UNIVERSITY OF OTTAWA

OTTAWA-CARLETON INSTITUTE FOR MECHANICAL AND AEROSPACE ENGINEERING

Modeling and Simulation of the Locomotion Mechanics of a Class of Legged Autonomous Robots

Author:
Bhargav Konidala

Supervisor:
Davide Spinello

A thesis submitted in partial fulfillment of the requirements for the Master of Applied Science degree

in

Mechanical Engineering

uOttawa

Modeling and Simulation of the Locomotion Mechanics of a Class of Legged Autonomous Robots

Bhargav Konidala

Abstract

Autonomous robots are employed in several important tasks, for example, from health care to military and defense applications involving operations in hazardous and inaccessible environments. Legged autonomous robots can be advantageous due to high adaptability and stability over any terrain, superior obstacle avoidance capability, and advantages through redundancy by utilizing multiple legs. Compared to rigid-legged robots, flexible-legged robots are highly compliant, suitable for non-destructive inspection applications, and possess enhanced gait control with improved energy efficiency. An approach to designing flexible-legged robots is to mimic desirable features evolved via natural selection in biological organisms. Conceptualizing new biologically inspired flexible-legged robots can expand the usability and improve the efficiency of robots in different applications.

In this project, the inspiration for locomotion design is the mobility principle utilized by small-scale organisms in the form of beating protrusions referred to as cilia or flagella. Notably, the collective beating dynamics of ciliary arrays reveal essential characteristics such as synchronization, phase locking, and metachronal coordination suitable for terrestrial and aquatic robot locomotion.

This thesis presents the formulation, simulation, and analysis of a planar bio-inspired flexible-legged robot for terrestrial locomotion. Each leg of the robot is modeled as a bundle of flexible filaments using constrained Euler elastica that is suitable to describe some of the characteristics of cilia or flagella. The legs/protrusions are mechanically coupled through the base, representing the robot's payload, via linear springs or elastic lumped elements, to produce certain desired collective beating patterns upon individual moment actuations. The locomotion mechanism is illustrated in simulation, wherein the results pave the ground for future work with refined modeling to account for hardware implementation constraints.

Acknowledgements

To begin with, I would like to extend my utmost appreciation and gratitude to my supervisor Dr. Davide Spinello for his unwavering guidance and immense patience in assisting me throughout my academic journey. Under his mentorship, I acquired numerous invaluable skills and further developed existing ones within an incredibly supportive environment. Completing this thesis would not have been possible without his valuable advice and continuous support.

Additionally, my heartfelt thanks to all my family members, mentor, and friends for their constant support. Their resolute belief in me, encouragement to strive for excellence, and patient wait for my success with this project have meant the world to me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Robot locomotion generally refers to the set of features of a robotic device that enable navigation through the deployed workspace or environment. Movement of a robot in unstructured or unknown environments is highly advantageous and opens the possibility of many applications, including non-destructive environmental inspection [1–3], commercial implementations [4, 5], military and defense applications [6–8], and operation in hazardous and generally not accessible environments [9–11].

Based on a robot's environment, locomotion strategies can be classified as terrestrial, aquatic, and aerial. Further, a locomotion strategy that enables the robot to traverse two or more environments is defined as multi-modal locomotion [12]. Apart from the classifications listed above, some literature also classifies robots designed for climbing into a separate category known as scansorial [13]. No matter the environment, the fundamental problem statement for designing a robot locomotion mechanism remains the same. The locomotion mechanism must enable the robotic device to move anywhere within the defined environment and let the robot perform its tasks, if any, without interruptions. The problem statement can be further broadened to let the locomotion mechanism improve robot performance factors such as versatility, robustness, compliance, energy consumption, environmental disturbance rejection, and unknown environment exploration.

