

Design and Development of a Modular Center-Articulated Multi-modal Inspection Robot for Complex and Confined Environments

A Report

By

K A Vyvaswath

Research Intern

Robotics Innovations Lab

Department of Design and Manufacturing

Indian Institute of Science

May 2025

Abstract

This report presents the design, modeling, and prototyping of a novel modular, center-articulated, multi-modal robotic platform developed for inspection tasks in complex and confined environments such as pipelines and uneven terrains. The robot features two driven modules connected via a novel two-degree-of-freedom (2-DoF) joint. A wall-press mechanism with both passive and active components allows adaptive operation in pipes of varying diameters, while a configurable wheel distance mechanism enhances ground stability. Kinematic, dynamic, and static force models of the robot have been developed and presented. The dynamic behavior of the robot is modeled using the Lagrangian approach, and simulation studies under different torque and damping conditions are presented. A functional prototype was fabricated using a combination of 3D printing and precision machining, and it was evaluated in diverse testing scenarios including straight pipes, bends, and uneven ground. The robot demonstrated robust mobility, adaptable configuration, and efficient joint articulation.

Contents

1	Introduction	4
2	Design	7
2.1	Overview of the Design	7
2.2	Initial Design and Limitations	8
2.3	Design Modifications and Improvements	9
2.3.1	Locomotion Mechanism	9
2.3.2	Configurable Wheel Distance Mechanism	11
2.3.3	2 DoF Joint	11
3	Robot Modeling	13
3.1	Kinematics	13
3.2	Dynamics	16
3.3	Mechanism Modeling	18
4	Simulation and Results	22
4.1	Case 1: Free Response	22
4.2	Case 2: Simulation with Wheel Torque and No Joint Damping	24
4.3	Case 3: Simulation with Wheel Torque and Joint Damping	25
4.4	Case 4: Simulation with Wheel and Joint Actuation under Damped Conditions	27
5	Assembly and Testing	30
5.1	Fabrication	30
5.2	Electronics	34
5.3	Prototype Development and Testing	36
6	Conclusion	38
7	Future Scope	38
8	References	40
9	Appendix	42
9.1	Derivation of Dynamics Equations using Lagrangian Approach	42
9.2	Dynamics Simulation Codes	45
9.2.1	Simulation MATLAB File: Simulate	45
9.2.2	Simulation MATLAB File: Dynamics	48

List of Figures

2.1	CAD Model of the Robot	7
2.2	Sectional View of the Robot	8
2.3	Wall Press Mechanism	8
2.4	Wheel Torque Simulation	10
2.5	Locomotion Mechanism	11
2.6	Joint Torque Simulation	12
2.7	2 DoF Joint Module	12
2.8	Robot with the 2 DoF Joint	13
3.1	Robot Coordinate Frames	13
3.2	Forces in the Wall Press Mechanism	19
3.3	Normal Forces acting on the robot	20
4.1	Case 1 - Robot trajectory	23
4.2	Case 1 - Angle variations during simulation.	23
4.3	Case 1 - Total energy of the system over time.	24
4.4	Case 2 - Robot trajectory	24
4.5	Case 2 - Angle Variations during Simulation	25
4.6	Case 2 - Total energy of the system over time.	25
4.7	Case 3 - Robot trajectory	26
4.8	Case 3 - Angle Variations during Simulation	26
4.9	Case 3 - Total energy of the system over time.	27
4.10	Case 4 - Robot trajectory	28
4.11	Case 4 - Angle Variations during Simulation	28
4.12	Case 4 - Total energy of the system over time.	29
5.1	Fabricated Leg	31
5.2	Wall Press Mechanism with Lead Screw	32
5.3	2 DoF Joint and Assembly	34
5.4	System Architecture	35
5.5	Fabricated Robot	36
5.6	Prototype Testing in Multiple Environments	37

1 Introduction

Inspection robots are robotic systems capable of navigating and inspecting various environments autonomously, without requiring human intervention. These robots are essential due to their ability to access areas that are otherwise challenging or impossible for humans to reach. Moreover, they enhance operational efficiency by performing tasks without fatigue. One critical application for such robotic systems is the inspection of pipelines. Pipelines require continuous monitoring to address issues such as clogging, aging, cracks and corrosion. Research in in-pipe inspection robots focuses on reducing inspection and maintenance costs while minimizing the labor intensity for workers.

In various industrial sectors, particularly those involving pipelines for gas, water, sewage, and air conditioning, the maintenance and inspection of these conduits are critical yet challenging tasks. Pipelines often traverse underground or within complex infrastructures, making external inspection difficult and costly. Traditional inspection methods face limitations in navigating multiple bends, T-junctions, and vertical sections, often failing to traverse beyond a few curves. This necessitates the development of advanced in-pipe inspection robots capable of maneuvering through complex pipeline networks. For pipelines consisting only of straight sections, no special mechanism is required for steering the robot. However, most pipelines incorporate bends or curved sections to divert the direction of the transported materials. In such environments, a robot equipped with a steering joint becomes essential.

Pipeline robots use various mechanisms to navigate pipes [1]. The locomotion mechanism, combined with the wall-press mechanism, enables these robots to traverse pipes of varying diameters and conditions. Based on their locomotion mechanisms, pipeline robots can be broadly categorized into wheeled robots, tracked robots, and bio-inspired robots. Wheeled robots are among the most common types used for pipeline inspection. They typically operate using rectilinear motion or screw-type motion. In the rectilinear type, the wheels propel the robot directly along the pipe axis. In screw-type motion the wheels are mounted at an angle to the pipe axis, allowing the robot to achieve a helical or screw-like motion. Tracked robots use tracks instead of wheels, which increases

the contact area with the pipe walls, providing more stability and support. Bio-inspired robots mimic the locomotion mechanisms of natural organisms, such as snakes, worms, or caterpillars, to adapt to the complex geometry of pipelines [2]. Wheeled robots are valued for their simplicity, speed, and energy efficiency.

Wheeled pipe robots incorporate a wall-press mechanism that generates sufficient normal force and traction. This mechanism, in combination with an adjustment mechanism, enables the robot to traverse pipes with varying diameters. Variations in pipe diameter often result from factors such as structural cracks or the accumulation of debris. These diameter variations directly influence the normal force exerted by the preloaded spring within the wall-press mechanism [3]. Such deviations can compromise the robot's traction and overall stability. To address this, the adjustment mechanism actively compensates for these changes, ensuring the normal force remains within optimal ranges. This integration allows wheeled robots to adapt effectively to dynamic pipeline conditions, maintaining reliable operation.

Center articulation, a concept utilized in ground-based vehicles like Load-Haul-Dump (LHD) machines, involves two bodies connected by an active joint, enabling steering through body articulation [4]. This design allows for sharp turning capabilities, which are advantageous in confined spaces. In mobile robotics, center-articulated designs have been explored to enhance maneuverability [5][6]. In multi-axis wheeled vehicles or car-like robots, when the steering angle is not zero, each wheel follows a distinct path. This becomes problematic in pipelines or narrow corridors, where maneuvering room is limited. Without articulation, these robots require more space to turn compared to their articulated counterparts.

There have been numerous studies focused on the development of pipeline inspection robots. Authors in [7] designed a tracked robot with an active adjustment strategy, which also demonstrated the capability to alter its attitude within the pipeline. Similarly, [8] introduced a modular walking device (MWD) featuring adjustable wheel brackets, Mecanum wheels, and a variable diameter mechanism to accommodate pipelines with varying diameters and complex geometries. The MWD can execute rectilinear, rotational,

and helical motions, allowing it to navigate obstacles up to 15 mm with.

While extensive research has been conducted on pipeline inspection robots, existing designs are often limited to structured environments and lack adaptability to unstructured terrains. Most robots are either specialized for in-pipe navigation or ground locomotion, requiring multiple systems for comprehensive inspection. In contrast, the developed robot in discussion introduces a multi modal locomotion capability, seamlessly transitioning between structured (pipes) and unstructured (ground) environments. It also consists of a 2-DoF freedom joint that combined the principles of center articulation with a gear driven actuation system. Moreover, its multi-modal functionality allows for effective ground-based locomotion steering, much like LHD vehicles [4], but with pitch and roll control, making it a robust and scalable solution for diverse environments. This enables it to navigate through pipelines while also traversing open terrain for environmental monitoring. By eliminating the need for separate specialized robots, it enhances operational efficiency and broadens the scope of real-world applications.

This report presents an improved design and the development of a center-articulated, modular, multi-modal robot. Section 2 describes the mechanical design, highlights certain limitations that were overcome, and presents the improved configuration of the robot. Section 3 presents the kinematic and dynamic modeling of the articulated robot using the Lagrangian approach. The following section discusses simulations that demonstrate the robot’s behavior under various input and damping conditions. Section 5 discusses the fabrication, assembly and testing of the robot in different environments. Sections 6 and 7 conclude the report and highlight potential directions for future work. Appendix, Section 9 includes the MATLAB codes used in the modeling process.

2 Design

2.1 Overview of the Design

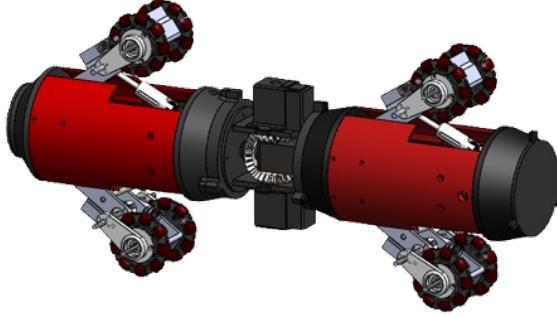


Figure 2.1: CAD Model of the Robot

The design of the robot focuses on creating a multi-modal robotic platform for inspection of complex environments. The system is tailored to navigate both pipelines and ground terrains. It comprises two primary components, the front and the rear modules, which are connected via a 2 Degrees of Freedom (DOF) joint, as shown in the CAD model in the Figure 2.1. This joint, actuated by high-torque servo motors, provides relative movement between the modules in three movement modes, enabling the robot to traverse bends and junctions. It also enables steering on the ground. A sectional view of the robot is shown in the Figure 2.2, highlighting all of its integral components.

