix, as the kinematic model considers the link touching the ground to be indexed as `1`. Using these shifted indexes, the Jacobian matrix `J` and its

Algorithm 1: Command Function

Input: e, Y_target, theta_mes, omega_mes, G_mes, j
Output: e, u

for $k \leftarrow 1$ **to** N **do**
  shifted_theta(k) ← theta_mes[((k + j − 2) mod N) + 1];
  shifted_omega(k) ← omega_mes[((k + j − 2) mod N) + 1];
  ; // Convert physical indexing $\hat{i}$ to logical indexing i
J ← jacobian(shifted_theta, shifted_omega); // Eq. (9)
J_dot ← jacobian_dot(shifted_theta, shifted_omega); // Eq. (13)
x_ddot ← Kp · e + Kd · e_dot ; // Eq. (10)
shifted_u ← command(x_ddot, J, J_dot, shifted_theta, shifted_omega); // Eq. (22)
for $k \leftarrow 1$ **to** N **do**
  u(k) ← shifted_u[((k − j) mod N) + 1];
  ; // Convert logical indexing i back to physical indexing $\hat{i}$
return e, u;

time derivative J_dot are calculated to relate joint velocities and accelerations to the robot's body velocity and acceleration. Next, a desired acceleration x_ddot is computed from a PD controller using the position error e and its derivative e_dot. This desired acceleration is then mapped back to joint space through an inverse kinematics command function (22), yielding the shifted control inputs shifted_u. Finally, these control inputs are rearranged to the original indexing and returned along with the updated error. This process enables the controller to consistently generate commands relative to the current ground contact, ensuring smooth and accurate rolling behavior.

B. Results

Fig. 10 compares the robot's displacement trajectories for a target displacement of 2.5 m under two control strategies: (i) with nullspace control, and (ii) without nullspace control. The former, designed to preserve the robot's circular shape, yields a consistently higher forward velocity in the transient phase. Within the first 5 s, the robot using the *Nullspace* strategy surpasses 2.5 m, whereas with the *No nullspace* strategy, it only reaches about 2.3 m over the same period. After the transient response, both configurations converge to the 2.5 m target, but their regulation behaviors differ: the *No nullspace* case overshoots to nearly 2.9 m and oscillates with slower decay, while the *Nullspace* case exhibits reduced overshoot, faster damping, and smoother convergence. These results highlight that preserving the circular shape via

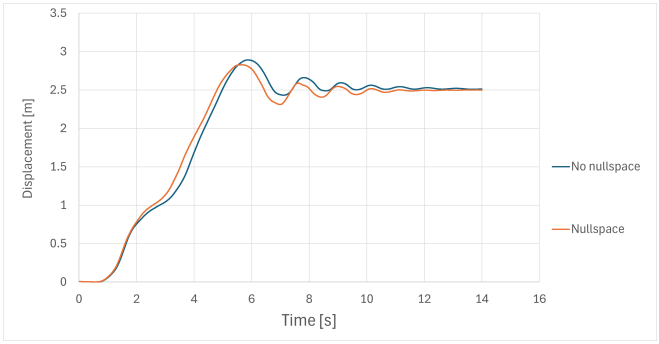

Fig. 10. Time evolution of the robot's displacement for two control configurations. The *No nullspace* strategy corresponds to $K_{\text{null}} = 0$, whereas the *Nullspace* strategy uses $K_{\text{null}} \neq 0$.

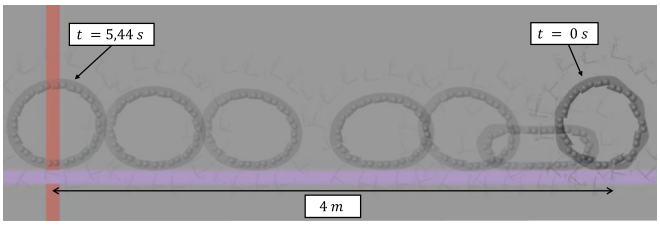

Fig. 11. Superimposed frames from the simulation showing the robot's motion over time. Each frame corresponds to a 1-second interval, except for the final frame.

nullspace control not only enhances motion stability but also improves propulsion efficiency.

The robot's motion during the experiment can be divided into three main phases:
1) **Inflation** – The structure expands to reach a nearly circular shape, improving rolling dynamics.
2) **Acceleration** – The CoM is intentionally shifted toward the target direction, generating a tipping motion that initiates rolling. This phase produces an elliptical deformation due to the asymmetrical load distribution.
3) **Deceleration** – The CoM shift is reduced, stabilizing the robot and slowing down the motion.

The circular shape achieved during the inflation phase is critical, as it minimizes rolling resistance and enables faster displacement. As illustrated in Fig. 11, and considering the robot's diameter of approximately 0.7 m (corresponding to a height of 69.18 cm), the robot is capable of covering nearly 4 m in under 6 seconds, corresponding to an average speed of approximately 0.67 m/s. This performance confirms the effectiveness of the proposed control strategy within the moderate-speed operating regime.

C. Discussion

The results confirm that CoM-Jacobian-based control with null-space optimization markedly improves rolling locomotion. The null-space controller achieved less overshoot (2.6 m vs. 2.9 m) and faster convergence, confirming that preserving a near-circular shape enhances rolling efficiency [4].