This thesis focuses on modeling a multi-legged locomotion strategy for terrestrial locomotion of robotic devices. The proposed mechanism utilizes flexible filaments as legs to allow movement of the robotic device over a flat and even terrain. This work is the first step towards interactions with different types of terrain. The goal is to gain some insight into the possibility of leveraging the dynamics of coupled flexible filaments for terrestrial locomotion, extending a feature typically used for aquatic locomotion.

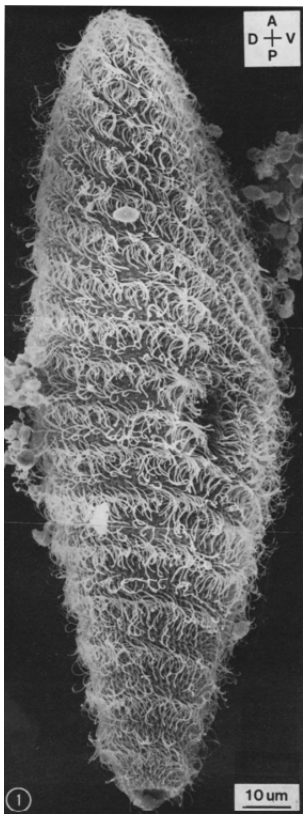
1.1 Motivation and Inspiration

Over the past few years, much research has been conducted on multi-legged robots for terrestrial locomotion. Walker-type robots designed using flexible elements improve dynamic performance by providing compliance and offering significant advantages compared to their rigid-body counterparts. The benefits include better gait control, energy optimization, shock absorption, improved tractability over rough and uneven terrains, and better speed with simplified control of locomotion dynamics in some situations [14].

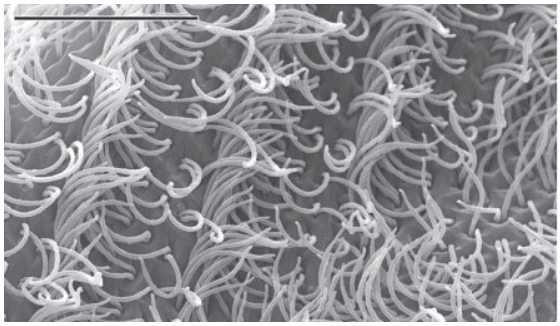
While many available robot locomotion design solutions offer the benefits listed above, the degree to which each benefit offers an advantage to the robot varies with each design. Quantitative comparisons between significant performance characteristics of different locomotion mechanisms designed for the same purpose can vary significantly. The variations in the characteristics can be attributed to a range of factors, such as using a simplified dynamics model due to assumptions, underperformance due to real-world mechanical constraints on the system dynamics, and unavailability or limitations of technology to realize locomotion mechanism concepts to their fullest potential.

For the reasons stated above, it is essential to conceptualize new locomotion mechanics for robotic device motion. Designing new locomotion mechanisms offers more options to users that can be tailored to solve problems with many detailed requirements. Further, newly created locomotion technologies can be refined in the future to be put to use in more complex situations.

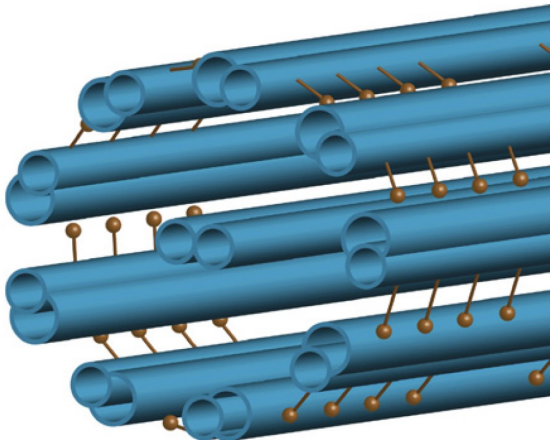
For this thesis, the inspiration for the locomotion mechanism comes from the mobility principle used by small-scale biological entities to travel in fluid environments. These microorganisms possess hair-like protrusions on the surface of their bodies, termed cilia or flagella, based on the biological purpose of these protrusions. The fundamental structure of these protrusions is called an axoneme, a cylindrical arrangement of pairs of elastic rod-like filaments connected by links. The elastic nature of these filaments gives the axonemal structure flexibility. Also, the axoneme consists of dynein motor proteins that allow it to oscillate. While lacking a centralized control over the protrusions, the microorganisms can move through aquatic environments by actuating individual surface protrusions that synchronize among themselves and generate beating patterns to push the surrounding fluid efficiently [15, 16]. The generated beating patterns include synchronization, phase locking, and metachronal coordination, among others [17]. Metachronal