The robot also houses a wall press mechanism which allows the robot to support itself within pipes, enhancing traction while adapting to changes in pipe diameter, as shown in the Figure 2.3. This mechanism combines both passive and active components. The design utilizes a pantograph mechanism housing a spring, along with a lead screw mechanism to adjust the robot's diameter. When the wheels of the robot press against the pipe walls, the spring generates the necessary normal force. For small changes in pipe diameter, the spring compensates by maintaining the normal force. However, when the normal force exceeds acceptable limits, the active mechanism engages to adapt to larger

variations in pipe diameter, ensuring stable operation.

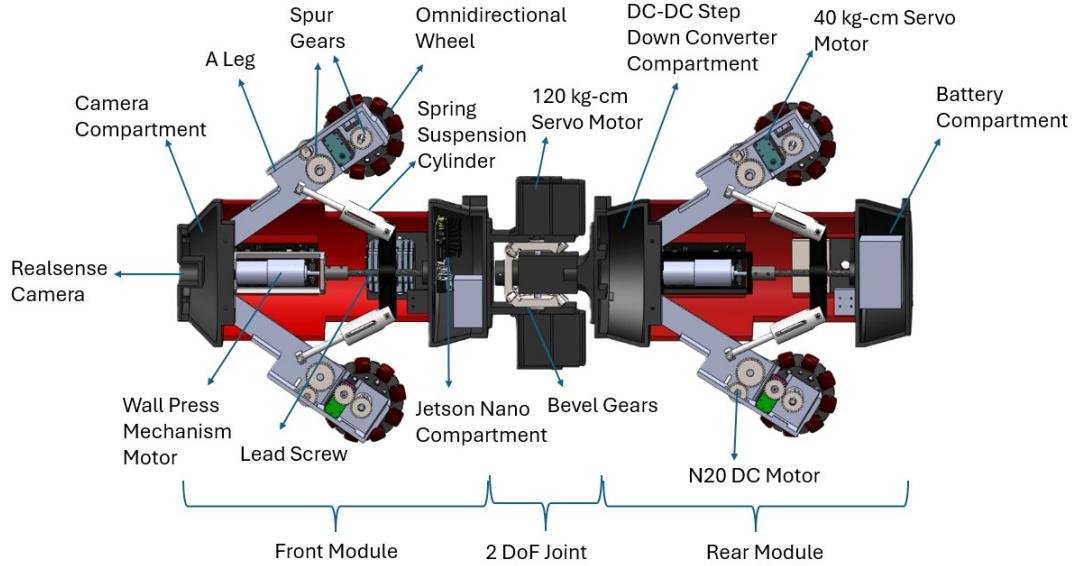


Figure 2.2: Sectional View of the Robot

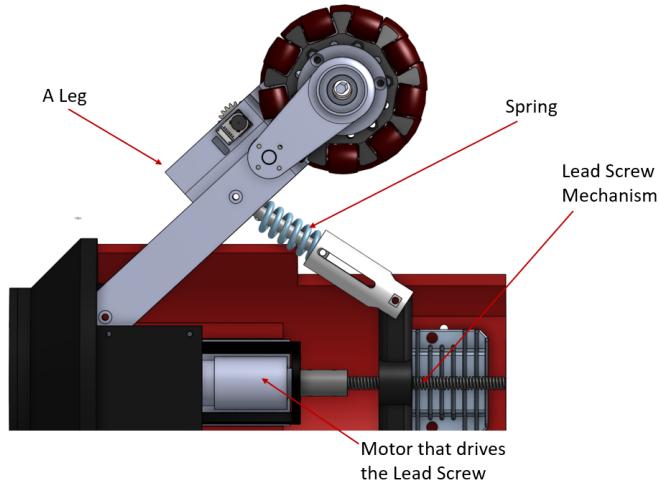


Figure 2.3: Wall Press Mechanism

2.2 Initial Design and Limitations

The robotic system under development comprised four legs, each housing an independent locomotion mechanism. In the initial design, power transmission from the motor to the drive shaft was achieved using worm gear mechanisms. While this configuration offered compactness and perpendicular shaft transmission, it proved to be inefficient in terms

of torque transmission. During experimental evaluation, it was observed that the motors were unable to generate sufficient torque to drive the robot through certain critical configurations, particularly those involving higher load or constrained motion. This limitation significantly affected the robot's overall performance and mobility, necessitating a reevaluation and redesign of the locomotion subsystem.

Another critical objective of the robot was its ability to operate effectively both within pipes and on open ground. This required the robot to adapt its support polygon by increasing the lateral distance between its wheels during ground traversal to enhance stability. Consequently, an additional expansion mechanism had to be designed and integrated within the same leg that housed the primary locomotion system.

This dual-function design introduced significant mechanical challenges. Specifically, the wheels were required not only to rotate about the drive shaft for locomotion but also to translate along the shaft to adjust the inter-wheel distance. Integrating both these degrees of motion within a compact leg module led to design constraints, including limited space and increased part complexity.

The two-degree-of-freedom (2-DoF) joint also presented performance limitations. The joint is essential for navigating through pipe bends, overcoming obstacles inside the pipe, and is a crucial component for navigation on the ground. The earlier chosen 50 kg·cm servo motors were not providing enough torque for the pitch motion of the robot's modules. This required a redesign of some components of the joint and also the selection of better actuators in order to maintain the required torque during actuation.

2.3 Design Modifications and Improvements

2.3.1 Locomotion Mechanism

Based on the limitations observed in the initial configuration, the locomotion mechanism underwent several design iterations to achieve a more efficient and reliable transmission system. Bench testing revealed that the previously implemented worm gear mechanism introduced significant inefficiencies, primarily due to its low mechanical efficiency and limited torque transmission. To address this issue, a spur gear-based transmission mech-

anism was proposed, offering improved efficiency and simpler integration.

Despite the use of a high-efficiency spur gear setup, the existing motor continued to struggle in delivering the required torque at low speeds, especially under load. To better understand the torque demands, a simulation study was conducted using SolidWorks Motion Analysis. The robot was simulated in a vertical pipe traversal scenario as shown in the Figure 2.4, and the results indicated a torque requirement of approximately 2 Nm to maintain adequate traction at sufficient normal force.

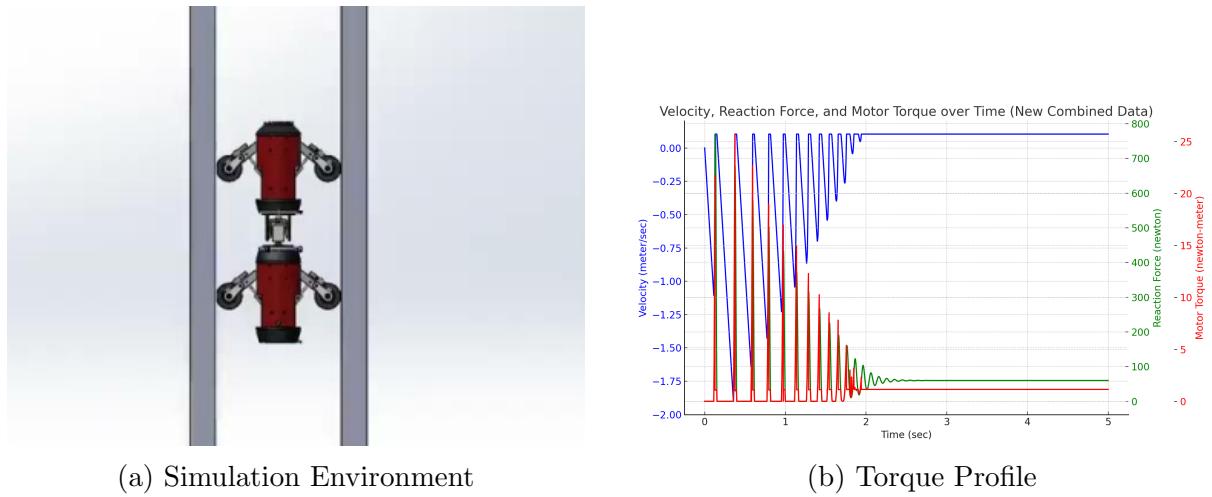


Figure 2.4: Wheel Torque Simulation

Based on these findings, a new set of high-torque motors (Waveshare 40 kg.cm) was selected. The integration of the new motors necessitated a complete redesign of the leg module, constrained by the fact that other interfacing components had already been finalized. A new spur gear mechanism was developed to transmit power from the motor to the drive shaft, along with a motor mount to securely house the upgraded motor and minimize reaction forces transmitted from the shaft.

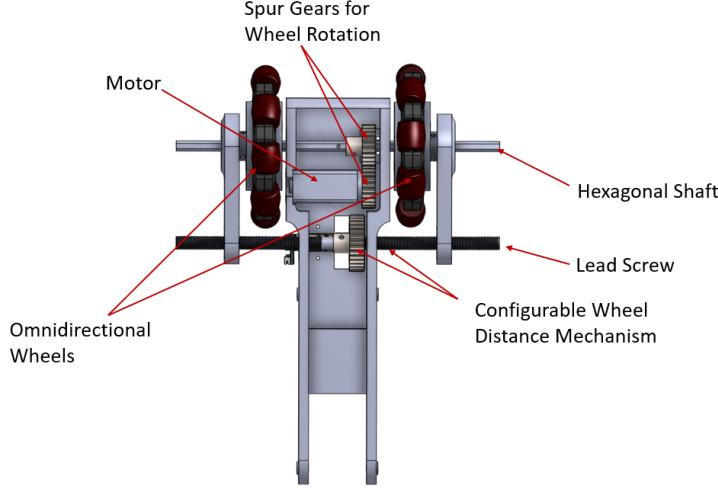


Figure 2.5: Locomotion Mechanism

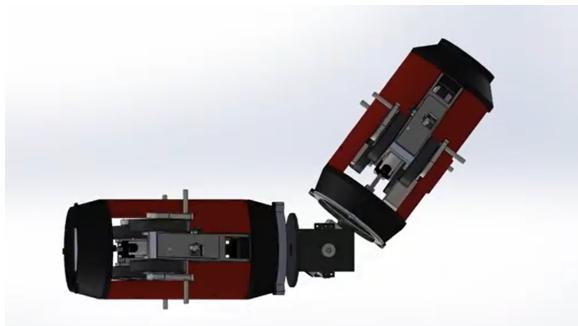
2.3.2 Configurable Wheel Distance Mechanism

Integrating the configurable wheel distance mechanism with the locomotion system within the same leg module posed significant design challenges, as previously discussed in Section 2.2. To accommodate both rotational and translational motion, a hexagonal shaft was employed. The wheel is mounted on this shaft using a linear bearing, which allows for smooth linear movement while simultaneously ensuring rotational coupling with the shaft, thanks to its hexagonal cross-section.

