École Nationale Supérieure de Techniques Avancées,
Spécialité Robotique

CONTROL AND SIMULATION OF AN ARTICULATED ANNULAR PNEUMATIC ROBOT

Ghita Tazeroualt Tlamcani

Internship supervisor:

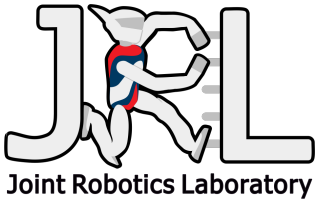
Hiroshi Kaminaga

cnrs-aist jrl (Joint Robotics Laboratory)

Academic supervisor:

Prof. Luc Jaulin

University Bretagne Occidentale, Lab-STICC, UMR 6285, ENSTA



Master 1 internship report - academic year 2024/2025

Acknowledgments

First and foremost, I would like to express my sincere gratitude to AIST and CNRS for offering me the opportunity to carry out this enriching research internship. This experience has been invaluable for both my academic development and personal growth.

I am deeply thankful to my supervisors at JRL, Hiroshi Kaminaga and Rafael Cisneros-Limón, for their constant guidance, insightful advice, and support throughout the internship. I would also like to extend my appreciation to Naoko Sakai for the invaluable administrative support provided during my stay in Japan and my time in the laboratory.

My heartfelt thanks go as well to all the members of the CNRS-AIST JRL for their warm welcome, collaborative spirit, and the stimulating environment they created. It has been a true pleasure to work and exchange ideas with everyone during this internship.

Abstract

This internship report presents the development of a model-based control strategy for a rigid analog of the HIDAS inflatable robot at the CNRS-AIST Joint Robotics Laboratory in Tsukuba, Japan, as part of the Moonshot Program on space robotics.

A simplified sixteen-link rigid robot with rotational joints was designed, actuated by single-acting pneumatic cylinders, and refined for compactness and symmetry. A kinematic control strategy based on center of mass (CoM) manipulation using Jacobian methods was implemented, with null-space optimization maintaining the robots circular shape during rolling motion. Simulations validated rolling locomotion of nearly 4 meters in under 6 seconds, with improved stability.

The work also includes a dynamic model incorporating holonomic constraints, Coriolis forces, and gravity, as well as preliminary Unscented Kalman Filter implementation for state estimation. Key contributions are: (1) mechanical redesign, (2) CoM-based control with null-space optimization, (3) simulation framework, and (4) initial state estimation.

These results demonstrate the feasibility of applying CoM-shifting strategies from soft to rigid-link robots, providing a foundation for future hardware implementation in space robotics, particularly inflatable structures for lunar habitats.

Keywords: Annular robot, pneumatic actuation, center of mass control, Jacobian methods, space robotics, Moonshot program

Résumé

Ce rapport de stage présente le développement d'une stratégie de contrôle basée sur un modèle pour un robot rigide analogue à lHIDAS, un robot pneumatique gonflable, au CNRS-AIST Joint Robotics Laboratory à Tsukuba, Japon, dans le cadre du programme Moonshot sur la robotique spatiale.

Un robot rigide simplifié à seize segments articulés a été conçu, actionné par des vérins pneumatiques à simple effet, et optimisé pour la compacité et la symétrie. Une stratégie de contrôle cinématique basée sur le déplacement du centre de masse (CoM) utilisant les méthodes jacobiennes a été mise en œuvre, avec une optimisation dans l'espace nul pour maintenir la forme circulaire du robot lors de la locomotion roulante. Les simulations ont validé un déplacement d'environ 4 mètres en moins de 6 secondes, avec une meilleure stabilité.

Le travail inclut également un modèle dynamique intégrant les contraintes holonomes, les forces de Coriolis et la gravité, ainsi qu'une implémentation préliminaire d'un filtre de Kalman Unscented pour l'estimation d'état. Les contributions principales sont : (1) la refonte mécanique, (2) le contrôle cinématique basé sur le CoM avec optimisation dans l'espace nul, (3) le cadre de simulation, et (4) le travail initial sur l'estimation d'état.

Ces résultats démontrent la faisabilité de l'application des stratégies de déplacement du CoM des robots souples aux architectures rigides, offrant une base pour une future mise en œuvre matérielle dans la robotique spatiale, notamment pour les structures gonflables destinées aux habitats lunaires.

Mots-clés : Robot annulaire, actionnement pneumatique, contrôle du centre de masse, méthodes jacobiennes, robotique spatiale, programme Moonshot

Contents

Introduction

I considered my second-year engineering internship to be an excellent opportunity to challenge myself. Japan was an ideal choice for a research internship in robotics, as it is a global leader in this field. The culture, markedly different from what I am accustomed to in Western and North African contexts, added a unique dimension to the experience. Moreover, the focus of my internship on space robotics made the opportunity even more fascinating and engaging. In fact, during my first interview, I was introduced to the project as working on the control of an inflatable robot designed to serve as habitats on the Moon.

This project is part of Japans Moonshot Research and Development Program, a national initiative aimed at tackling ambitious technological challenges to achieve long-term societal and scientific goals. Specifically, the program includes projects that explore advanced robotics and space technologies to support human activities beyond Earth. Within this context, my internship focused on developing control strategies for inflatable robotic structures that could serve as lunar habitats.

The main objective of my internship was to develop a rigid analog of the inflatable robot HIDAS, enabling more controlled experiments and simulations. My work primarily focused on the modeling, control, and simulation of this rigid prototype, which involved designing appropriate control algorithms and testing their performance under various conditions. This approach allowed me to gain hands-on experience with both theoretical and practical aspects of robotics research.

1 Laboratory and Project Context

1.1 The Laboratory

The Joint Robotics Laboratory (JRL) is a research laboratory established through a collaboration between Frances Centre National de la Recherche Scientifique (CNRS) and Japans National Institute of Advanced Industrial Science and Technology (AIST). It is located in Tsukuba, Japan, a city renowned for its high concentration of scientific research centers.

JRL specializes in humanoid robotics and related technologies, including perception, locomotion, motion planning, object manipulation, and humanrobot interaction. Its work combines fundamental research with technological development, contributing significantly to complex robotics projects as well as to open-source software platforms such as mc_rtc[1]. The team brings together researchers and engineers of diverse nationalities and maintains active collaborations with numerous academic and industrial partners in both France and Japan.

The laboratory is particularly well known for its contributions to the development

[1] *mc_rtc* is an open-source robot control framework. It provides a modular architecture to describe, plan, and execute complex tasks in humanoid robotics, particularly for manipulation and locomotion. `https://jrl.cnrs.fr/mc_rtc/index.html`

of humanoid robots, such as the HRP series, widely used in academic research and industrial demonstrations. The research topics covered at JRL span a broad spectrum, including sensor perception and fusion, dynamic locomotion, complex motion planning, bimanual manipulation, and natural humanrobot interaction. This diversity creates an inspiring and stimulating environment for students, offering a unique opportunity to observe how highly specialized expertise converges to create fully integrated robotic systems.

1.2 The Moonshot project

This internship project is part of Japans Moonshot R&D Program, a large-scale initiative launched by the Cabinet Office of Japan in 2020 to address major societal and technological challenges through disruptive innovations by 2050. Among its ambitious goals, the program dedicates significant resources to the **exploration and utilization of outer space**, focusing on the development of autonomous robots for space missions, the creation of sustainable lunar and Martian infrastructures, and the enhancement of humanrobot collaboration in extreme environments.

The Moonshot Project promotes a multi-disciplinary approach that combines advanced robotics, AI, and aerospace engineering to make human and robotic activities in space more sustainable and efficient. During my internship, I also had the opportunity to visit the Moon surface emulation facility at JAXA in Sagamihara, where various robotic systems are tested under conditions simulating the lunar environment[2]. During this visit, I observed a version of the inflatable robot HIDAS, which is at the core of my internship project and the system I was tasked with replicating in a rigid analog for control and simulation purposes.

1.3 My mission

The main goal of this internship is to work on a model-based control strategy for the HIDAS robot, a system known for its complex dynamics that make it challenging to control effectively. To address this difficulty, the project focuses on developing a rigid analog of the HIDAS robot, enabling easier modeling, simulation, and control validation.

More details about the HIDAS robot, including its design, specifications, and operational characteristics, will be presented in the following section.

Initially, my objective was to build upon the work of a previous intern who had developed the CAD model, the simulation environment, and the initial control strategies for the inflatable robot. My role was to adapt this framework to integrate real-life sensors, implement an observer for state estimation, and eventually build the first prototype.

However, as I became more familiar with the project, we identified several areas that required significant improvements. The control algorithms needed to be refined for better mathematical consistency, the simulation lacked smoothness and accuracy, and the robots geometry required adjustments to achieve a more compact and symmetrical design. Consequently, I decided to restart the development process, redesigning both

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

the control framework and the CAD model from the ground up to ensure a more robust and scalable foundation for the project.

2 Design and Modeling of a Pneumatic Annular Robot

2.1 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. 1). 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].