(a) Microscopic view of the Paramecium [18]



(b) Magnified view of the cilia [19]



(c) Three-dimensional axoneme model [20]

Figure 1.1: Scanning electron micro graph of a freeze-dried Paramecium fixed during forward swimming, magnified view of the metachronal waves of the cilia, and three-dimensional model of an axonemal structure

coordination among an axonemal array consists of synchronous oscillation of all the axonemes with a constant phase shift between contiguous axonemes, resulting in a traveling wave propagating through the array. Figure 1.1 shows a microscopical appearance of a Paramecium during swimming, where the cilia on the surface are beating metachronously. Figure 1.1 also shows a three-dimensional representation of an axonemal structure.

Recent literature [21] has shown that the emergent synchronous behavior in surface protrusions is aided by a combination of two phenomena known as hydrodynamic and basal coupling. Hydrodynamic coupling results from the interaction forces between the generated flow field of an oscillating protrusion and surrounding beating protrusions. Basal coupling is due to the presence of a direct mechanical linkage between the bases of the protrusions through flexible intracellular connections. Synchronization between two beating protrusions aided through basal coupling essentially represents a microscopic version of Huygens synchronization,

wherein the pendula of two neighboring clocks suspended from a beam are observed to gradually synchronize due to weak mechanical coupling between the bases of clocks [22]. Overall, microorganisms can minimally actuate individual protrusions on the surface of their bodies and generate beating patterns with the help of hydrodynamic and basal coupling. Hence, it should also be possible to achieve beating patterns, such as metachronal coordination, in axonemal arrays through pure basal mechanical coupling when there are no hydrodynamic interaction forces.

Coincidentally, metachronal coordination is essential for the terrestrial locomotion of many arthropods such as crabs and millipedes [23]. Also, the gait utilized by many multi-legged robots involves metachronal coordination [24]. Therefore, terrestrial locomotion can be achieved by using an array of basal coupled axonemes as the legs of a robot that coordinate metachronously. Here, basal coupling represents the linkage between the bases of the legs through a flexible mechanical component. As observed in arrays of biological axonemes, basal mechanical coupling can transfer forces between one oscillating leg to another, ultimately leading to synchrony. This terrestrial locomotion mechanism could help overcome some challenges that multi-legged robots face. Since the axonemal structure is flexible, they are a robust solution for the legs of a robot, as they can adapt and morph to different terrains. Also, instead of forcing each leg to coordinate metachronously, the actuation inputs to each leg can be reduced, and basal coupling can be exploited to achieve metachronal coordination. Once the dynamics of this system are predicted, important optimizations for terrestrial robotic locomotion can be considered, such as maximizing autonomy and minimizing energy consumption.

1.2 Thesis Objectives and Contributions

The main goal of this work is to introduce and test the viability of a multi-legged robot locomotion strategy inspired by the aquatic mobility features of microorganisms. The research work presented in this thesis contributes towards the overall objective in the following ways:

- **Mathematical modeling**: present the locomotion mechanism concept, which utilizes the flexible axonemal structure as the legs of the robot, metachronal coordination between the legs to move the robot, and basal mechanical coupling in the form of elastic lumped elements that link the bases of the legs to possibly aid in metachronal coordination. Since this is the first iteration of the

proposed concept, the locomotion mechanism is implemented and modeled for a planar robot with a rigid payload.