1) Control Strategy Effectiveness: The coordinated actuation strategy enables the robot to cover about 4 m in under

6 s, demonstrating strong potential for planetary surface mobility. The three-phase motion—inflation, acceleration, and deceleration—produces smoother profiles than traditional methods [2].

Early experiments showed a directional bias due to uneven actuator orientation. Implementing an alternating pattern around the loop greatly reduced this asymmetry, yielding balanced motion despite a slight residual speed difference. The springless pneumatic actuators replicate the inflation/deflation behavior of inflatable habitat modules and naturally support the asymmetric actuation profile required for rolling, avoiding complex spring mechanisms that pose risks in space [13].

2) Model Simplification and Limitations: The simplified torque-to-pressure conversion (Equation (22)) assumes inertial forces dominate and neglects Coriolis, centrifugal, and gravity effects. This is valid for moderate-speed rolling of the lightweight, segmented prototype. Simulations—covering 4 m in under 6 s at an average speed of ≈ 0.67 m/s—show adequate performance. This trade-off aligns with trends in soft robotics favoring simplified actuation models to reduce hardware complexity [14]. Remaining limitations include the assumption of uniform ground contact and the fact of neglecting of pneumatic-structural coupling. Future work on discrete dynamic formulations [15] and comprehensive control strategies [16] may overcome these issues.

V. Conclusions

This work presented a CoM-Jacobian-based control with null-space optimization for annular pneumatic robots, achieving efficient rolling while maintaining circular shape. Key contributions include: (1) a simplified rigid-segment design inspired by inflatable habitat modules, (2) an inverse-Jacobian control law for coordinated actuation, and (3) null-space optimization for shape preservation.

The robot achieved $\approx$4 m displacement in under 6 s, confirming superior performance to baseline methods. A 16-segment prototype is under fabrication at CNRS-AIST JRL with CNC-machined links and a distributed sensor suite (potentiometers, IMUs, and force sensors) to provide full state feedback for real-time control via Extended Kalman Filtering.

These results establish a practical foundation for advanced control of soft robotic structures and contribute toward autonomous deployment of inflatable habitat modules in future space missions.

Acknowledgments

This work is supported by JST [Moonshot R&D] [Grant Number JPMJMS223B].

References

[1] Y. Sakuraba, M. Yamashita, S. Okayama, J. Koyanagi, R. Higuchi, T. Aoki, and S. Kimura, "Autonomous shape and motion control system for multi-chambered inflatable structures," in *i-SAIRAS 2024 - 17th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2024.

[2] 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," *Robotics and Autonomous Systems*, vol. 140, p. 103715, 2021.

[3] Y. Matsumoto, H. Nakanishi, and S. Hirai, "Rolling locomotion of a deformable soft robot with built-in power source," in *Advances in Mobile Robotics*, August 2008, pp. 365–372.

[4] M. Puopolo, J. Jacob, and E. Gabino, "Locomotion of a cylindrical rolling robot with a shape changing outer surface," *Robotics*, vol. 7, no. 3, p. 52, 2018.

[5] N. A. Mansour, T. Jang, H. Baek, B. Shin, B. Ryu, and Y. Kim, "Compliant closed-chain rolling robot using modular unidirectional sma actuators," *Sensors and Actuators A: Physical*, vol. 310, p. 112024, 2020.

[6] M. G. Puopolo and J. D. Jacob, "Velocity control of a cylindrical rolling robot by shape changing," *Advanced Robotics*, vol. 30, no. 23, pp. 1484–1494, 2016.

[7] J. M. Porta, L. Ros, O. Bohigas, M. Manubens, C. Rosales, and L. Jaillet, "An open-source toolbox for motion analysis of closed-chain mechanisms," in *Computational Kinematics*, ser. Mechanisms and Machine Science, F. Thomas and A. P. Gracia, Eds. Dordrecht: Springer, 2014, vol. 15.

[8] J. M. Gandarias, Y. Wang, A. Stilli, A. J. García-Cerezo, J. M. G. de Gabriel, and H. A. Wurdemann, "Open-loop position control in collaborative, modular variable-stiffness-link (vsl) robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1772–1779, April 2020.

[9] J. Sastra, S. Chitta, and M. Yim, "Dynamic rolling for a modular loop robot," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and D. Rus, Eds. Berlin, Heidelberg: Springer, 2008, vol. 39.

[10] D. Mellinger, V. Kumar, and M. Yim, "Control of locomotion with shape-changing wheels," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1750–1755.

[11] T. Dewolf, "Robot control part 4: Operation space control," https://studywolf.wordpress.com/2013/09/17/robot-control-4-operation-space-control/, 2015, accessed: 13 May 2025.

[12] F. Vigoriti, F. Ruggiero, V. Lippiello, and L. Villani, "Control of redundant robot arms with null-space compliance and singularity-free orientation representation," *Robotics and Autonomous Systems*, vol. 100, pp. 186–193, 2018.

[13] M. S. Xavier, A. J. Fleming, and Y. K. Yong, "Finite element modeling of soft fluidic actuators: overview and recent developments," *Advanced Intelligent Systems*, vol. 3, no. 2, p. 2000187, 2021.

[14] N. K. Uppalapati and G. Krishnan, "Towards pneumatic spiral grippers: modeling and design considerations," *Soft Robotics*, vol. 5, no. 6, pp. 695–709, 2018.

[15] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserat approach for multisection soft manipulator dynamics," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1518–1533, 2018.

[16] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robotics*, vol. 5, no. 2, pp. 149–163, 2018.