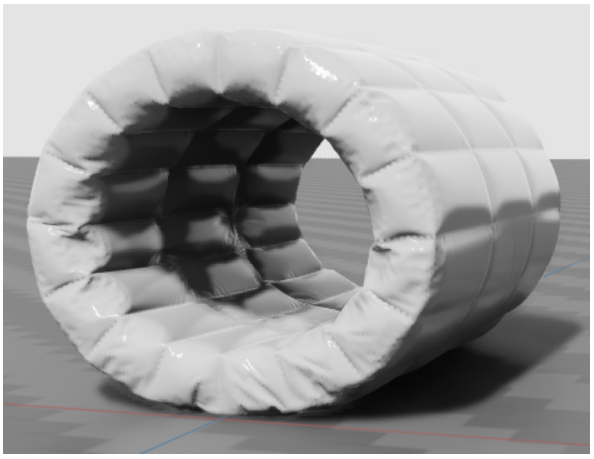


Figure 1: The original HIDAS robot.

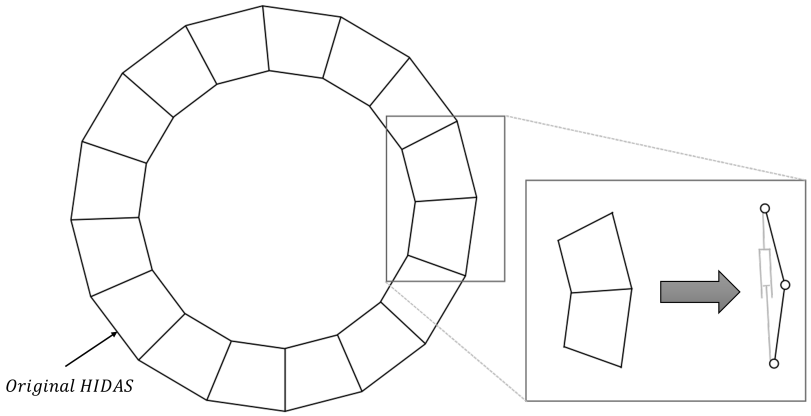


Figure 2: Design simplification from HIDAS inflatable module to a kinematic chain composed by rigid segments driven by pneumatic actuators.

To address this challenge, we developed a simplified rigid-segment robot that preserves HIDASs modular actuation concept in the motion plane while allowing straight-

forward 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. 2): applying pressure causes extension, while releasing pressure through the exhaust results in passive retraction without restoring spring-like forces.

2.2 Mechanical Architecture

The simplified robot consists of sixteen rigid links connected in a closed loop by rotational joints (Fig. 3) [2]. Each link and joint can adjust its stiffness using pneumatic actuators [3], allowing the robot to dynamically adapt its shape for better stability and improved rolling performance. This concept builds on previous work [4] involving a ten-module loop robot capable of dynamic rolling. In that study, the robot leveraged sensor feedback and precise control to modify its shape at the optimal moments, shifting its center of mass to roll forward efficiently while maintaining structural stability. Their findings demonstrated that maintaining a shape as close as possible to a perfect circle reduces motion resistance and enables smoother and faster rolling. These insights strongly influenced the design choices adopted for our segmented robot.

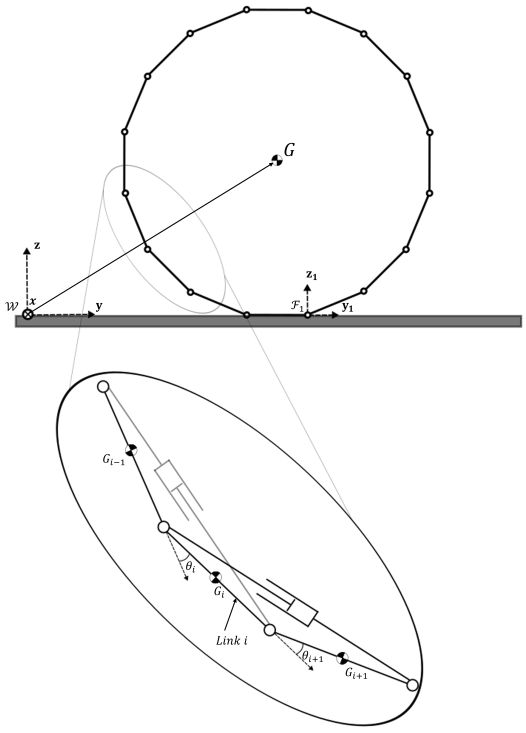


Figure 3: Kinematic model of the annular robot.

The initial CAD design (Fig. 4), however, revealed several limitations. While asymmetry in the internal structure is inevitable due to the use of single-actuated cylinders, the external shape also exhibited avoidable asymmetries. Inspired by a more circular

design proposed in [5], we refined the model to achieve a smoother, more symmetric outer surface. Additionally, the original dimensions—approximately 0.9 m in height and 0.2 m in width—were not optimal. The disproportionate height-to-width ratio made the robot less compact and more prone to balance issues, motivating a redesign for better stability and space efficiency.

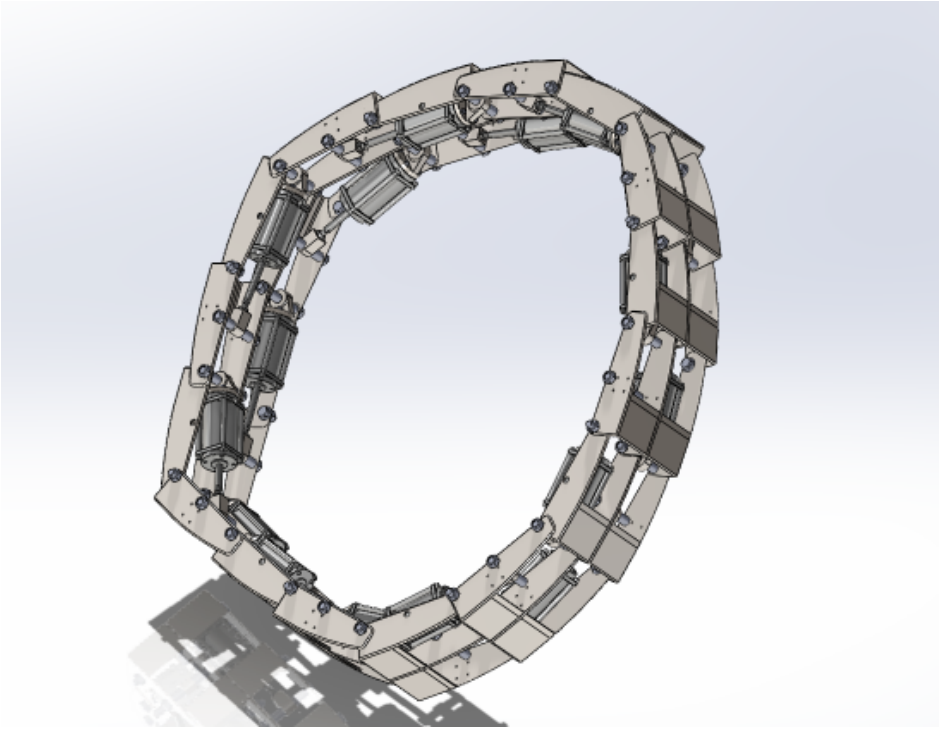


Figure 4: Initial CAD.

2.2.1 Initial Unit Design

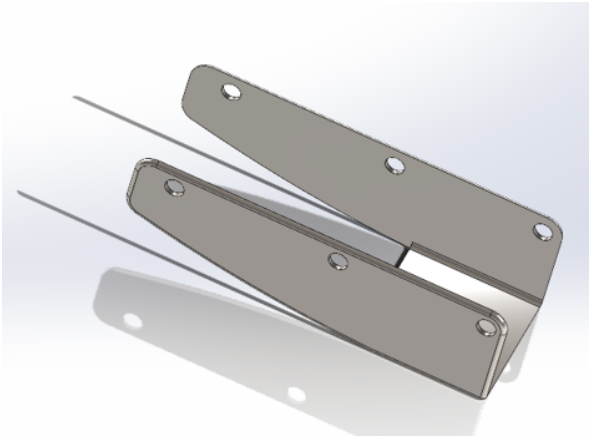


Figure 5: Initial design of a single unit.

The first unit prototype (Fig. 5) was designed with three axes: two lateral axes for connecting the cylindrical bodies of the actuators, and one central axis for attaching

the actuator rod. Depending on the overall organization of the robot, these axes may or may not host actuator components, providing flexibility in how units and cylinders are combined.

The arrangement of actuators and units is illustrated in Fig. 6. The units are organized in two rows, enabling control of all sixteen angles in the loop. The figure highlights the periodic pattern that emerges: pairs of units placed in two rows with their corresponding actuators attached. This layout served as the basis for later refinements aimed at improving compactness, balance, and symmetry.