- **Simulation**: test the viability of the terrestrial locomotion mechanism by simulating the motion of the proposed planar robot over a flat and even terrain. The contact mechanics between the robot and the terrain are modeled suitably.
- **Analysis**: study the simulation results and illustrate the desired features of the locomotion mechanism while examining possible pitfalls.

Ultimately, the analysis presented in this thesis serves as a step toward leveraging the desirable features of these fluid-based microorganisms for terrestrial robot locomotion and future work on the realistic implementation of the proposed locomotion mechanism.

1.3 Thesis Outline

The thesis is outlined as follows:

- **Chapter 2**: literature review on the fundamental concepts of terrestrial robots and their types, multi-legged robot locomotion mechanics and gaits, and contact mechanics of leg-terrain interactions. Also, concepts related to structural configuration and biomechanics of cilia or flagella and models related to synchronous beating patterns in ciliary arrays in fluid environments are explored.
- **Chapter 3**: mathematical modeling of the proposed terrestrial robot consisting of an array of axonemes acting as legs of the robot that are coupled at the base via linear springs. The springs also connect the axonemes with the rigid body of the robot. Each axoneme is modeled as a nonlinear rod with internal forces by using constrained Euler elastica, resulting in nonlinear partial differential equations. The equations are solved through the finite element method by considering their weak formulation. The robot body dynamics are derived through Euler-Lagrange equations. Also, the interaction forces between the terrain and robot are modeled suitably. Supplementary material related to the mathematical modeling is presented in Appendix A.

- **Chapter 4**: discussion of the results obtained through the simulation of the locomotion of the proposed robot model over a flat terrain. The analysis of the accuracy of the mathematical modeling of the robot is presented. The code used for the simulation of the robot is shown in Appendix B.
- **Chapter 5**: brief summary of the work presented in this thesis, conclusions determined from the results, viability and drawbacks of the proposed robot locomotion solution, and future possible improvements to the robot.

Chapter 2

Literature Review and Background Theory

2.1 Terrestrial Robots

Terrestrial robots utilize contact forces generated by the interaction between the actuating mechanism and the terrain to create thrust and move on the ground. The number of robots that fall under this category is vast and varies significantly from each other in terms of design complexity, control strategy, actuating mechanisms, performance capabilities, and application space. Hence, off-the-shelf robots typically do not exist for most terrestrial applications. Instead, based on the problem statement, optimal robots are designed and developed using a combination of various available technologies. Ongoing research involves developing algorithms to find an optimal robot design and assembly from a given set of primary building elements. The primary building elements may include actuators, linking mechanical objects, and different types of joints. For instance, [25] demonstrates an algorithm to develop a set of equally optimal robot design solutions from a few primary robot elements to achieve multiple objectives.

2.1.1 Types of Terrestrial Robots

Though a strict classification template does not exist, based on the propulsion mechanism utilized, terrestrial robots may be primarily classified as follows [26–28]

2.1.1.1 Wheeled/Tracked Robots

Most terrestrial robots utilize wheels to move on the ground due to simplicity in design and control, cost efficiency, and inherent stability because of consistent contact with the surface during motion over even terrains [29]. Though highly advantageous for solving most common problems, wheeled robots do not function effectively over uneven and rocky terrains due to a decreased availability of contact points with the ground for the wheels. Also, low friction makes wheeled robots operate poorly over smooth and slippery surfaces. However, some of the problems caused explicitly by the terrain can be alleviated by increasing the number of wheels and using a better suspension system to maintain contact between wheels and the ground during travel. However, adding more wheels also adds more weight to the robot due to actuators, brakes, and steering mechanisms, which might lead to limitations in robot functions [30].