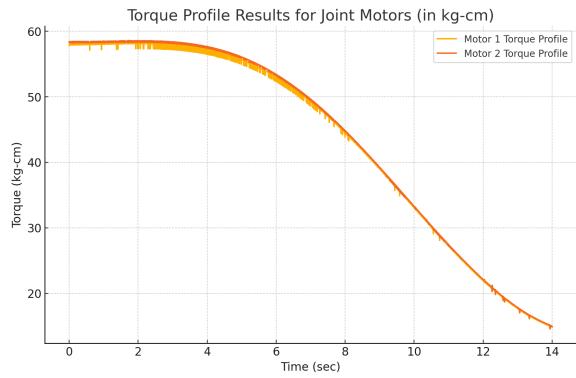
The linear motion of the wheel is driven by a lead screw mechanism housed within the leg. This mechanism enables dynamic adjustment of the distance between the wheels in each module, effectively expanding or contracting the robot's support polygon as required. An expanded support polygon significantly improves the robot's stability during turning maneuvers and while navigating uneven or complex terrains, both on flat ground and within pipes.

2.3.3 2 DoF Joint

In scenarios involving vertical pipe bends, the two-degree-of-freedom (2-DoF) joint experienced significant torque demands. To accurately estimate the required actuation torque, a simulation study was conducted on SolidWorks Motion Analysis Software.



(a) Simulation Environment



(b) Torque Profile

Figure 2.6: Joint Torque Simulation

The results indicated peak torque requirements of approximately 60 kg·cm for each motor as shown in the Figure 2.6. The previously selected servo motors were rated well below this threshold, resulting in inadequate articulation and instability during incline testing. To address this issue, new high-torque actuators were selected, specifically,

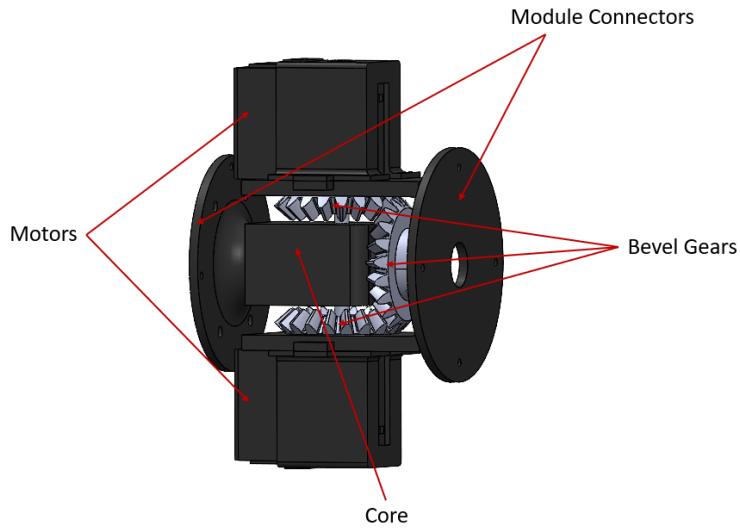


Figure 2.7: 2 DoF Joint Module

the Waveshare RSBL120-24 servo motors. These are high-precision, high-torque devices equipped with 360° magnetic encoders, providing accurate position feedback essential for controlled articulation. The integration of these motors required partial redesign of the joint housing and mounting interfaces to accommodate the new actuators while maintaining alignment with the rest of the robot structure.

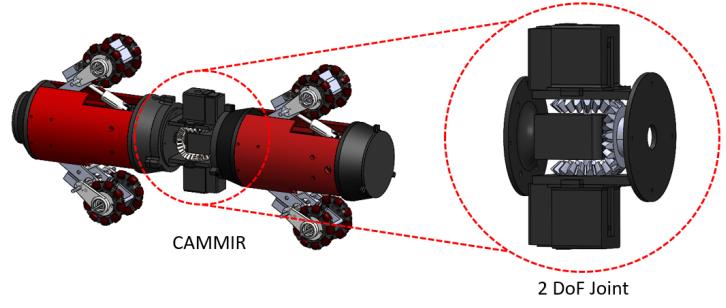


Figure 2.8: Robot with the 2 DoF Joint

3 Robot Modeling

3.1 Kinematics

Figure 3.1 shows the schematic of the robot which consists of two modules, with their corresponding frames \mathbf{F}_B and \mathbf{F}_T , the base frame and the trailer frame. For consistency, the terms front module and base module refer to the same physical component of the robot and are used interchangeably throughout this report. The coordinates of the base and the trailer with respect to the inertial frame \mathbf{F}_I are defined as (x_b, y_b) and (x_t, y_t) . The velocity of the base in the base frame is v_b . The velocity of the base module (front

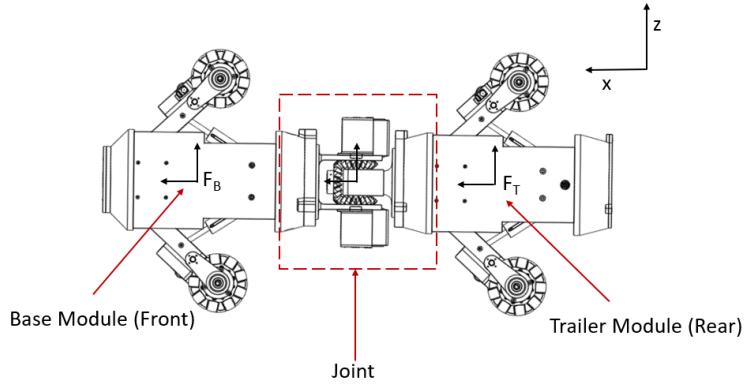


Figure 3.1: Robot Coordinate Frames

module) is given by:

$$\dot{x}_b = v_b \cos \psi_b, \quad \dot{y}_b = v_b \sin \psi_b \quad (1)$$

The orientation of the front module is given by the angle ψ_b , while the articulation angle θ defines the relative orientation between the base and the trailer. The orientation of the trailer module is therefore:

$$\psi_t = \psi_b + \theta \quad (2)$$

where ψ_t represents the heading of the trailer. Since the trailer module is connected to the base module via a joint, its position in the global frame is determined by the position and orientation of the base module. The joint connecting the two modules is located at

$$\mathbf{p}_{jb}^{F_B} = \begin{bmatrix} -L_1 \\ 0 \\ 0 \end{bmatrix} \quad (3)$$

in the base frame, where L_1 is the distance from the center of the front module to the joint. Similarly, the position of the trailer module relative to the joint in its own frame is:

$$\mathbf{p}_{tj}^{F_T} = \begin{bmatrix} -L_2 \\ 0 \\ 0 \end{bmatrix}. \quad (4)$$

Thus, the position of the trailer module in the global frame is:

$$\begin{bmatrix} x_t \\ y_t \\ 0 \end{bmatrix} = \begin{bmatrix} x_b \\ y_b \\ 0 \end{bmatrix} + \mathbf{R}_z(\psi_b) \mathbf{p}_{jb}^{F_B} + \mathbf{R}_z(\psi_b + \theta) \mathbf{p}_{tj}^{F_T}, \quad (5)$$

where \mathbf{R}_z is the rotation matrix about the z -axis.

To find the velocity of the trailer module, we differentiate its position:

$$\dot{\mathbf{p}}_t = \begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ 0 \end{bmatrix} + \mathbf{R}_z(\psi_b)\boldsymbol{\omega}(\psi_b)_{\times}\mathbf{p}_{jb}^{F_B} + \mathbf{R}_z(\psi_b + \theta)(\boldsymbol{\omega}(\psi_b)_{\times} + \boldsymbol{\omega}(\theta)_{\times})\mathbf{p}_{tj}^{F_T} \quad (6)$$

where, $\boldsymbol{\omega}(\psi_b)_{\times}$ and $\boldsymbol{\omega}(\theta)_{\times}$ are the skew symmetric matrices of the angular velocity vectors $\begin{bmatrix} 0 & 0 & \dot{\psi}_b \end{bmatrix}^T$ and $\begin{bmatrix} 0 & 0 & \dot{\theta} \end{bmatrix}^T$ respectively.

The velocity of the trailer module can be expressed as:

$$\mathbf{v}_t = \begin{bmatrix} v_t \\ 0 \end{bmatrix} \quad (7)$$

$$\mathbf{e}_y^t \cdot \dot{\mathbf{p}}_t^{F_T} = 0. \quad (8)$$

$$\mathbf{e}_x^t \cdot \dot{\mathbf{p}}_t^{F_T} = v_t. \quad (9)$$

By transforming and substituting equation (6) in to the constraints (8) and (9), we get the expressions for $\dot{\psi}$ and v_t .

$$\dot{\psi}_b = \frac{-(L_2\dot{\theta} + v_b \sin \theta)}{L_2 + L_1 \cos \theta}. \quad (10)$$

$$v_t = \frac{L_1 v_b + L_2 v_b \cos \theta + L_1 L_2 \dot{\theta} \sin \theta}{L_2 + L_1 \cos \theta}. \quad (11)$$

This means that the velocity of the trailer module is dependent on the velocity of the front module, the steering angle and it's rate, and the geometric parameters L_1 and L_2 . When the steering angle is constant, i.e. when the steering rate $\dot{\theta} = 0$, the velocity of the trailer module can be written as

$$v_t = \frac{v_b(L_1 + L_2 \cos \theta)}{L_2 + L_1 \cos \theta}. \quad (12)$$

When $L_1 = L_2$, both the module's velocities are equal. But in a general case, when

$L_1 \neq L_2$, v_t is dependent on θ .

3.2 Dynamics

The dynamic model of the articulated robot is derived using the Lagrangian formulation [9][10]. The kinetic energy of the base module is given by:

$$K_b = \frac{1}{2} \left(m_b (\dot{x}_b^2 + \dot{y}_b^2) + I_b \dot{\psi}_b^2 \right) \quad (13)$$

where m_b is the mass of the base module, I_b is its moment of inertia, (\dot{x}_b, \dot{y}_b) are its translational velocity components, and $\dot{\psi}_b$ is its angular velocity.

The kinetic energy of the trailing module is:

$$K_t = \frac{1}{2} \left(m_t (\dot{x}_t^2 + \dot{y}_t^2) + I_t (\dot{\psi}_b + \dot{\theta})^2 \right) \quad (14)$$

where m_t is the mass of the trailing module, I_t is its moment of inertia, (\dot{x}_t, \dot{y}_t) are the translational velocity components, and $(\dot{\psi}_b + \dot{\theta})$ represents the trailer's angular velocity.