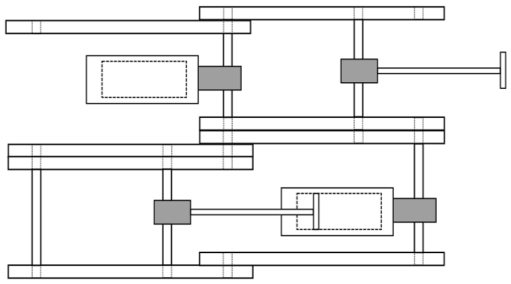


Figure 6: Schematic of the actuator-unit arrangement in the initial prototype, showing the periodic two-row organization.

Another important characteristic of this prototype is its manufacturability as a single piece. Thanks to the flat portion on one side of the unit, combined with the 2 mm thickness and material choice, the design could be produced by cutting an aluminum sheet and bending it into the final shape.

2.2.2 Refined Unit Design

The objective of the new design was to achieve a more compact, symmetric, and efficient structure. To this end, we reduced the configuration from three axes to only two lateral axes. As a result, the length of each unit is directly constrained by the actuator geometry. Specifically, the angle between two consecutive links must remain within the range $[0, \frac{\pi}{2}]$ rad. We therefore imposed that the actuator reaches its maximum extension (with a margin of 5 mm) when the angle between the corresponding links is 0 rad. This condition determines the distance between the two lateral axes, denoted as L throughout the paper.

Inspired by the closed five-arc-shaped-bar linkage robot [5], we also introduced an arced outer surface. This feature ensures that the sixteen links form a perfect circle when all joint angles are equal to $\frac{\pi}{8}$ rad. To achieve this, we first calculated the radius of the circle formed by the axes of all links, given by:

$$r = \frac{L}{2\sin\left(\frac{\alpha}{2}\right)}, \quad \alpha = \frac{\pi}{8}. \tag{1}$$

From this circle, we defined a second, concentric circle corresponding to the outer surface of the unit (Fig. 7). This outer arc raises the structure sufficiently above the ground to prevent the actuators from making contact during operation.

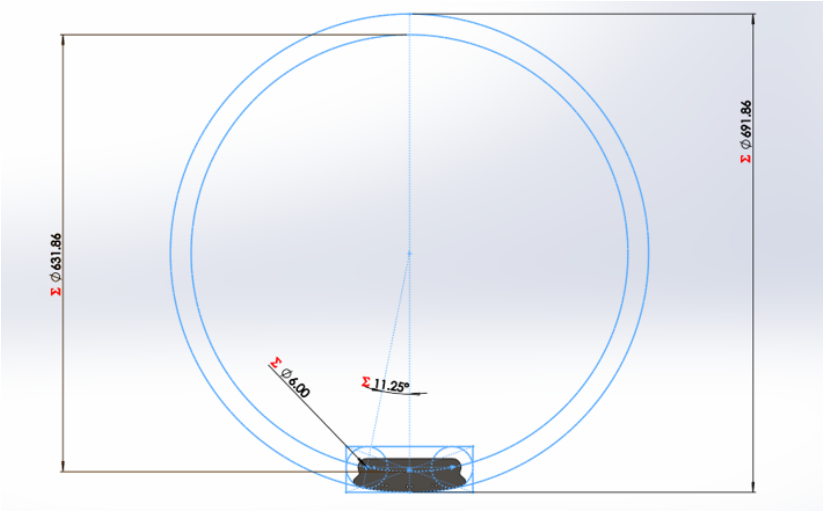


Figure 7: Front view of the refined CAD design, showing the calculated circular geometry.

After obtaining the overall shape of the unit, we focused on removing the excess material. First, we identified the portions of the outer surface that would not come into contact with the ground. To do this, we measured the angle between two consecutive units when the actuator is at its maximum retraction, which corresponds to 115.5° in Fig. 8. This measurement allowed us to determine the points of contact that limit the surface potentially touching the ground by finding the intersection between the outer arced shape of the unit and the bisector of this angle. A margin of 15° was included for safety. Finally, we defined a 10 mm region around the lateral axes and removed all material extending beyond this boundary.

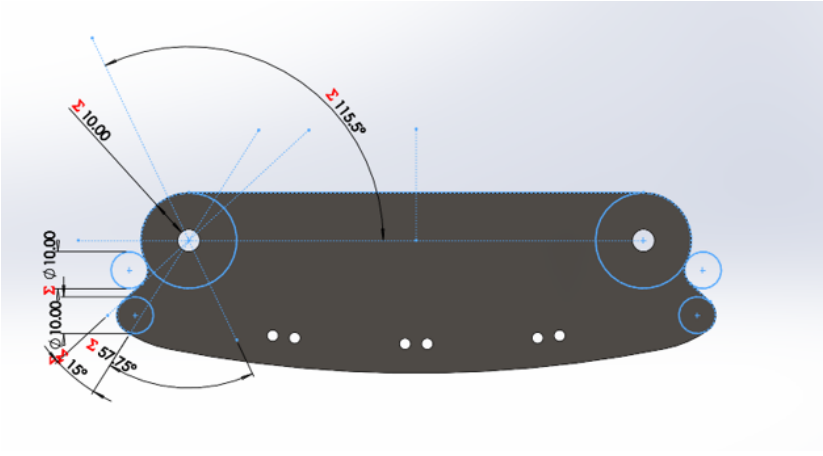


Figure 8: Front view of the refined CAD design, indicating the geometry and boundaries used to remove excess material.

After these modifications, it was no longer possible to manufacture a unit from a single bendable sheet. We therefore designed separate links to attach two sheets together to form a complete unit (Fig. 9). These links are 15 cm long, which increases the overall width of the robot and improves its stability. With this length, two rows of units now form a total width of approximately 35 cm, compared to only 18 cm in the

previous design. This wider layout also accommodates the transition from three axes to two, as each unit now has enough space to fit two actuators side by side.

The link width is minimized so that it does not interfere with the arced outer surface, even though the link itself is flat. The links are attached to the unit surfaces using two screws and are designed for easy manufacturing. With a thickness of 2 mm, they are bendable along the sides, allowing secure attachment to the rest of the assembly (Fig. 10).

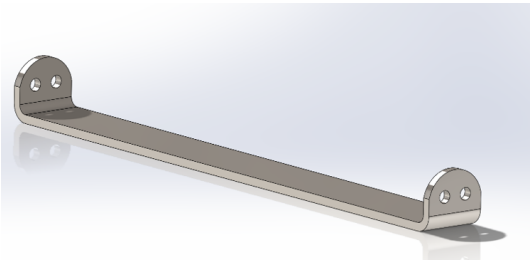


Figure 9: Isometric view of the link design.

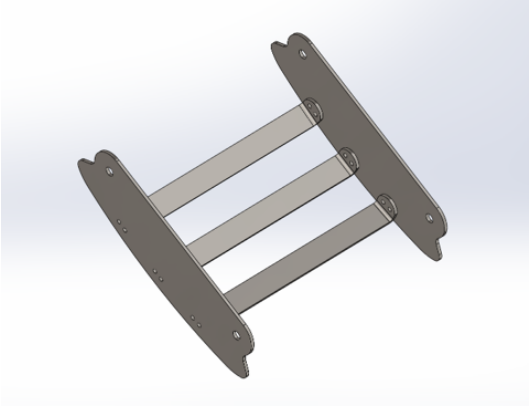


Figure 10: Isometric view of the complete unit design.

2.2.3 The Assembly

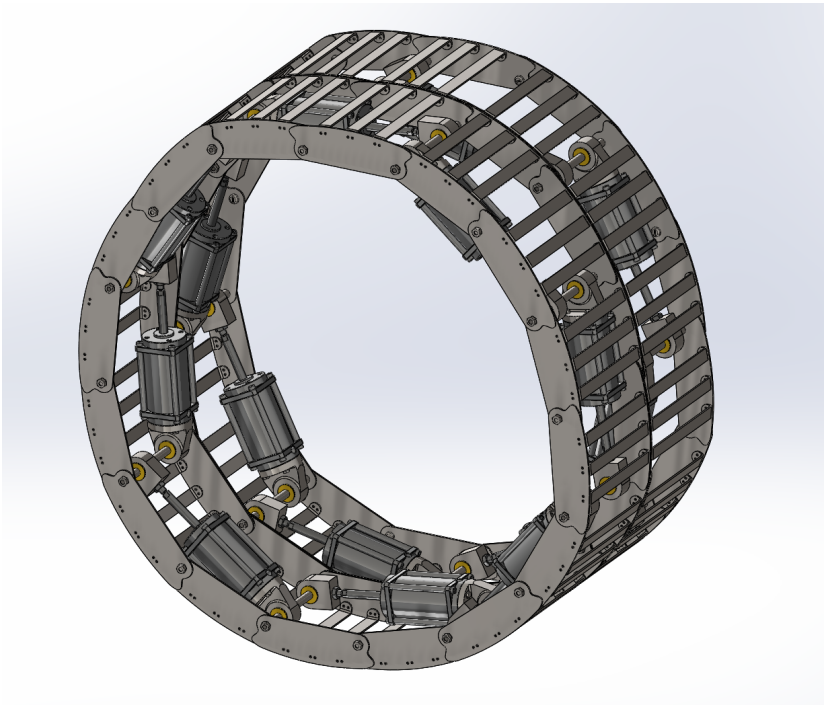


Figure 11: Detailed view of the assembly, showing spacers, and bearings.