Tracked robots are utilized to overcome the contact problem with wheeled robots over uneven and smooth terrains to gain better traction and locomotion stability. However, the motion control of tracked robots is complex. The most common way to move tracked robots is through skid-steering. Skid-steering requires actuating tracks on each side of the robot at different speeds to generate lateral slippage and initiate a turn. During skid-steering, the slippage and the nonlinear track-terrain interaction dynamics are challenging to calculate and predict, due to which trajectory tracking control is required to maintain a tracked vehicle in the desired trajectory [31]. Further, skid-steering may also cause loss of contact points between tracks and terrain, leading to inaccurate measurements by the odometry sensors. Also, since each wheel of the tracks on the robot is independently driven, straight-line travel over uneven, complex, and smooth terrains can be difficult as each part of the track might have frictional differences with other parts of the track, requiring different individual wheel velocities to propel the vehicle ahead. Due to different wheel velocities, the overall velocity of a track on one side of the robot may be higher than the other, causing the robot to deviate from the straight-line path.

2.1.1.2 Sliding Robots

Another approach to overcome the contact and mobility problems with wheeled and tracked robots over rough, uneven terrains and smooth, low-friction viscous terrains such as dunes is to use robots that slide on the terrain. Sliding robots utilize various gait patterns on the contact area between the body and the terrain to push against

the terrain, generating reaction forces that propel the robot towards the desired direction [32, 33]. Sliding robots primarily initiate the movement by undulating laterally (snake-like gait) [34, 35] or pushing longitudinally (worm-like gait) [36]. Some robots utilize a combination of the two gaits to improve the overall terradynamic capabilities of the robot and gain an advantage over terrains where one type of locomotor function is a better performer than the other. For instance, a worm-like robot that utilizes a longitudinal peristaltic wave to propel itself toward the desired direction can benefit from lateral undulation by possessing more anchoring points for better propulsion and modification of the robot's orientation in the lateral direction to avoid obstacles much more quickly [37]. A typical sliding robot design links many compartmentalized units through joints that allow the structure to sway upon actuation. Also, due to the many actuatable individual units, the robotic system inherently possesses more degrees of freedom than necessary to move on the ground, making it a highly redundant system and, hence, more reliable for navigating challenging terrains. Though complex controllers allow sliding robots to navigate over terrains complicated for wheeled/tracked robots with reasonable tracking control, the overall locomotion efficiency of sliding robots is low over smooth and viscous terrains. Furthermore, the large number of degrees of freedom poses a complex problem in controlling and reducing the cost of actuating them. Also, the small size of sliding robots translates to lower payload capacity and speeds than other types of robots [38]. Some robots utilize articulated passive wheeled modules that replicate a snake-like gait instead of sliding on the terrain for better locomotion speed. However, passive wheeled snake-like robot designs assume a no-slip condition on the terrain when, in reality, it is difficult to control the joint torques such that the wheels do not slip [39]. Due to the body shape, typical applications of snake-like and worm-like robots include surveillance, inspection, and sampling in highly constrained and humanly inaccessible environments [40].

2.1.1.3 Legged Robots

The disadvantages of sliding robots in terms of lower payload capacity, higher energy consumption in actuation, and lower travel speed can be offset using multi-legged robots. Since this thesis sets forth the design of a biologically-inspired flexible multi-legged robotic system, the concepts involved in the types, design, locomotion mechanics, and control of legged robots are discussed in detail in Section 2.2.

2.1.1.4 Hybrid/Other Robots

Terrestrial robots that utilize either a combination of the actuation mechanisms used by the three primarily classified robots or travel through other mediums along with terrestrial locomotion are classified as hybrid/other robots. For example, the survey presented in [41] exhibits many available hybrid mobile robots that change the travel medium or locomotion mechanism to optimize operational efficiency.