The component of kinetic energy due to the rotation of wheels is

$$K_w = m_w (\dot{\phi}_b^2 + \dot{\phi}_t^2) \quad (15)$$

The Lagrangian of the system is:

$$L = K_b + K_t + K_w - U \quad (16)$$

The generalized coordinates for the robot are defined as

$$\mathbf{q} = \begin{bmatrix} x_b & y_b & \psi_b & \theta & \phi_b & \phi_t \end{bmatrix}^T \quad (17)$$

where ϕ_b is the angular position of the wheels of the front module and ϕ_t is the angular position of the wheels of the second module. Hence we can write that

$$\dot{\phi}_b = \frac{v_b}{r} \quad (18)$$

$$\dot{\phi}_t = \frac{v_t}{r} \quad (19)$$

and r is the radius of the wheel.

By applying the Euler-Lagrange equations with Lagrange multipliers for constraints, we get the equations of motion in the form of

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = \Gamma + A(q)^T \lambda \quad (20)$$

where $M(q) \in \mathbf{R}^{6 \times 6}$ and $C(q, \dot{q})$ is the matrix of Centripetal and Coriolis force terms. Γ is the matrix of generalized external forces on the system. $A(q)$ is the constraint matrix derived from the kinematics of the robot. From equations (1), (12) and (13), we have

$$A(q)\dot{q} = 0 \quad (21)$$

where $A(q) \in \mathbf{R}^{2 \times 6}$. By rearranging the equation (21), The kinematics of the robot can also be written as

$$\dot{q} = J(q)\dot{\eta} \quad (22)$$

where:

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} r \cos(\psi_b) & 0 \\ r \sin(\psi_b) & 0 \\ -\frac{r \sin(\theta)}{L_2+L_1 \cos(\theta)} & -\frac{L_2}{L_2+L_1 \cos(\theta)} \\ 0 & 1 \\ 1 & 0 \\ \frac{L_1+L_2 \cos(\theta)}{L_2+L_1 \cos(\theta)} & \frac{L_1 L_2 \sin(\theta)}{r(L_2+L_1 \cos(\theta))} \end{bmatrix} \quad (23)$$

$$\dot{\eta} = \begin{bmatrix} \dot{\phi}_b \\ \dot{\theta} \end{bmatrix} \quad (24)$$

Differentiating the equation (22) yields the following expression

$$\ddot{q} = \dot{J}\dot{\eta} + J\ddot{\eta} \quad (25)$$

and pre-multiplying the equation (22) with $A(q)$ gives

$$A(q)\dot{q} = A(q)J(q)\dot{\eta} = 0 \quad (26)$$

Eliminating the Lagrange multipliers using the above equations (20), (25) and (26) , we get the following equation of motion in its reduced form.

$$\overline{M}(\eta)\ddot{\eta} + \overline{C}(\eta, \dot{\eta})\dot{\eta} = \overline{\Gamma} \quad (27)$$

where,

$$\overline{M}(\eta) = J(q)^T M(q) J(q)$$

$$\overline{C}(\eta, \dot{\eta}) = J(q)^T M \dot{J}(q) + J(q)^T C(q, \dot{q}) J(q)$$

$$\overline{\Gamma} = J(q)^T \Gamma$$

3.3 Mechanism Modeling

The wall press mechanism incorporates both active and passive components. The passive mechanism consists of a spring that is compressed to provide the necessary normal force against the pipe walls. The active mechanism includes an adjustable component that allows the spring to be loaded and compressed further against the pipe walls, in case of a change in diameter of the pipe. At static equilibrium, using the forces acting on the robot as illustrated in Figure 3.2, the relationship between the applied wall-press force and the spring compression is derived.

$$F_n = c \cdot k \cdot x,$$

where c is:

$$c = \frac{L_1}{L_1 + L_2} \cdot \frac{\sin(\alpha + \beta)}{\mu \sin(\beta) + \cos(\beta)}.$$

Here, μ is the coefficient of friction, k is the spring stiffness, and x is the spring compression.

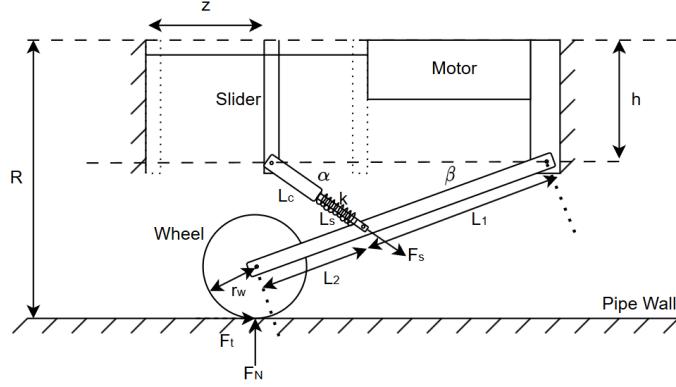


Figure 3.2: Forces in the Wall Press Mechanism

To ensure stable locomotion of the robot inside an inclined pipe, the traction force generated by the motor torque must remain within the limits of available friction. If the required traction exceeds the maximum frictional force, slippage will occur, resulting in loss of control and energy efficiency. To prevent this, the normal force can be increased by actuating the wall-press mechanism by further compressing the springs to enhance the frictional grip. If the wheel torque exceeds the maximum static friction limit, the wheel begins to slip, in the sense that friction is not enough to propel the wheel forward. The necessary traction force to overcome the gravitational component along the pipe's axis increases with the pipe's inclination.

Another important factor to consider in the forces acting on the wheels is rolling resistance. Rolling resistance leads to energy losses during locomotion and must be minimized to improve the robot's efficiency. It can be modeled as a resistive force proportional to the normal force acting on the wheels:

$$F_{rr} = C_{rr}N \quad (28)$$

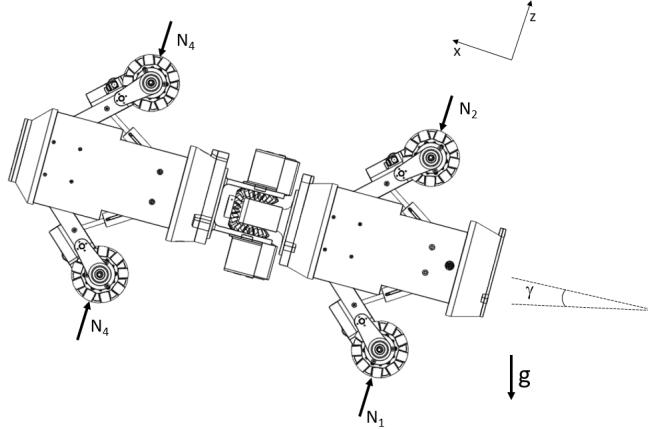


Figure 3.3: Normal Forces acting on the robot

where C_{rr} is the rolling resistance coefficient. Since the robot employs omniwheels with rollers, accounting for rolling resistance becomes especially important, as the additional moving parts and contact surfaces can increase energy losses if not properly considered in the design.

The maximum static friction is related to the normal force as

$$f_{s,max} = \mu N \quad (29)$$

The minimum traction force needed to support the robot in an inclined pipe is given by

$$\sum_{i=1}^4 T_i \geq F_{rr} + mg \sin \gamma \quad (30)$$

The torque limit has to be satisfied in order to avoid slippage.

$$T_i = \frac{\tau_i}{r} \leq \mu N_i \quad (31)$$

Here, γ denotes the inclination angle of the pipe, T_i is the traction force, τ_i is the motor torque, and F_{rr} is the rolling resistance

The force balance perpendicular to the axis of the pipe is given by:

$$\sum_{i=1}^4 (-1)^{i-1} N_i = \sum_{i=1}^4 (-1)^{i-1} F_{n_i} + mg \cos \gamma \quad (32)$$

Here, F_{n_i} represents the force exerted by the wall-press mechanism on the pipe walls.

When the pipe is vertical, this is the only component contributing to the normal force.

4 Simulation and Results

To analyze the dynamic response of the robot under various operating conditions, simulations were conducted on MATLAB using the equations derived in the Section 3. These simulations were aimed at understanding the influence of actuation torques and joint damping on the articulated system’s performance. In each of the simulation cases, the robot’s trajectory is plotted to visualize its overall motion. The wheel angle (ϕ) and joint angle (θ), along with their respective angular velocities, are analyzed to understand the dynamic behavior of the system. Additionally, the total mechanical energy of the system is plotted to verify the physical accuracy and numerical stability of the simulation model. Conservation or consistent dissipation of energy, depending on whether damping is present, serves as an important check to ensure that the dynamics have been implemented correctly and that the results are physically meaningful.

The system dynamics were simulated using a custom MATLAB function based on the derived reduced dynamic equations. The integration was performed using MATLAB’s built-in ode45 solver. The state vector consisted of the joint angle θ , wheel angle ϕ , and their respective angular velocities. The function returned the time derivatives based on numerically evaluated mass and Coriolis matrices. A typical simulation duration ranged from 0 to 20 seconds, with the solver automatically adjusting the step size for accuracy and stability.

4.1 Case 1: Free Response

In the initial case, the robot was simulated without any input torque or damping applied to the articulated joint. This free response simulation helped observe the system’s natural behavior and inherent instabilities due to the absence of control inputs and damping. The initial joint angle was set to 10 deg. The initial wheel angular velocity was set to 2 rad/s. Since no external torque was applied and the system was undamped, the wheel eventually came to rest due to the absence of sustained motion. As there was no energy input into the system, the total mechanical energy remained constant throughout the simulation

duration, as shown in Figure 4.3.