Once the units were designed, we assembled them into a closed loop of sixteen

units arranged in two layers (Fig. 11). We used 6 mm shafts threaded at the ends, allowing them to be tightened with nuts. Screws and nuts were also used to connect two consecutive units whenever no shaft directly linked them. On each shaft, spacers were placed to properly align the actuator body and rod. In addition, flanged plain bearings were installed at the revolute joints to guide the rotation and prevent friction against the spacers (Fig. 12).

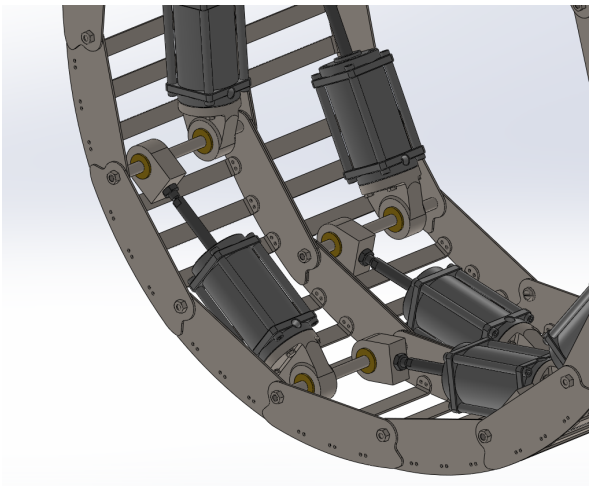


Figure 12: Detailed view of the assembly, showing shafts, spacers, and bearings.

3 Model-Based Control Using Center of Mass Jacobian

3.1 Notation and Kinematic Modeling

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 [2]. 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:

$$ {}^{i}\mathbf{G}_i = \begin{bmatrix} -\alpha L \\ 0 \end{bmatrix}, \tag{2} $$

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:

$$ {}^1\mathbf{G}_i = \begin{bmatrix} \sum_{k=2}^{i} \cos(\beta_k) - \alpha \cos(\beta_i) \\ \sum_{k=2}^{i} \sin(\beta_k) - \alpha \sin(\beta_i) \end{bmatrix}, \tag{3} $$

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

$$ \beta_k = \sum_{j=2}^{k} \theta_j. \tag{4} $$

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

$$ {}^1\mathbf{G} = \frac{\sum_{i=1}^{N} m\, {}^1\mathbf{G}_i}{\sum_{i=1}^{N} m}, \tag{5} $$

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

$$ {}^1\mathbf{G} = \frac{L}{N} \begin{bmatrix} -\frac{1}{2} + \sum_{i=2}^{N} \cos(\beta_i)\big(N - i + 1 - \alpha\big) \\ -\frac{1}{2} + \sum_{i=2}^{N} \sin(\beta_i)\big(N - i + 1 - \alpha\big) \end{bmatrix}. \tag{6} $$

From [6], we know that shifting the center of mass (CoM) can initiate a tipping motion which, combined with an approximately circular shape, enables the robot — with a diameter of about 0.28 m — to achieve significant speeds of up to 1.6 m/s [7]. 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.

3.2 Center of Mass Jacobian

The Jacobian matrix links the speeds of the joints to the movement speed of the robots center of mass (CoM). In our model, the CoMs 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:

$$J_G(\mathbf{q}) = \frac{\partial\, {}^1\mathbf{G}}{\partial \mathbf{q}}(\mathbf{q}) \tag{7}$$

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

$$\mathbf{q} = \left[\theta_1, \theta_2, \ldots, \theta_N\right]^\top. \tag{8}$$

By applying (7) to our CoM vector (6), we get the following equation:

$$J_G = [\mathbf{J}_1 \quad \ldots \quad \mathbf{J}_k \quad \ldots \quad \mathbf{J}_N] \in \mathbb{R}^{2\times N}, \tag{9}$$

with each column J_k given by:

$$\mathbf{J}_k = \frac{L}{N}\begin{bmatrix} -\sum_{i=k}^{N} \sin(\beta_i)(N-i+1-\alpha) \\ \sum_{i=k}^{N} \cos(\beta_i)(N-i+1-\alpha) \end{bmatrix}. \tag{10}$$

3.3 Control Design Based on CoM Jacobian

To achieve controlled rolling with a segmented annular robot, we need to shift its center of mass (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 [8], improving on basic sequential control methods [9], while keeping the robots 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:

$$\ddot{\mathbf{x}} = K_p\mathbf{e} + K_d\dot{\mathbf{e}}, \tag{11}$$

where $\mathbf{e}$ is the position error and $\dot{\mathbf{e}}$ its time derivative. The gain matrices K_p, K_d $\in \mathbb{R}^{2\times N}$ 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:

$$\ddot{\mathbf{q}} = J_G^{\#}\big(\ddot{\mathbf{x}} - \dot{J}_G\dot{\mathbf{q}}\big), \tag{12}$$

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:

$$\dot{J}_G = [\dot{\mathbf{J}}_1 \ \dots \ \dot{\mathbf{J}}_k \ \dots \ \dot{\mathbf{J}}_N], \tag{13}$$

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

$$\dot{\mathbf{J}}_k = -\frac{L}{N}\begin{bmatrix} \sum_{i=k}^{N} \dot{\beta}_i \cos(\beta_i)\big(N - i + 1 - \alpha\big) \\ \sum_{i=k}^{N} \dot{\beta}_i \sin(\beta_i)\big(N - i + 1 - \alpha\big) \end{bmatrix}. \tag{14}$$

3.4 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 [10], which allows secondary objectives to be fulfilled without interfering with the primary CoM control task. In particular, maintaining the robots circular shape is important for stable and efficient rolling locomotion [6]. 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:

$$P_{\text{null}} = I_N - J_G^{\#} J_G, \tag{15}$$

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 robots configuration $\mathbf{q}$ towards a reference circular configuration $\mathbf{q}_{\text{ref}}$:

$$\mathbf{z} = K_{\text{null}}(\mathbf{q}_{\text{ref}} - \mathbf{q}), \tag{16}$$

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 :

$$\ddot{\mathbf{q}} = J_G^{\#}\big(\ddot{\mathbf{x}} - \dot{J}_G\dot{\mathbf{q}}\big) + P_{\text{null}}\mathbf{z}. \tag{17}$$

3.5 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 robots 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 joints 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 τ using:

$$\boldsymbol{\tau} = I\ddot{\mathbf{q}}, \tag{18}$$

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.

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

$$\mathbf{F} = \boldsymbol{\tau} \oslash \mathbf{L}_{\text{lever}}, \qquad \boldsymbol{\tau},\, \mathbf{L}_{\text{lever}} \in \mathbb{R}^N, \tag{19}$$

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

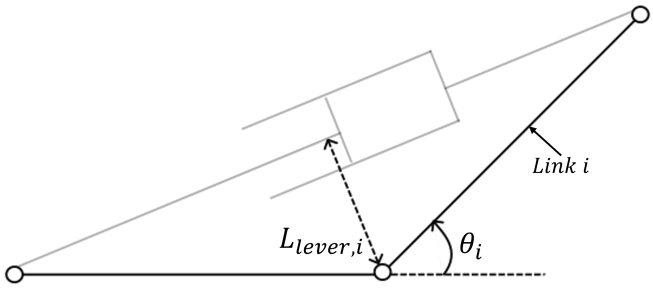


Figure 13: Schematic of the lever arm geometry.

As illustrated in Fig. 13, L_{lever} is the perpendicular distance from the joint axis to the actuators 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:

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

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:

$$L_{\text{lever},i} = L\left|\sin\left(\frac{\theta_i}{2}\right)\right|, \tag{21}$$

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:

$$\mathbf{P} = \frac{\mathbf{F}}{S}. \tag{22}$$

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 (18), (19), and (22):

$$u_i = \frac{I}{L_{\text{lever},i}S}\ddot{q}_i, \tag{23}$$

where $\ddot{q}_i$ is the angular acceleration of link i from (17).

4 Simulation

4.1 Simulink® Model Architecture

To validate the proposed control strategy and assess the robots rolling performance, a simulation environment was developed using MATLAB® Simulink® and Simscape®, as shown in Fig. 14. 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.

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. 15, highlighting the main signal flow and interactions between these subsystems. Figs. 16, 17, and 18 are simplified diagrams of the corresponding Simulink® blocks.