2.2 Multi-Legged Robots: An Overview

Unlike sliding robots, where the contact area moves with respect to the surroundings, or wheeled robots, where the contact points maintain continuous contact with the terrain, legged robots always maintain discrete contact points with the terrain such that the contact points do not move with respect to the surroundings during the motion of the robot. For sliding and wheeled robots, a continuous path is required for traversal, possibly finite over rugged terrains with various obstacles. In contrast, the typical path trajectory of a legged robot consists of a series of discrete footholds along the terrain, and robot motion is induced by moving the leg about the foothold. Due to the discrete contact points feature, legged robots can choose from many options the best area on the terrain for foot placement and are well-suited to navigate complex terrains compared to other terrestrial robots. Though wheeled robots have higher payload capacities, move extremely fast and efficiently over suitable surfaces, and can move over small obstacles by employing mechanisms used by all-terrain vehicles [42, 43], users prefer legged robots due to superior mobility in natural terrain. Apart from stepping over obstacles or ditches, legged robots can also climb over obstacles, walk on debris, and ascend stairs, ladders, and uneven terrains [44, 45].

Legged robots can utilize telescopic legs to adjust their leg lengths to change their body orientation or to modify the position of their body's center of gravity during travel, making the robot less prone to tip and over and more reliable. Further, telescopic legs allow the robot to adjust its step length and place its extended feet in slightly deep ditches to secure a strong foothold and to move ahead on the terrain [46]. Even without telescopic legs, legged robots can vary their leg configuration to accommodate surface irregularities and establish contacts at desired points along the terrain. When traveling on soft surfaces, legged robots have a better advantage over wheeled/tracked robots because discrete footholds deform the terrain less, and

lower energy is required to get the legs out of depressions. Furthermore, suitable control strategies can be applied to initiate a soft landing of the legs on the terrain by utilizing terrain parameters obtained either through previous knowledge of the terrain [47], through sensors attached to the legs [48], or through sensing both physical properties and terrain profile via compliant leg elements [49].

Any legged robot comprises two primary units: the body and the legs. Typically, the body of a legged robot houses the power supply, control system, most of the actuators, instruments necessary for the robot to execute tasks, and additional payload if required to be carried as part of a task. The design of the robot's body is crucial for improving the locomotion efficiency of the robot. Adjusting the position of the center of gravity of the robot body through altering dimensions and changing weight distributions allows for better stability of the robot. Also, reducing the overall weight of the robot improves the locomotion speed. Furthermore, suppose the robot's body is slender with an adaptive body shape, and the legs of the robot are compliant. In that case, the compliancy can implement shape tracking and material parameter deduction of the terrain [49]. By recognizing the shape of the terrain, the adaptive body of the robot can be morphed into the shape of the terrain, allowing for better contact points for the legs of the robot over unstructured terrains. Also, the legs of a legged robot act as an active suspension to the robot's body, unlike wheeled/tracked robots requiring a dedicated suspension system to manage the vibrations induced from the terrain onto the robot. Due to the active suspension feature, the path of the trunk of the robot is decoupled from the paths of the feet, making it travel smoothly and freely over uneven terrains. The robot can use decoupling to increase its speed and efficiency on rough terrains. Moreover, legged robots have control over the force distribution through the foothold points, which can be optimized to improve robot-ground interaction.

2.2.1 Advantages and Disadvantages of Multiple Legs

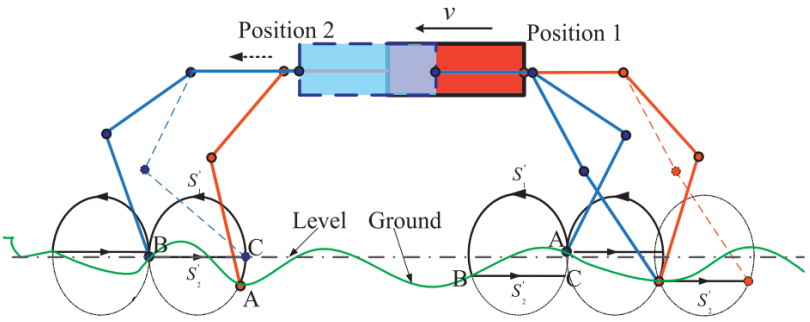
An obvious advantage posed by multiple legs is improved static stability over unstructured terrains, a higher number of isolated footholds that optimize support and traction of the robot during motion, and increased precision in the degree of body orientation adjustments. Adding more legs also increases the degrees of freedom, giving the legged robot redundant legs, greater mobility, and flexibility. Furthermore, adding more than the required number of legs can improve robustness. For instance, in the case of failure of one or more legs, the robot might still be

able to perform its tasks at reduced efficiency by utilizing the remaining legs through a suitable control strategy. Additionally, with more legs, the robot is less susceptible to deadlock situations wherein the robot cannot move in the desired direction. Typically, legged robots avoid deadlock situations by utilizing adaptive gait planning algorithms [50].