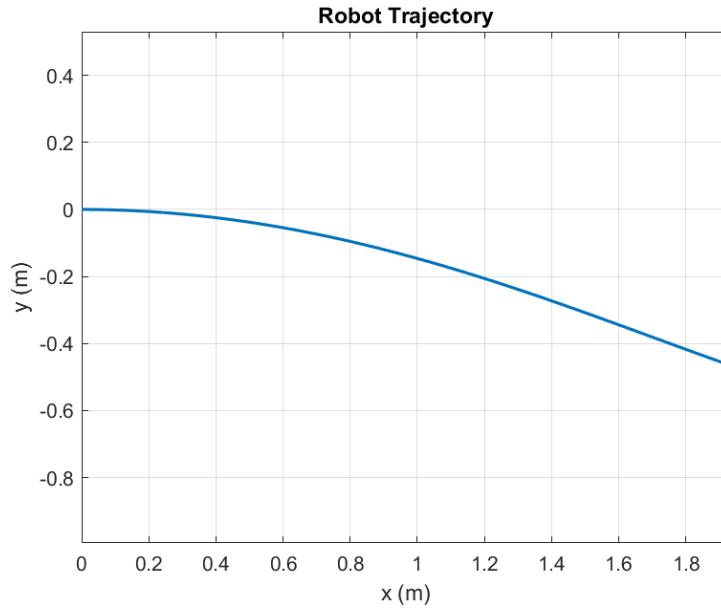


Figure 4.1: Case 1 - Robot trajectory

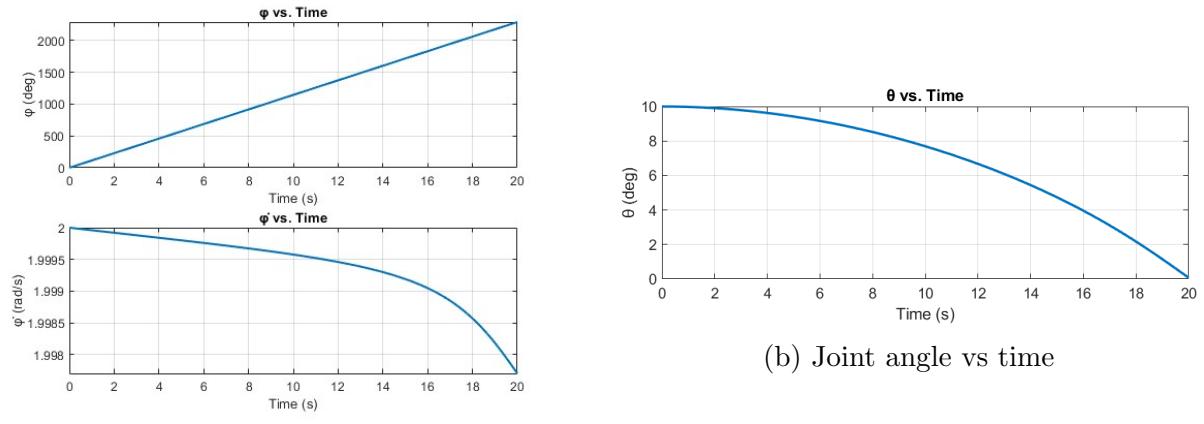


Figure 4.2: Case 1 - Angle variations during simulation.

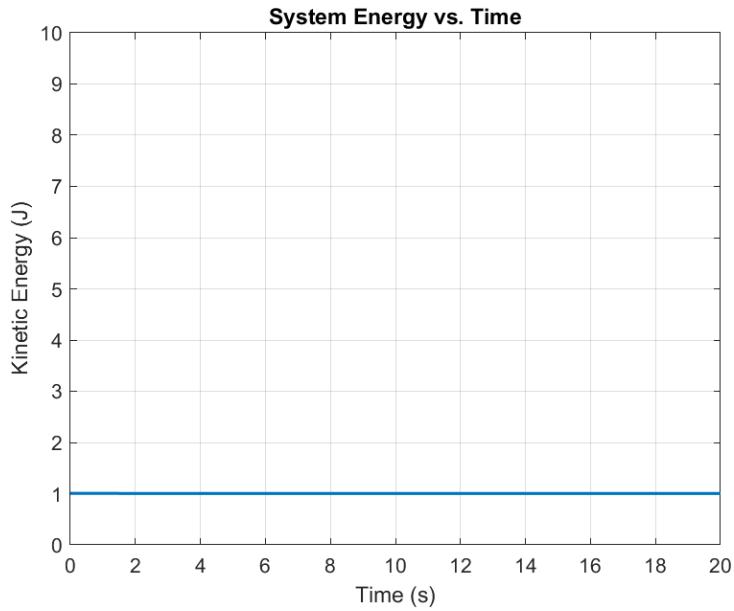


Figure 4.3: Case 1 - Total energy of the system over time.

4.2 Case 2: Simulation with Wheel Torque and No Joint Damping

In the subsequent simulation, a 1 Nm torque was applied to the wheels while keeping the joint undamped. This setup allowed the examination of how the system behaved when only the wheels were actuated. The initial joint angle was set to 10 deg, but the robot is at rest. As shown in Figure 4.5b, the joint undergoes oscillatory motion due to the absence of damping in the system. The resulting motion exhibited instability in the joint.

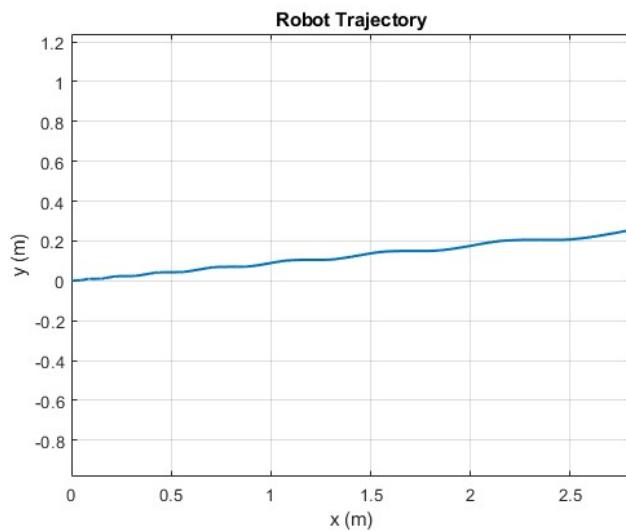
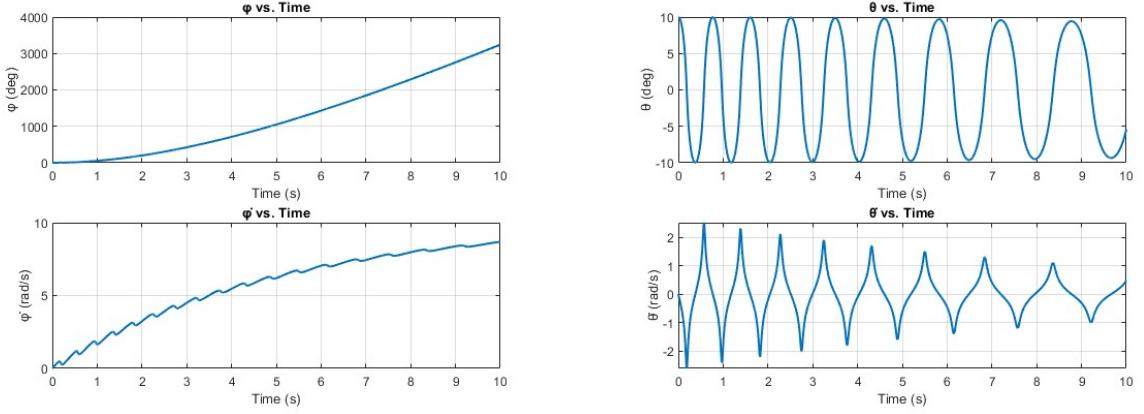


Figure 4.4: Case 2 - Robot trajectory



(a) Wheel angle and Angular Velocity

(b) Joint angle and Angular Velocity

Figure 4.5: Case 2 - Angle Variations during Simulation

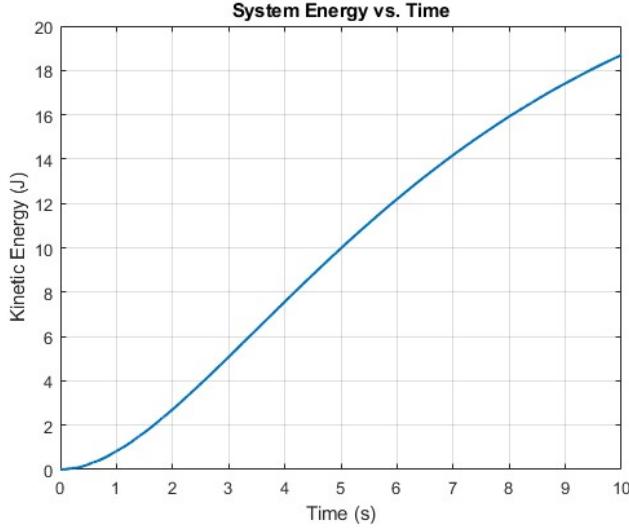


Figure 4.6: Case 2 - Total energy of the system over time.

4.3 Case 3: Simulation with Wheel Torque and Joint Damping

To mitigate the instability observed, damping was introduced at the joint while maintaining wheel torque input. The effect of this modification was a noticeable improvement in stability and smoother articulation as seen in the Figure 5.6b. The joint quickly returns to its neutral position, and the trailing module follows the front module of the robot.

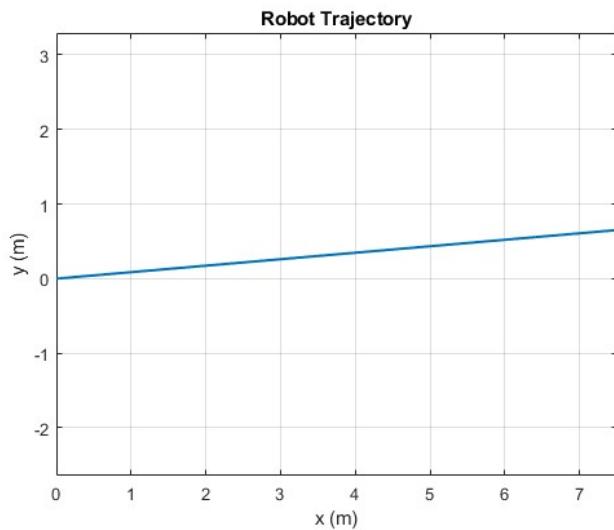
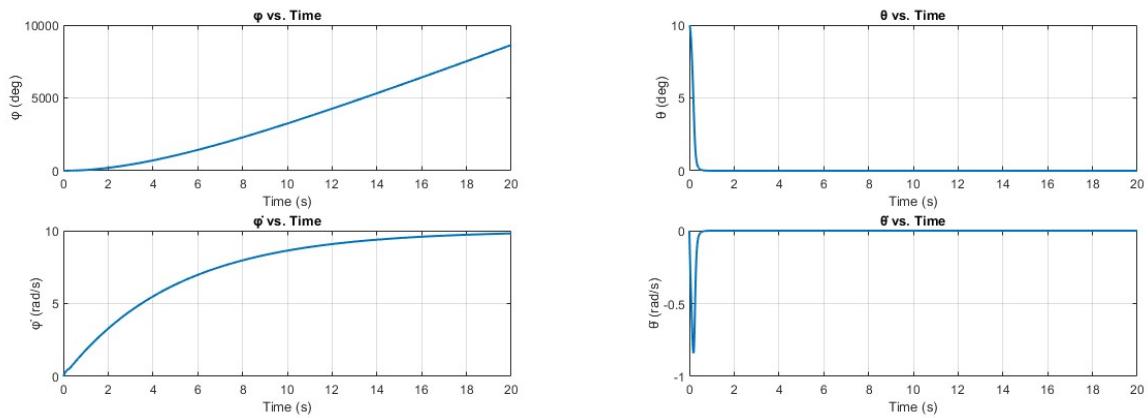