The *mechanism subsystem* models the physical interactions among the solid components (Fig. 16). 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 *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 elementseither directly or through rigid transformsallowing us to determine the CoM of the entire robot, $\mathbf{G}_{\text{mes}}$. This information

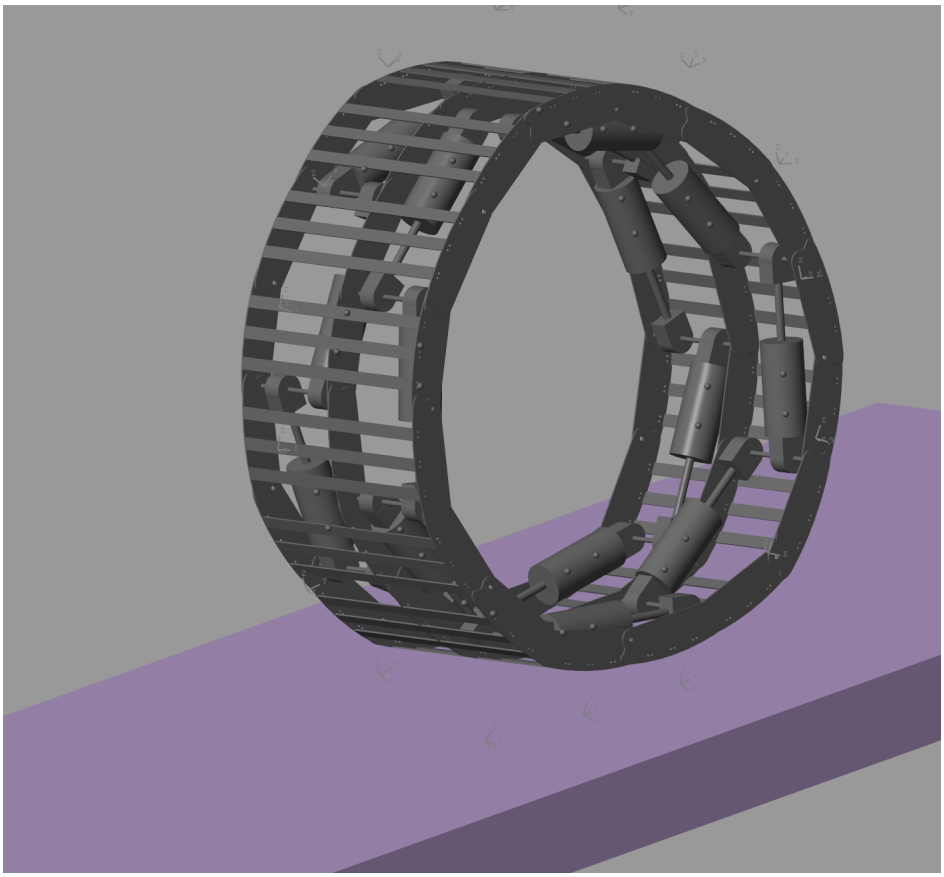


Figure 14: Visualization of the annular robot in the MATLAB® Simulink® simulation environment. The robot has a height of $h = 0.7\,\mathrm{m}$ and a width of $w = 0.3\,\mathrm{m}$.

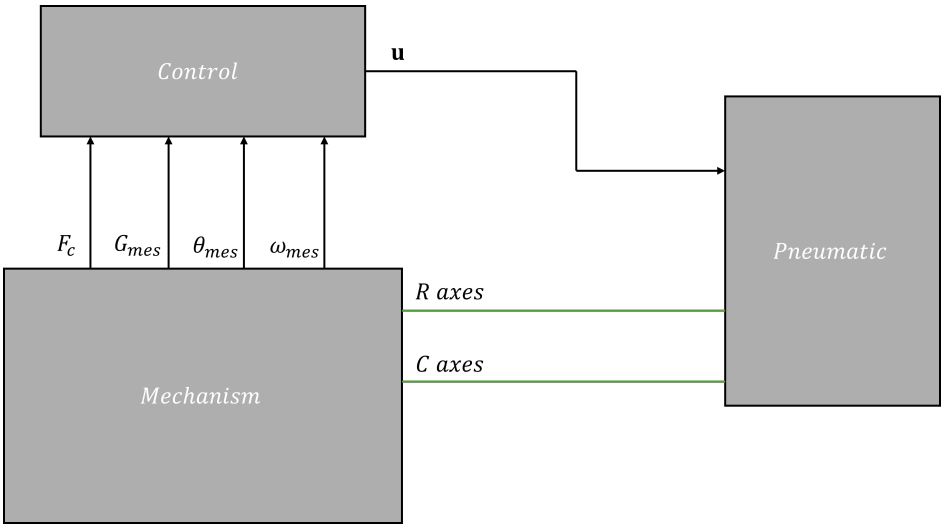


Figure 15: Main subsystems of the Simulink® model. C denotes the cylinders, and R denotes the rods.

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*.

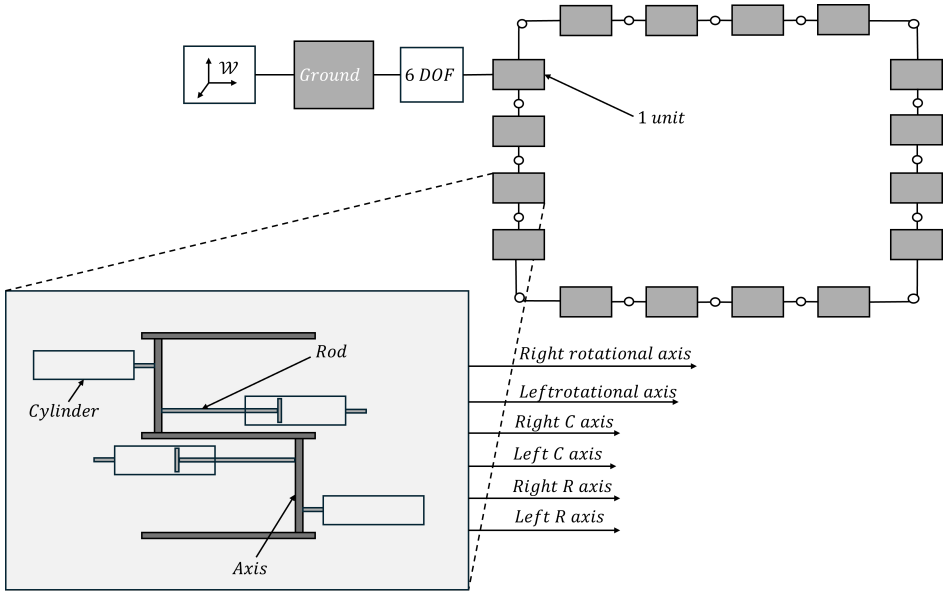


Figure 16: Details of the mechanism subsystem in the Simulink® model. The zoomed-in section shows a top view of a single unit. C denotes the cylinders, and R denotes the rods. Black arrows represent Simulink® signals (vectors), green lines denote axes used for physical connections, and purple dashed lines indicate air pressure signals.

The *pneumatic subsystem* consists of sixteen blocks modeling each of the pneumatic actuators. As shown in Fig. 17, each block represents a springless, single-acting pneumatic cylinder, as mentioned in Sec. II. Within each block, as illustrated in Fig. 17, 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. 18).

The first step 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

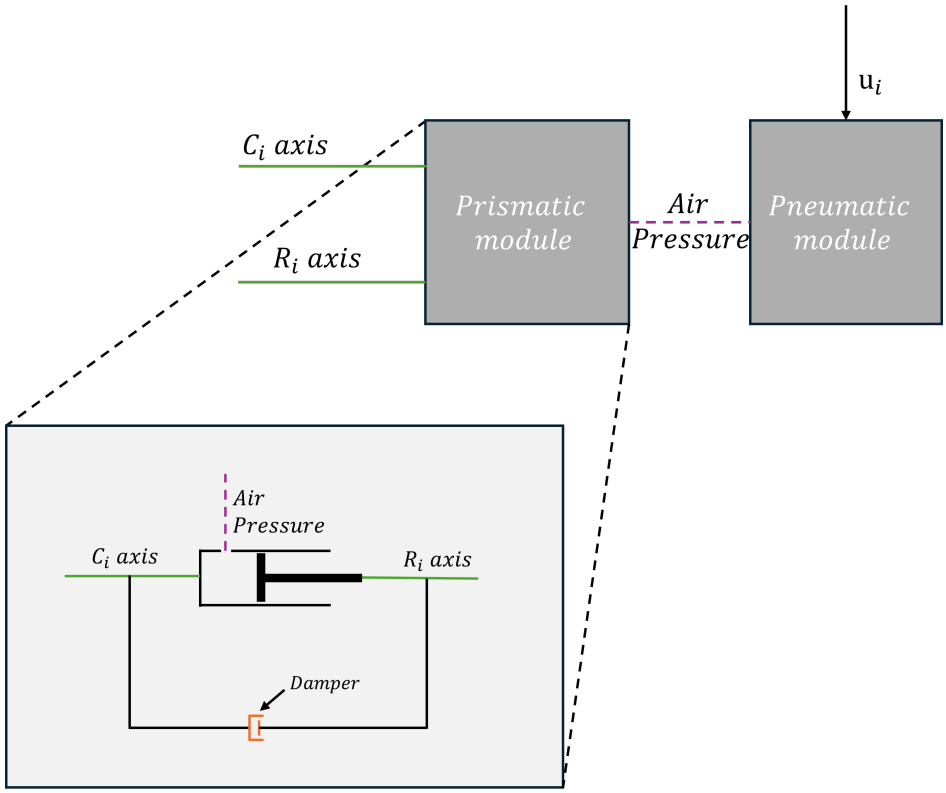


Figure 17: Details of the pneumatic subsystem in the Simulink® model. C_i denotes the i-th cylinder, and R_i denotes the i-th rod. The color code is the same as Fig. 16.

incorporated in the feedback loop. Finally, the function outputs the control command **u**.