Figure 4.7: Case 3 - Robot trajectory



(a) Wheel angle and Angular Velocity

(b) Joint angle and Angular Velocity

Figure 4.8: Case 3 - Angle Variations during Simulation

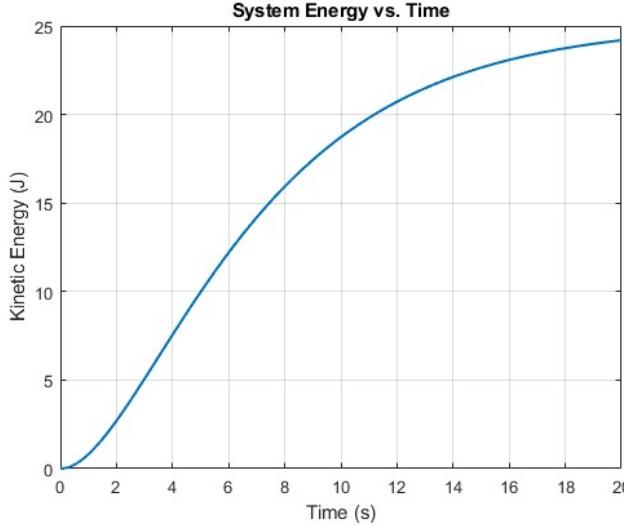


Figure 4.9: Case 3 - Total energy of the system over time.

4.4 Case 4: Simulation with Wheel and Joint Actuation under Damped Conditions

In the final scenario, both the wheels and the joint were actuated, with 1 Nm and 0.1 Nm torques respectively with damping retained at the joint. A change in the joint angle θ is observed in the simulation due to the applied input torque. The rate of change of θ remains stable, attributed to the presence of damping in the system. As energy is introduced through the actuators, a gradual increase in the total energy of the system is evident over time. This configuration demonstrated the coordinated movement of the modules, showing improved control over joint motion and better alignment during locomotion. The robot follows a steep curve due to the gradual increase in the articulation angle, as shown in Figure 4.10.

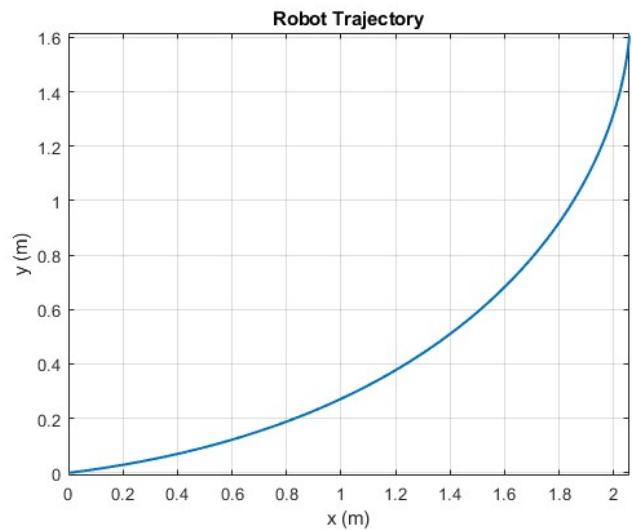
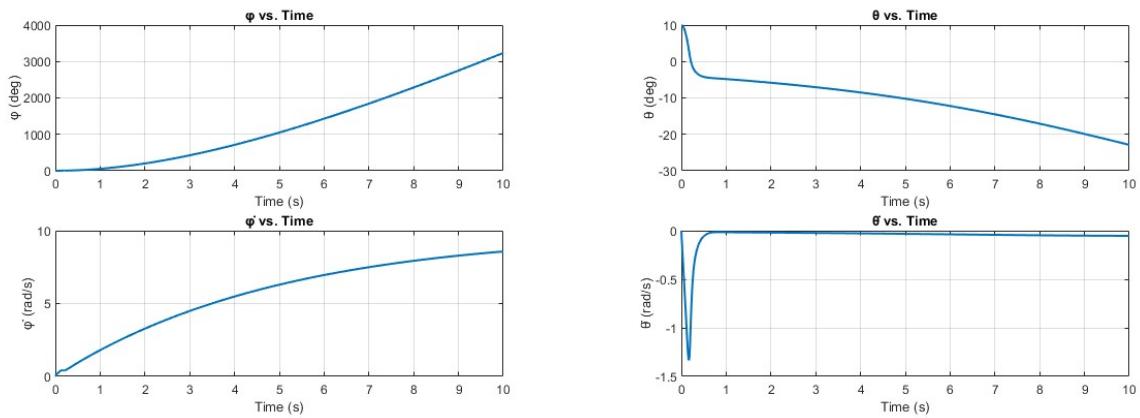


Figure 4.10: Case 4 - Robot trajectory



(a) Wheel angle and Angular Velocity

(b) Joint angle and Angular Velocity

Figure 4.11: Case 4 - Angle Variations during Simulation

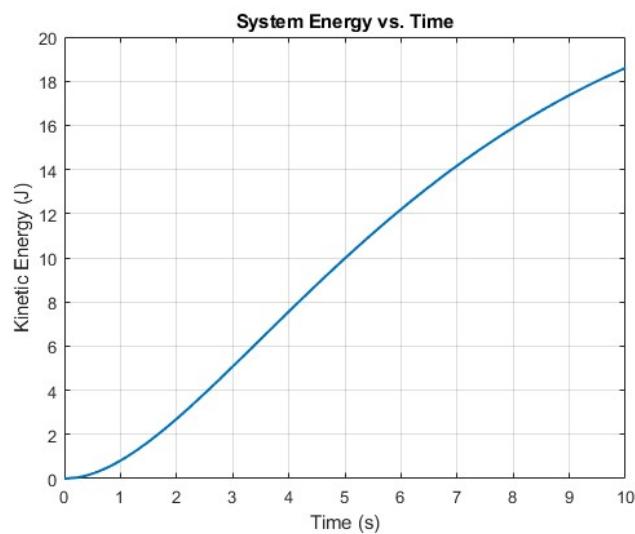


Figure 4.12: Case 4 - Total energy of the system over time.

5 Assembly and Testing

5.1 Fabrication

Upon completion of the design modifications, the fabrication of all major components of the robot was carried out. In this phase, it was ensured that the designed parts were accurately translated into physical components while maintaining dimensional integrity and mechanical functionality. A combination of rapid prototyping and precision machining techniques was employed. Components were fabricated either through in-house processes or outsourced, depending on complexity and material constraints. Engineering drawings were prepared for all components, and the fabrication process was closely monitored. Throughout the process, assembly trials were performed at intermediate stages to verify fit, identify potential interferences, and assess ease of integration with existing modules.

The redesigned leg modules were 3D printed using PLA material, as seen in the Figure 5.1. The motor was installed into the leg module and secured using a 3D-printed mount. The transmission mechanism employed a spur gear system, with the gears manufactured from steel using advanced techniques such as Wire Electrical Discharge Machining (Wire EDM). These gears were mounted onto a procured hexagonal shaft and meshed with a pinion gear connected to the motor.



Figure 5.1: Fabricated Leg

Two omnidirectional wheels were mounted on either side of the hexagonal shaft. The wheel mounts were machined from aluminum using lathe operations. Aluminum was chosen for its high strength-to-weight ratio, ensuring mechanical robustness while minimizing additional mass. Each wheel was interference-fitted onto a linear bearing, which was then placed over the hexagonal shaft. This configuration enabled both rotational and linear motion of the wheels. The configurable wheel distance mechanism was realized using a lead screw system. Rotation of the lead screw was actuated by an N20 motor through a spur gear arrangement.

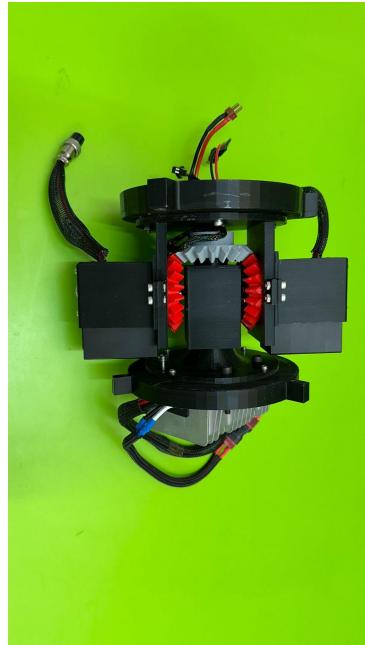


Figure 5.2: Wall Press Mechanism with Lead Screw

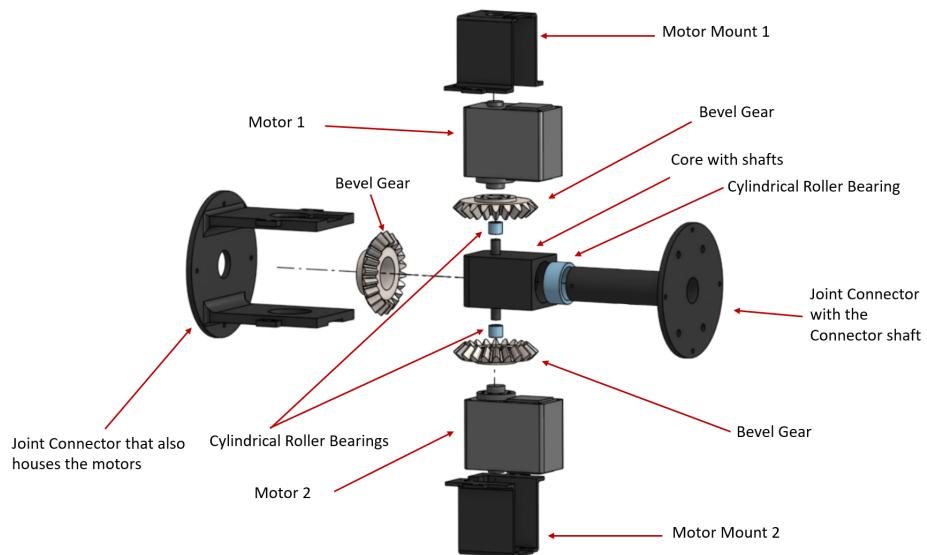
The wall-press mechanism consisted of arms actuated by lead screws (Figure 5.2), which, when rotated, expanded the legs radially to exert force against the inner surface of the pipe. These arms were spring-loaded to enable passive force regulation, allowing the mechanism to adapt to varying pipe diameters and surface conditions. All other components of the wall-press mechanism were fabricated in-house using 3D printing using PLA. Additional components of the robot, such as mounts for electronics and motor housings, were 3D printed using PLA material. As these parts are not subject to significant mechanical loads, PLA was deemed sufficient to meet the design requirements without compromising strength. Washer type load cells were embedded in each leg to measure normal force in real time.