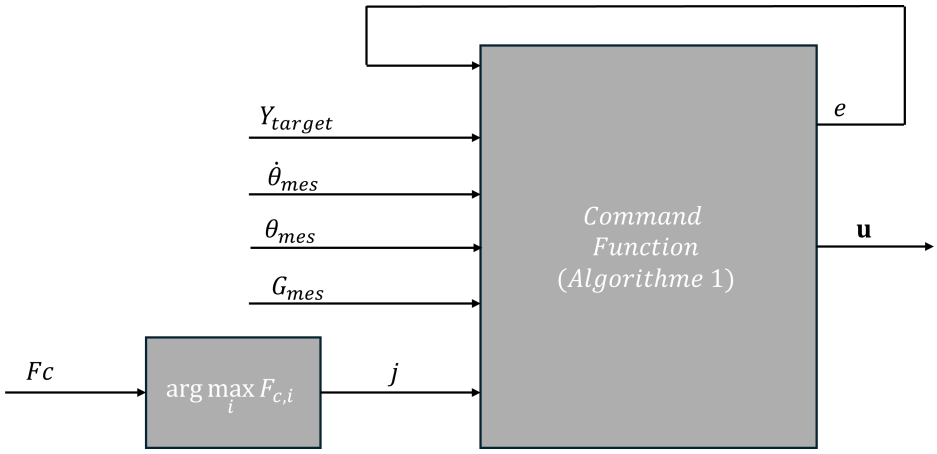


Figure 18: Details of the control subsystem in the Simulink® model. The color code is the same as Fig. 16.

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 time derivative

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

`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 (23), 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.

4.2 Results

Fig. 19 compares the robots 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 one, designed to preserve the robots 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 by using the No nullspace strategy, it only reaches about 2.3 m over the same period. After the transient response is over, both configurations converge to the 2.5 m target, but their regulation behavior differs: 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 nullspace control not only enhances motion stability but also improves propulsion efficiency.

The robots 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.

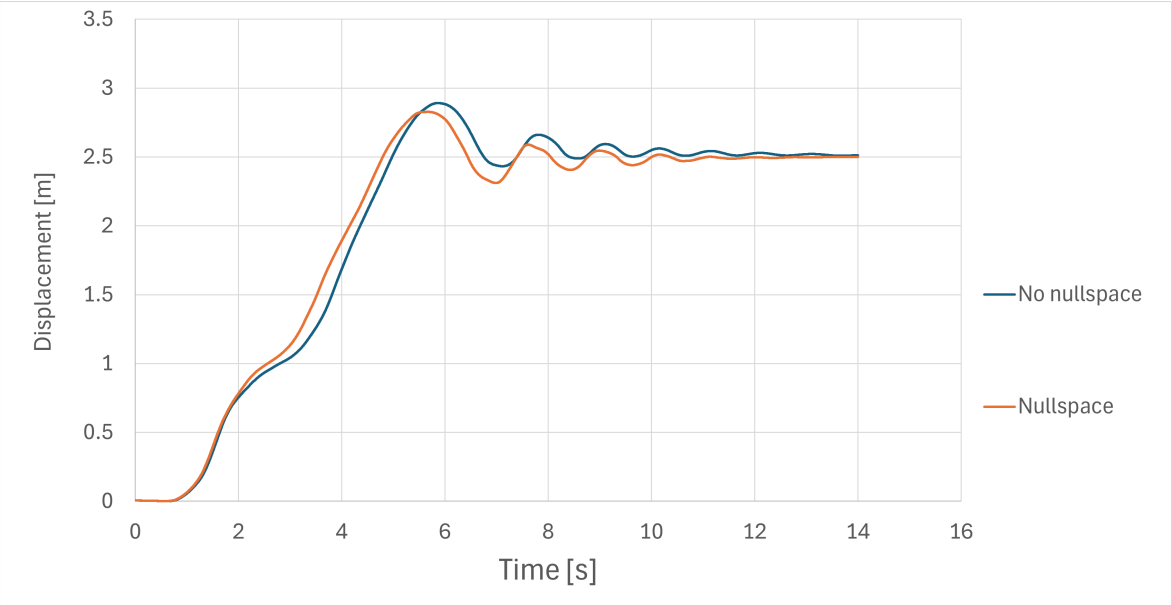


Figure 19: Time evolution of the robots displacement for two control configurations. The *No nullspace* strategy corresponds to $K_{\text{null}} = 0$, whereas the *Nullspace* strategy uses $K_{\text{null}} \neq 0$.

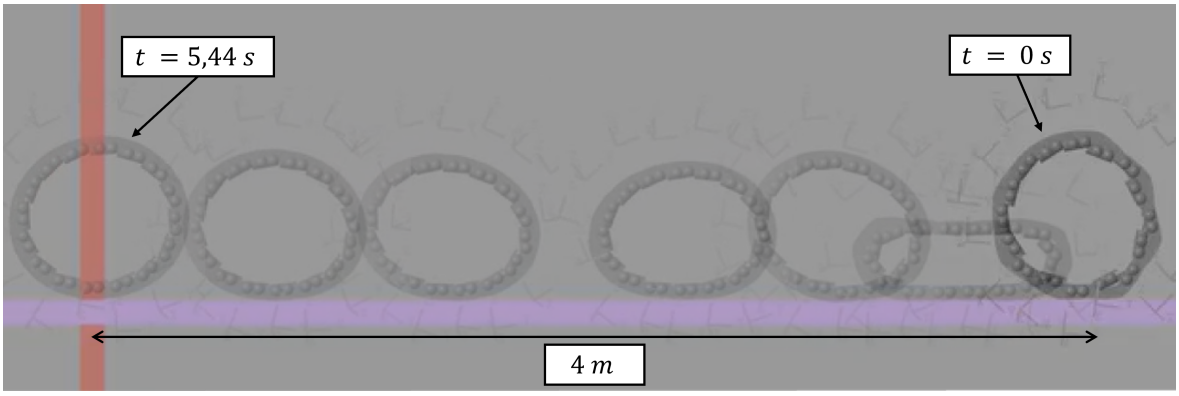


Figure 20: Superimposed frames from the simulation showing the robots motion over time. Each frame corresponds to a 1-second interval, except for the final frame.

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. 20, and considering the robots diameter of approximately 0.7 m, the robot is capable of covering nearly 4 meters in under 6 seconds using this control strategy.

5 Updating the Dynamic Model

In the previous simulations, we relied on a coarse acceleration-to-torque approximation. To improve accuracy, we implemented a new simulation based on a more realistic

dynamic model. However, the robot is complex due to its large number of joints and closed-loop structure. Therefore, we adopted a numerical method to derive the dynamic equations, which can be expressed as:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} + J_h^T \boldsymbol{\lambda}, \tag{24}$$

where
- $M(\mathbf{q})$ is the mass (inertia) matrix,
- $C(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis and centrifugal matrix,
- $\mathbf{G}(\mathbf{q})$ is the gravity vector,
- $\boldsymbol{\tau}$ is the vector of input joint torques,
- J_h is the Jacobian matrix of the holonomic constraints,
- $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers associated with these constraints.

5.1 The Jacobian Matrix of the Holonomic Constraints

The holonomic constraints for the closed-loop structure arise from the requirement that the end-effector returns to its initial position, forming a closed kinematic chain. These constraints can be expressed through a constraint function $\boldsymbol{\phi}(\mathrm{q}) = 0$, which represents the closure conditions. For our planar chain, the constraint function is defined as:

$$\boldsymbol{\phi}(\mathbf{q}) = \sum_{k=1}^{N} L \begin{bmatrix} \cos(\beta_k) \\ sin(\beta_k) \end{bmatrix} = \mathbf{0} \tag{25}$$

This constraint function ensures that the sum of all link vectors equals zero, effectively closing the loop. The constraint Jacobian J_h is obtained by differentiating the constraint function with respect to the joint coordinates:

$$J_h = \frac{\partial \boldsymbol{\phi}(\mathbf{q})}{\partial \mathbf{q}} \tag{26}$$

This Jacobian captures how changes in joint angles affect the constraint violations. For our system, the explicit form of the constraint Jacobian is:

$$J_h = [\mathbf{J}_{h,1} \quad \dots \quad \mathbf{J}_{h,k} \quad \dots \quad \mathbf{J}_{h,N}] \in \mathbb{R}^{2\times N}, \tag{27}$$

with each column $\mathbf{J}_{h,k}$ defined by:

$$\mathbf{J}_{h,k} = \begin{bmatrix} -\sum_{l=k}^{N} L\sin(\beta_l) \\ \sum_{l=k}^{N} L\cos(\beta_l) \end{bmatrix}, \tag{28}$$

5.2 The Mass Matrix

The mass matrix was computed using a Jacobian-based formulation that accumulates the contributions of each links translational and rotational kinetic energy. For each link i, the translational part is obtained from the linear velocity Jacobian J_{v_i}, while

the rotational part is obtained from the angular velocity Jacobian J_{w_i}. The contribution of a link with mass m and inertia I is thus given by $mJ_{v_i}^T J_{v_i} + IJ_{w_i}^T J_{w_i}$.

The explicit forms of these Jacobians in our planar chain are:

$$J_{v_i} = [\mathbf{J}_{v_{i,1}} \quad \dots \quad \mathbf{J}_{v_{i,k}} \quad \dots \quad \mathbf{J}_{v_{i,N}}] \in \mathbb{R}^{2\times N}, \tag{29}$$

with each column $\mathbf{J}_{v_k}$ defined by:

$$\mathbf{J}_{v_{i,k}} = \begin{bmatrix} -\sum_{l=k}^{i} L\sin(\beta_l) + \frac{L}{2}\sin(\beta_i) \\ \sum_{l=k}^{i} L\cos(\beta_l) - \frac{L}{2}\cos(\beta_i) \end{bmatrix}, \quad \text{for } k \le i, \tag{30}$$

$$J_{w_i} = [\mathbf{J}_{w_{i,1}} \quad \dots \quad \mathbf{J}_{w_{i,k}} \quad \dots \quad \mathbf{J}_{w_{i,N}}] \in \mathbb{R}^{1\times N}, \tag{31}$$

with each column $\mathbf{J}_{w_k}$ given by:

$$\mathbf{J}_{w_{i,k}} = \begin{cases} 1, & k \ge i, \\ 0, & k < i. \end{cases} \tag{32}$$

where L is the link length, β_i is the joint angle of link i, and N is the total number of joints. Summing these contributions over all links yields the unconstrained joint-space inertia matrix. To account for the closed-loop structure, we additionally incorporate a penalty term based on the holonomic constraint Jacobian J_h, giving

$$M = \sum_{i=1}^{N} \left(mJ_{v_i}^T J_{v_i} + IJ_{w_i}^T J_{w_i} \right) + \mu J_h^T J_h, \tag{33}$$

where μ is a weighting factor enforcing the constraints. This approach approximates the constrained dynamics while keeping the mass matrix computation straightforward and numerically tractable [11].

5.3 The Coriolis and Centrifugal Matrix

The Coriolis and centrifugal matrix $C(\mathbf{q}, \dot{\mathbf{q}})$ captures the velocity-dependent forces arising from the motion of the manipulator links. For planar manipulators, the angular velocity components contribute only constant terms to the dynamics, allowing the Coriolis matrix to be expressed solely in terms of the linear velocity contributions. This simplification leads to a computationally efficient factorization that leverages the kinematic structure of the planar chain. Following the approach presented in Bjerkeng and Pettersen's work [12], the Coriolis matrix for our planar closed-chain system is computed using the factorization:

$$C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \sum_{i=1}^{N} m_i J_{v_i}^T \dot{J}_{v_i} \dot{\mathbf{q}} \tag{34}$$

where J_{v_i} is the linear velocity Jacobian for link i (as defined in the previous subsection) and $\dot{J}_{v_i}$ represents its time derivative. This formulation explicitly shows that for planar

manipulators, only the linear velocities contribute to the Coriolis/centripetal force, significantly simplifying the computation compared to spatial manipulators. The time derivative of the velocity Jacobian $\dot{J}_{v_i}$ is computed by differentiating the kinematic relationships with respect to time:

$$\dot{J}_{v_i} = [\dot{\mathbf{J}}_{v_{i,1}} \quad \dots \quad \dot{\mathbf{J}}_{v_{i,k}} \quad \dots \quad \dot{\mathbf{J}}_{v_{i,N}}] \in \mathbb{R}^{2\times N}, \tag{35}$$

with each column $\dot{\mathbf{J}}_{v_k}$ defined by:

$$\dot{\mathbf{J}}_{v_{i,k}} = \begin{bmatrix} -\sum_{l=k}^{i} L\cos(\beta_l)\dot{\beta}_l + \frac{L}{2}\cos(\beta_i)\dot{\beta}_i \\ -\sum_{l=k}^{i} L\sin(\beta_l)\dot{\beta}_l + \frac{L}{2}\sin(\beta_i)\dot{\beta}_i \end{bmatrix}, \quad \text{for } k \leq i, \tag{36}$$

This factorization maintains the important skew-symmetric property $\dot{M} = C + C^T$ required for stability analysis and control design, while providing a clear connection between the Coriolis forces and the underlying kinematic structure of the planar chain. The resulting matrix is sparse and computationally efficient to evaluate, making it well-suited for real-time control applications.

5.4 The Gravity Vector

The gravity vector $\mathbf{G}$ accounts for the gravitational forces acting on each link of the manipulator. For our planar closed-chain system, the gravity vector is computed by summing the contributions from all links, where each link's gravitational effect is determined by its mass and the velocity Jacobian evaluated at its center of mass. The explicit form of the gravity vector is given by:

$$\mathbf{G}(\mathbf{q}) = \sum_{k=1}^{N} J_{v,k}^T m_k g_{vec} \tag{37}$$

where $g_{vec} = [0, -9.81]^T$ represents the gravitational acceleration vector in the planar coordinate system. The velocity Jacobian $J_{v,k}$ for each link k relates the joint velocities to the linear velocity of the link's center of mass, and its transpose $J_{v,k}^T$ maps the gravitational forces back to the joint space. This formulation ensures that the gravity vector properly captures the potential energy contributions of all links in the kinematic chain, providing the necessary torques to counteract gravitational effects during motion. The resulting gravity vector is configuration-dependent and varies with the joint angles $\mathbf{q}$, reflecting how the gravitational load distribution changes as the manipulator moves through different poses.

5.5 Computing the Joint Torques

The presence of holonomic constraints makes the direct inversion of the unconstrained dynamics insufficient, since the resulting accelerations may violate the closure conditions. To resolve this, the dynamics and constraint equations are combined into the

augmented system

$$\begin{bmatrix} M & J_h^T \\ J_h & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} - \mathbf{h} \\ -\dot{J}_h \dot{\mathbf{q}} \end{bmatrix}, \tag{38}$$

which enforces both the equations of motion and the acceleration-level constraints simultaneously. The block structure ensures that the solution lies in the admissible subspace of motions, while the multipliers $\boldsymbol{\lambda}$ capture the internal forces required to maintain loop closure.

From this system, the constraint forces are obtained by eliminating $\ddot{\mathbf{q}}$:

$$\boldsymbol{\lambda} = (J_h M^{-1} J_h^T)^{-1} \Big(J_h M^{-1} (\mathbf{h} - M\ddot{\mathbf{q}}) + \dot{J}_h \dot{\mathbf{q}} \Big), \tag{39}$$

and the physically consistent torques then follow as

$$\boldsymbol{\tau} = \mathbf{h} + M\ddot{\mathbf{q}} - J_h^T \boldsymbol{\lambda}. \tag{40}$$

This approach guarantees that the applied torques not only generate the target accelerations but also inherently satisfy the holonomic constraints by compensating for the corresponding reaction forces. In other words, the multipliers act as internal forces that do not alter the motion in unconstrained directions but preserve the closed-loop structure.

A detailed treatment of this constrained dynamics formulation can be found in Featherstones *Rigid Body Dynamics Algorithms* [13] and in Spong, Hutchinson, and Vidyasagars *Robot Modeling and Control* [14].

5.6 Results

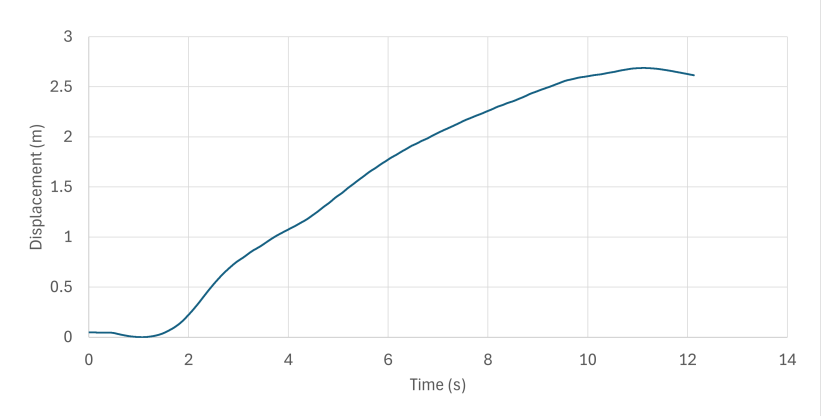


Figure 21: Displacement evolution using the updated dynamic model.