The components of the 2-DoF joint were also fabricated using high-resolution 3D

printing to ensure strength and dimensional accuracy. Given the torque requirements of the joint, a fine printing setting was used to achieve higher infill density and structural integrity. The joint employs two motor-driven bevel gears, symmetrically positioned and meshed with a central bevel gear as shown in the Figure 5.3b. This arrangement enables three distinct motions, yaw, roll, and pitch through sequential operations using only two actuators. A specially designed support core serves as the mounting base, maintaining precise alignment of the bevel gears to ensure consistent and reliable meshing during operation.



(a) Fabricated 2 DoF Joint



(b) Exploded View of the Joint

Figure 5.3: 2 DoF Joint and Assembly

5.2 Electronics

The robotic platform employs a modular and distributed embedded control system, optimized for flexibility, fault tolerance, and expandability. As illustrated in Figure 5.4, the system consists of six peripheral micro controllers coordinated by a central controller, all interconnected via a robust CAN bus.

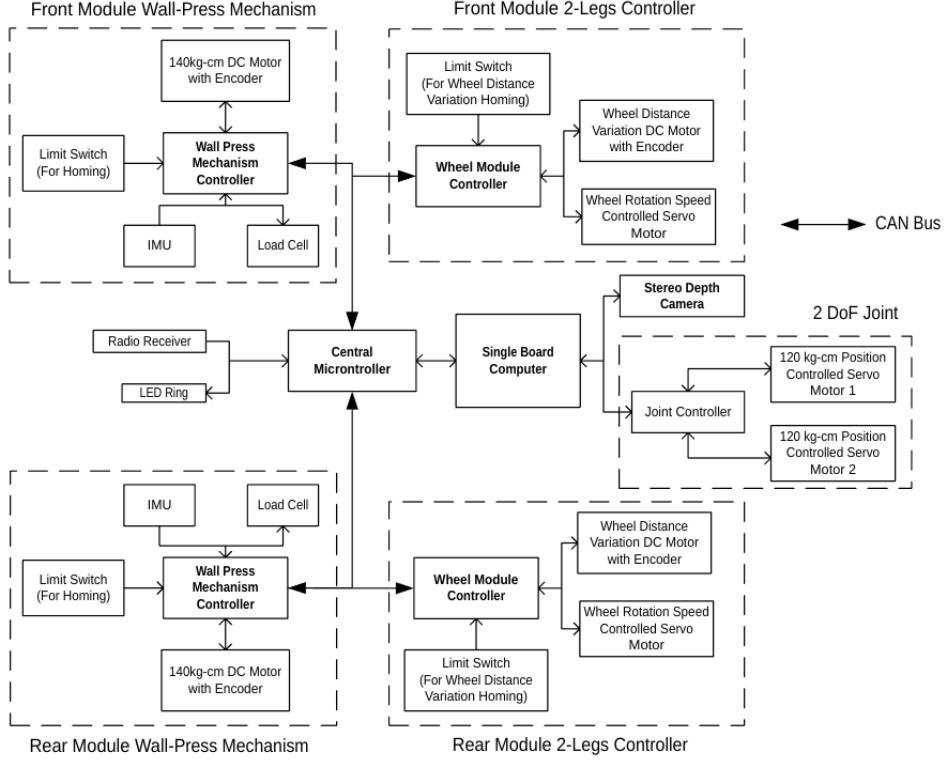


Figure 5.4: System Architecture

Two STM32 Blue Pill microcontrollers (one per module) are dedicated to controlling the wall-press mechanisms. These handle encoder feedback from the 140 kg·cm DC motors, limit switch inputs for homing, load cell data acquisition, and IMU-based orientation sensing. Each controller executes wall expansion or contraction in response to CAN commands.

Each module's wheel subsystem is managed by a dedicated STM32 Nucleo-32 board. These control two actuators, one for configurable wheel distance mechanism using a DC motor with encoder feedback, and another for wheel rotation using a speed-controlled servo motor. Limit switches ensure safe travel limits and reliable homing operations.

A central STM32-based board receives teleoperation inputs from a 2.4 GHz radio receiver via SBUS. It interprets these inputs and dispatches high-level control commands to individual subsystem controllers over the CAN network. It also serves as a bridge to the high-level computer (Jetson Nano) via UART for bi-directional telemetry and supervisory control. The Jetson Nano interfaces with the 2-DoF joint servos and the onboard Intel RealSense D435i stereo depth camera.

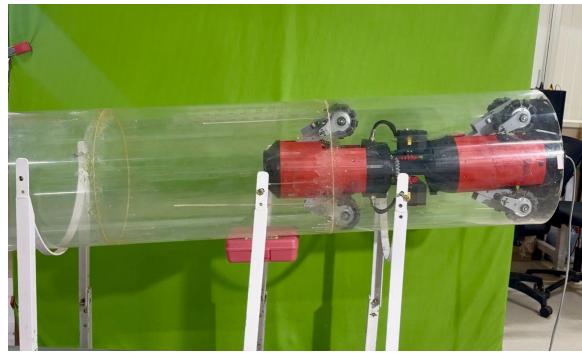
5.3 Prototype Development and Testing

The prototype was developed and tested across multiple environments to validate its design and functionality. A 2.4 GHz 6-channel radio transmitter was integrated for tele-operation. The SBUS signal from the receiver was decoded and mapped to control variables such as wall-press position, wheel rotation speed, joint angles, and wheel distance extension.



Figure 5.5: Fabricated Robot

The robot was tested in a series of controlled environments as shown in the Figure 5.6. Performance in each scenario was documented both visually and via telemetry. The system demonstrated responsiveness, stability, and structural robustness throughout the tests.



(a) Traversing an inclined straight pipe with a 400 mm inner diameter



(b) Navigating a 90-degree bend section



(c) Operating on flat and uneven ground

Figure 5.6: Prototype Testing in Multiple Environments

6 Conclusion

This report presented the design, modeling, simulation, and prototyping of a novel center-articulated, modular, multi-modal robotic platform capable of navigating both structured and unstructured environments. The design addresses limitations of previous iterations by enabling traversal in pipelines as well as on ground, with steering facilitated through a 2-DoF articulated joint system.

A Lagrangian-based modeling approach was used to analyze the system's kinematic and dynamic behavior. Simulation results validated the robot's response under various input and damping conditions, demonstrating smooth and controlled articulation.

The embedded control system, organized around a central controller and multiple peripheral microcontrollers communicating over a CAN bus, enabled distributed actuation and real-time control of the wall-press mechanisms, wheel rotation, and joint movement. A robust teleoperation interface was implemented using an SBUS-controlled radio transmitter.

The developed prototype was tested across multiple scenarios, including inclined straight pipes, 90-degree bends, and uneven terrain. The robot demonstrated reliable, responsive, and adaptable performance under all tested conditions.

Overall, the proposed system offers a multi-modal solution to key challenges in pipe inspection robotics, with the potential for real-world deployment across a wide range of industrial applications.

7 Future Scope

Although a dynamic model of the robot has been developed, the current implementation is limited to manual control. The integrated load sensor within the wall-press mechanism provides the capability to measure the normal force exerted on the pipe walls. In future iterations, this sensor data can be utilized for real-time force feedback control. By incorporating slip detection or estimation, the normal force and driving torque can be actively regulated to minimize slip during locomotion. This would enhance the

robot's traction and stability, especially in variable pipe conditions, and enable a more autonomous control framework.

8 References

- [1] J. Heesik, T. Kim, Y. Lee, Y. Kim, J. Kim, H. Lee, and H. Choi, “A review: Technological trends and development direction of pipeline robot systems,” *Journal of Intelligent Robotic Systems*, vol. 105, 07 2022.
- [2] J. Liu, Y. Tong, and J. Liu, “Review of snake robots in constrained environments,” *Robotics and Autonomous Systems*, vol. 141, p. 103785, 2021.
- [3] J. Tan, Y. Yang, Z. Fan, Y. Wang, Q. Zhou, and Z. Wang, “Overview of pipeline inspection robots,” *Academic Journal of Science and Technology*, vol. 9, p. 208–211, Jan. 2024.
- [4] C. Altafini, “Why to use an articulated vehicle in underground mining operations?,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 4, pp. 3020–3025 vol.4, 1999.
- [5] M. Delrobaei and K. A. McIsaac, “Design and steering control of a center-articulated mobile robot module,” *Journal of Robotics*, vol. 2011, no. 1, p. 621879, 2011.
- [6] S. Ohkawa, Y. Takita, H. Date, and K. Kobayashi, “Development of autonomous mobile robot using articulated steering vehicle and lateral guiding method,” *Journal of Robotics and Mechatronics*, vol. 27, no. 4, pp. 337–345, 2015.
- [7] W. Zhao, L. Zhang, and J. Kim, “Design and analysis of independently adjustable large in-pipe robot for long-distance pipeline,” *Applied Sciences*, vol. 10, no. 10, 2020.
- [8] N. Shi, H. Li, T. Xu, H. Hua, J. Ye, and Z. Chen, “Design, analysis, and optimization testing of a novel modular walking device for pipeline robots,” *Machines*, vol. 12, no. 10, 2024.
- [9] R. Hatab, “Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework,” *Advances in Robotics Automation*, vol. 02, 01 2013.

- [10] Y. Yavin, “Modelling the motion of an underground mining vehicle,” *Mathematical and Computer Modelling*, vol. 42, no. 9, pp. 1123–1130, 2005.

9 Appendix

9.1 Derivation of Dynamics Equations using Lagrangian Approach

```
1 clear; clc;
2
3 %symbolic variables
4
5 syms xb yb psib theta phib phit real
6 syms xb_dot yb_dot psib_dot theta_dot phib_dot phit_dot real
7
8 L1 = 0.28; % Distance from joint to front module wheels
9 L2 = 0.28; % Distance from joint to trailer module wheels
10 r = 0.05; % wheel radius
11
12 mb = 4.5; % mass of base module (front)
13 mt = 4.5; % mass of trailer module
14 mw = 0.12; % mass of wheels
15 Ib = 0.012; % moment of inertia of base module (front)
16 It = 0.012; % moment of inertia of trailer module
17 Iw = 0.01; % moment of inertia of wheels
18
19
20 y = [xb; yb; psib; theta; phib; phit];
21 y_d = [xb_dot; yb_dot; psib_dot; theta_dot; phib_dot; phit_dot];
22
23 % kinetic energy
24
25 T = (mt*((L2*cos(psib + theta)*(psib_dot + theta_dot) - yb_dot +
26 L1*psib_dot*cos(psib))^2 + ...
27 (xb_dot + L2*sin(psib + theta)*(psib_dot + theta_dot) +
28 L1*psib_dot*sin(psib))^2))/2 + ...
29 (It*(psib_dot + theta_dot)^2)/2 + ...
30 (Ib*psib_dot^2)/2 + ...
```

```

29 mw*(phib_dot^2 + phit_dot^2) + ...
30 (mb*(xb_dot^2 + yb_dot^2))/2;
31
32 % M(i,j) = d^2 T / d(qdot_i) d(qdot_j)
33
34 n = length(y);
35 M_sym = sym('M', [n n]);
36
37 for i = 1:n
38     for j = 1:n
39         M_sym(i,j) = diff(diff(T, y_d(i)), y_d(j));
40     end
41 end
42
43 M_sym = simplify(M_sym);
44
45 % Compute the Coriolis/Centrifugal Matrix C
46 % Using the Christoffel symbol approach:
47 % C_{i,j} * qdot_j = sum_{k} [C_{i,j,k} * qdot_j * qdot_k],
48 % where
49 % C_{i,j,k} = 0.5 * ( dM(i,j)/dq(k) + dM(i,k)/dq(j) - dM(j,k)/dq(i) ).
50
51 C_sym = sym('C', [n n]);
52
53 for i = 1:n
54     for j = 1:n
55         c_temp = sym(0);
56         for k = 1:n
57             c_ijk = 0.5*( diff(M_sym(i,j), y(k)) ...
58                         + diff(M_sym(i,k), y(j)) ...
59                         - diff(M_sym(j,k), y(i)) );
60             c_temp = c_temp + c_ijk*y_d(k);
61         end
62         C_sym(i,j) = c_temp;
63     end

```

```

64 end

65

66 C_sym = simplify(C_sym);

67

68 % Inorder to incorporate the kinematic constraints, we incorporate lagrange
69 % multipliers into the lagrange equations A(q)'*lambda = 0

70

71 J = [r*cos(psib), 0;
72      r*sin(psib), 0;
73      -(r*sin(theta))/(L2 + L1*cos(theta)), -L2/(L2 + L1*cos(theta));
74      0, 1;
75      1, 0;
76      (L1 + L2*cos(theta))/(L2 + L1*cos(theta)),
77      (L1*L2*sin(theta))/(r*(L2 + L1*cos(theta)))];

78

79

80 J_dot = [0, 0;
81      0, 0;
82      -(r*theta_dot*(L1 + L2*cos(theta)))/(L2 + L1*cos(theta))^2,
83      -(L1*L2*theta_dot*sin(theta))/(L2 + L1*cos(theta))^2;
84      0, 0;
85      0, 0;
86      (theta_dot*sin(theta)*(L1^2 - L2^2))/(L2 + L1*cos(theta))^2,
87      (L1*L2*theta_dot*(L1 + L2*cos(theta)))/(r*(L2 + L1*cos(theta))^2)];

88 M_reduced = J'*M_sym*J;

89 disp(simplify(M_reduced));

90

91 C_reduced = J'*M_sym*J_dot + J'*C_sym*J;
92 C_reduced = subs(C_reduced,psib_dot,-(L2*theta_dot + phib_dot*r*sin(theta))/(L2 +
93 L1*cos(theta)));
94 disp(simplify(C_reduced));

```

9.2 Dynamics Simulation Codes

9.2.1 Simulation MATLAB File: Simulate

```
1 % simulate.m
2 clear; clc; close all
3
4 %parameters
5 tspan = [0 20];
6
7 %initial angles
8 phib0 = deg2rad(0);
9 theta0 = deg2rad(10);
10
11 q_ind0 = [phib0; theta0];
12
13 %initial angular velocities
14 phib_dot0 = 1;
15 theta_dot0 = 0;
16 qdot_ind0 = [phib_dot0; theta_dot0];
17
18 %State vector: x = [phib; theta; phib_dot; theta_dot]
19 x0 = [q_ind0; qdot_ind0];
20
21 options = odeset('RelTol',1e-6, 'AbsTol',1e-8);
22
23 [T, X] = ode45(@dynamics, tspan, x0, options);
24
25 %trajectory
26 phib_dot = X(:,3);
27 theta = X(:,2);
28
29 v = 0.05 * phib_dot;
30
31 xPos = cumtrapz(T, v .* cos(theta));
32 yPos = cumtrapz(T, v .* sin(theta));
```

```

33
34 margin = 0.1;
35 xmin = min(xPos) - margin;
36 xmax = max(xPos) + margin;
37 ymin = min(yPos) - margin;
38 ymax = max(yPos) + margin;
39
40 r = 0.05;
41 L1 = 0.28;
42 L2 = 0.28;
43
44 phib_dot = X(:,3);
45 theta = X(:,2);
46 theta_dot = X(:,4);
47 psidot = -( L2*theta_dot + phib_dot.*r.*sin(theta) ) ...
48     ./ ( L2 + L1*cos(theta) );
49
50 psi = cumtrapz(T, psidot);
51
52 v = r * phib_dot;
53
54 xPos = cumtrapz(T, v .* cos(psi));
55 yPos = cumtrapz(T, v .* sin(psi));
56
57 %plot
58 figure;
59 plot(xPos, yPos, 'LineWidth', 1.5);
60 axis equal; grid on;
61 xlabel('x (m)'); ylabel('y (m)');
62 title('Robot Trajectory');
63
64 figure;
65 %phi
66 subplot(2,1,1);
67 plot(T, rad2deg(X(:,1)), 'LineWidth', 1.5);

```

```

68 xlabel('Time (s)');
69 ylabel('phi (deg)');
70 title('phi vs. Time');
71 grid on;
72 %phi velocity
73 subplot(2,1,2);
74 plot(T, X(:,3), 'LineWidth',1.5);
75 xlabel('Time (s)');
76 ylabel('phi_dot (rad/s)');
77 title('phi_dot vs. Time');
78 grid on;
79
80 figure;
81 %theta
82 subplot(2,1,1);
83 plot(T, rad2deg(X(:,2)), 'LineWidth',1.5);
84 xlabel('Time (s)');
85 ylabel('theta (deg)');
86 title('theta vs. Time');
87 grid on;
88 %theta velocity
89 subplot(2,1,2);
90 plot(T, X(:,4), 'LineWidth',1.5);
91 xlabel('Time (s)');
92 ylabel('theta_dot (rad/s)');
93 title('theta_dot vs. Time');
94 grid on;
95
96 %energy
97 E = zeros(length(T),1);
98 for i = 1:length(T)
99     theta_i      = X(i,2);
100    phib_dot_i = X(i,3);
101    theta_dot_i = X(i,4);
102
```

```

103 M11 = (4917*cos(theta_i) + 4932) / (9800*(cos(theta_i)+1));
104 M12 = (4932*sin(theta_i) + 4917*cos(theta_i)*sin(theta_i)) ...
105     / (3500*(cos(theta_i)+1)^2);
106 M22 = -(4917*cos(theta_i)^2 - 4932) / (625*(cos(theta_i)+1)^2);
107 M_i = [ M11, M12;
108         M12, M22 ];
109
110 qdot = [phib_dot_i; theta_dot_i];
111 E(i) = 0.5 * qdot' * M_i * qdot;
112 end
113
114 % plot
115 figure
116 plot(T,E,'LineWidth',1.5);
117 grid on
118 xlabel('Time (s)')
119 ylabel('Kinetic Energy (J)')
120 title('System Energy vs. Time')
121 xlim([0 10])
122 xticks(0:1:10)

```

9.2.2 Simulation MATLAB File: Dynamics

```

1 function xdot = dynamics(t, x)
2
3 syms psib theta phib real
4 syms psib_dot theta_dot phib_dot real
5
6 % x = [phib; theta; phib_dot; theta_dot]
7
8 q = x(1:2);
9 qdot = x(3:4);
10 fullState = {q(1), q(2), qdot(1), qdot(2)};
11
12 for k = 1:length(fullState)

```

```

13     fullState{k} = double(fullState{k});
14
15
16 M_reduced = [Matrix derived from Dynamics equations];
17
18 C_reduced = [Matrix derived from Dynamics equations];
19
20 allVars = [phib, theta, phib_dot, theta_dot];
21 M_func = matlabFunction(M_reduced, 'Vars', allVars);
22 C_func = matlabFunction(C_reduced, 'Vars', allVars);
23
24 M_val = M_func(fullState{:});
25 C_val = C_func(fullState{:});
26
27 %commanded torques
28 tau_phib_cmd = 0;
29 tau_theta_cmd = 0;
30
31 %damping coefficients
32 b_phi = 0;    % 0.05 wheel damping
33 b_theta = 0;   % 0.2 joint damping
34
35 phi_dot = x(3);
36 theta_dot = x(4);
37
38 % Total torques with damping
39 tau_phib = tau_phib_cmd - b_phi * phi_dot;
40 tau_theta = tau_theta_cmd - b_theta * theta_dot;
41
42 F_val = [tau_phib; tau_theta];
43
44 qdd = M_val \ (F_val - C_val * qdot);
45
46 xdot = [qdot; qdd];
47 end

```