In this simulation, the displacement obtained with the updated dynamic model is shown in Figure 21. Compared to the previous approximation-based method, the system evolves more slowly and reaches its steady value after a longer transient. This indicates that the new approach still requires careful tuning and refinement. However, the main advantage of this formulation is that the computation of the joint torques from the desired accelerations is carried out in a more mathematically rigorous way. Although the performance is currently less responsive than before, this method provides a more accurate foundation for representing the robots true dynamics.

6 Current Development

6.1 Electronics

One of the main issues of the simulation is that it assumes the use of sensors that are not actually available in practice. In fact, the simulated robot measures the relative angles and angular velocities between each link, the normal contact forces between the ground and each link, and even the position of the center of mass (CoM) of each unit, from which the global CoM position can be deduced. However, in reality it is not feasible to equip the robot with such a complete set of sensors. Instead, we restricted ourselves to a configuration closer to the original HIDAS setup, which relies mainly on IMUs and force sensors. Since no physical sensor directly provides the CoM position, we must rely on an observer to estimate the displacement ourselves. The design and implementation of this observer will be discussed in detail later in the report.

For the experimental setup, we selected a set of electronic components that balances practicality with compatibility to our sensing needs. The chosen hardware is summarized below:

- **STMicroelectronics SensorTile (STEVAL-STLCS01V1)**: This board integrates motion sensors and provides analog inputs, which is essential for connecting the Force Sensitive Resistors (FSRs). Its compact design and versatility make it a suitable choice for our application.
- **FSR Series**: Force Sensitive Resistors are used to measure the normal contact forces at the robots feet. They are lightweight, low-cost, and easy to integrate with the SensorTile thanks to its analog input capability.
- **STM32 Microcontroller**: The STM32 family provides high computational power with low energy consumption. It is widely supported and ensures reliable real-time processing for sensor data and control algorithms.
- **MUX $\mathbf{I^2C}$**: Since the SensorTile chips only allow two I^2C addresses, while our design requires up to sixteen SensorTiles, an I^2C multiplexer is necessary. This component allows us to expand the bus and handle all the sensors simultaneously while keeping wiring complexity manageable.

This combination ensures that we can capture motion and force data reliably, while maintaining flexibility for integration and scalability of the sensing system.

6.2 Kalman Filter

Assuming we are using the IMUs and force sensors we mentioned before, we tried to develop an Unscented Kalman Filter (UKF) using only the measurements of the accelerometers, the angular velocities of each link, and the force sensors. However, we ended up finding that the system would not be observable in that case. Therefore, in our model we are assuming we are also using potentiometers in order to measure the relative angles.

The UKF is then constructed around the state vector

$$x = \begin{bmatrix} y_{com} & z_{com} & \theta & \dot{\theta} \end{bmatrix}^T, \tag{41}$$

which evolves according to the continuous-time dynamics

$$\dot{x} = \begin{bmatrix} J \times \dot{\theta} \\ \dot{\theta} \\ h(u) \end{bmatrix}, \tag{42}$$

where J relates the angular velocity to the CoM velocity, and the function $h : \mathbb{R}^N \to \mathbb{R}^N$ represents the nonlinear dynamics of the system under control inputs.

For a controller output $u \in \mathbb{R}^N$, $h(u)$ maps the input through the following transformations:

$$u \text{ (\% of pressure)} \longrightarrow \mathbf{P} \longrightarrow \boldsymbol{\tau} \longrightarrow \ddot{\boldsymbol{\theta}}.$$

The discrete-time process model is then written as

$$x_{n+1} = x_n + \Delta t \cdot \dot{x}_n. \tag{43}$$

The measurement model is defined by

$$y_n = C\, x_n, \tag{44}$$

$$C = \begin{bmatrix} \mathbf{0}_{16\times 2} & I_{16} & B \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & \cdots & 0 & -1 \\ -1 & 1 & \cdots & 0 & 0 \\ 0 & -1 & \ddots & 0 & 0 \\ \vdots & & \ddots & 1 & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{16\times 16} \tag{45}$$

where the matrix C encodes the relationship between the state vector and the available sensor data (angular velocities and relative angles). In particular, the structure of C captures the fact that we have 16 sensor measurements combining the IMUs and potentiometers, but only a reduced number of independent state variables.

Within the UKF framework, sigma points are propagated through the nonlinear process function and then mapped through the measurement model using C. The filter then corrects the state estimate by comparing the predicted sensor outputs with the actual measurements. This formulation allows us to reconstruct unmeasured quantities such as the displacement of the CoM while accounting for sensor noise and model uncertainties.

Unfortunately, we did not have enough time to complete the implementation of this UKF. Our previous attempts using a similar approach resulted in the state estimates diverging, highlighting the challenges of making the system observable with the current sensor setup.

7 Challenges Encountered During the Internship

During this internship, I faced several challenges due to both my previous experience and the complexity of the project:

- **Learning new tools:** I had never used Simulink or Simscape Multibody before. The project required building a complex articulated robot, which involved

modeling multiple joints, sensors, and actuators. I had to learn how to set up the multibody simulations, define joint constraints, and manage the interactions between different components. This steep learning curve was a significant challenge at the beginning of the internship, but it allowed me to become proficient in tools that I had never explored before.

- **Taking over an existing project:** I joined the internship at a stage where the project had already been started by a previous intern. Catching up with the existing work was challenging, as I had to understand the modeling choices, the existing code, and the simulation setup. Additionally, our approaches to problem-solving were different, which required me to adapt and find a workflow that integrated both perspectives. This experience helped me develop skills in quickly understanding complex projects and collaborating effectively, even when there are differences in methodology.
- **Learning articulated robot theory from scratch:** My previous studies mainly focused on control theory for drone navigation, so I had very limited knowledge of articulated robots. I needed to learn concepts such as forward and inverse kinematics, Jacobians, inverse dynamics, and trajectory planning from scratch. This involved extensive literature review, studying tutorials, and applying theoretical concepts to practical simulations. While it was challenging to acquire this knowledge in a short time, it was also very rewarding and allowed me to broaden my expertise in robotics

8 Conclusions

This internship at the CNRS-AIST Joint Robotics Laboratory provided an invaluable opportunity to work on cutting-edge robotics research within Japan's ambitious Moonshot program. The project challenged me to develop a rigid analog of the HIDAS inflatable robot, combining theoretical control design with practical simulation implementation.

During this internship, I successfully redesigned the mechanical architecture for a more compact and symmetric structure, developed a kinematic model based on center of mass manipulation, and implemented model-based control strategies using Jacobian methods. The simulation environment in Simulink demonstrates the feasibility of the approach, with the robot achieving rolling locomotion through coordinated pneumatic actuation while maintaining a near-circular shape.

The transition from simplified dynamics to a more rigorous formulation incorporating holonomic constraints, Coriolis forces, and gravity effects represents a significant step toward realistic modeling. While this updated model requires further tuning, it provides a mathematically sound foundation for future development and eventual hardware implementation.

Working independently on this complex project pushed me far beyond my initial comfort zone. Having previously focused on drone control systems, I had to rapidly acquire expertise in articulated robot kinematics, multibody dynamics, and pneumatic actuation systems. The steep learning curve with Simulink and

Simscape Multibody, combined with the challenge of taking over and restructuring an existing project, demanded both technical adaptability and problem-solving resilience.

The incomplete implementation of the Unscented Kalman Filter and ongoing challenges with sensor integration remind us that significant work remains before hardware validation. Future work should focus on completing the observer design, refining the dynamic model parameters, and beginning hardware prototyping.

This internship has been transformative both professionally and personally. It expanded my technical expertise into new areas of robotics, provided insight into international research collaboration, and offered the opportunity to contribute to Japan's Moonshot program. Additionally, submitting my first academic article to the SII conference marked an important milestone in transitioning from student coursework to contributing original research.

The project's connection to future lunar habitats adds particular significance. While the immediate focus was on a terrestrial prototype, the ultimate vision of inflatable robotic structures supporting human activities beyond Earth provides compelling motivation for continued research.

List of Figures

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. 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.

[3] 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.

[4] 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.

[5] Y. Wang, C. Wu, L. Yu, and Y. Mei, "Dynamics of a rolling robot of closed five-arc-shaped-bar linkage," *Mechanism and Machine Theory*, vol. 121, pp. 75–91, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094114X17302379

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] A. S. Sathya and J. Carpentier, "Constrained articulated body dynamics algorithms," *IEEE Transactions on Robotics*, 2024, conditionally accepted preprint.

[12] M. Bjerkeng and K. Y. Pettersen, "A new coriolis matrix factorization," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 4974–4979.

[13] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2008.

[14] M. W. Spong, S. Hutchinson, M. Vidyasagar *et al.*, *Robot modeling and control*. Wiley New York, 2006, vol. 3.