Legged robots possess a few disadvantages that can be addressed by incorporating optimized robot designs and complex control systems. The locomotion kinematics and dynamics of legged robots are complex, and achieving synchronized motion with coordination between legs over any terrain or obstacles requires complicated control strategies. Though redundancy is an effective method to move the robot over uneven terrains, improve robustness, and move the robot in case of unpredictable situations, adding more than the required number of legs adds a significant amount of weight. Depending on the design of each leg, the number of actuators required to drive the legs can vary. For instance, if a leg is designed as three links connected via revolute joints, three actuators would be required to drive each joint. Besides the weight of the legs, trunk, devices housed inside the trunk, and other payloads that the robot carries, the actuators also contribute significantly to the robot's weight. If the robot is required to move faster, the power output from the actuators must increase, which means bigger, heavier, and more power-consuming actuators. Hence, when compared to wheeled and tracked robots, legged robots have higher energy consumption and lower payload-to-weight ratio. Also, compared to wheeled and tracked robots, the speed of legged robots is considerably lower on plain and regular surfaces. However, legged robots outmatch wheeled and tracked robots over irregular terrains in speed and efficiency.

2.2.2 Gait Patterns of Multi-Legged Robots

For a legged robot to move in the desired direction, the legs must coordinate and move or rotate in a pre-defined sequence following the body's movement and the shape of the terrain. The pre-defined sequence that the legs follow is termed gait. For any given gait, to generate motion toward the desired direction, each leg of the robot must accomplish two phases of a gait cycle. When the leg is in stance position, the leg is in contact with the ground and pushes against the ground to generate forces that propel the robot in the desired direction. During flight, the leg recirculates through the air and is not in contact with the ground. A leg in flight always returns to the stance configuration, completing the gait cycle. The periods at



(a) Biped robot gait cycle over an unstructured terrain [51]. The robot walks from right (position 1 depicted in the color orange) to left (position 2 depicted in the color blue). The tips of the legs move through curves S_1' and S_2'

(b) Gait cycle of a crab-inspired multi-legged robot over a flat terrain [52]. The robot walks from right to left in five stages from (i) through (v)

Figure 2.1: Illustration of gait cycles of a biped robot and a crab-inspired multi-legged robot

which each leg changes from one configuration to another and the time to complete a gait cycle are variable and significant factors in differentiating gaits. For instance, Figure 2.1 illustrates the gait cycle of a biped robot and crab-inspired multi-legged robot. Legged robots often use a combination of different gait patterns to locomote over various surfaces. There are many possible factors to use to classify legged robot gaits, such as change of orientation of the robot's body during travel, the direction of travel, type of walking, and periodicity between the movement of legs. Figure 2.2 shows various types of gaits utilized by legged robots over different terrains while indicating their periodic properties.

Generally, gaits can be classified according to gait function or motion periodicity [53]. Based on gait function, gaits are classified as level-walking or obstacle-crossing gaits. Gaits that do not require the robot to change the height or orientation of the body are called level-walking gaits. Whereas gaits that require the robot to change the body height and orientation, typically over highly irregular terrains, are termed obstacle-crossing gaits. Based on the motion periodicity, gaits are classified as periodic or aperiodic. While most applications involving traversal over flat and fair terrains require the robot to follow periodic gaits to operate efficiently and execute desired tasks, rough terrains require the robot to follow aperiodic gaits due

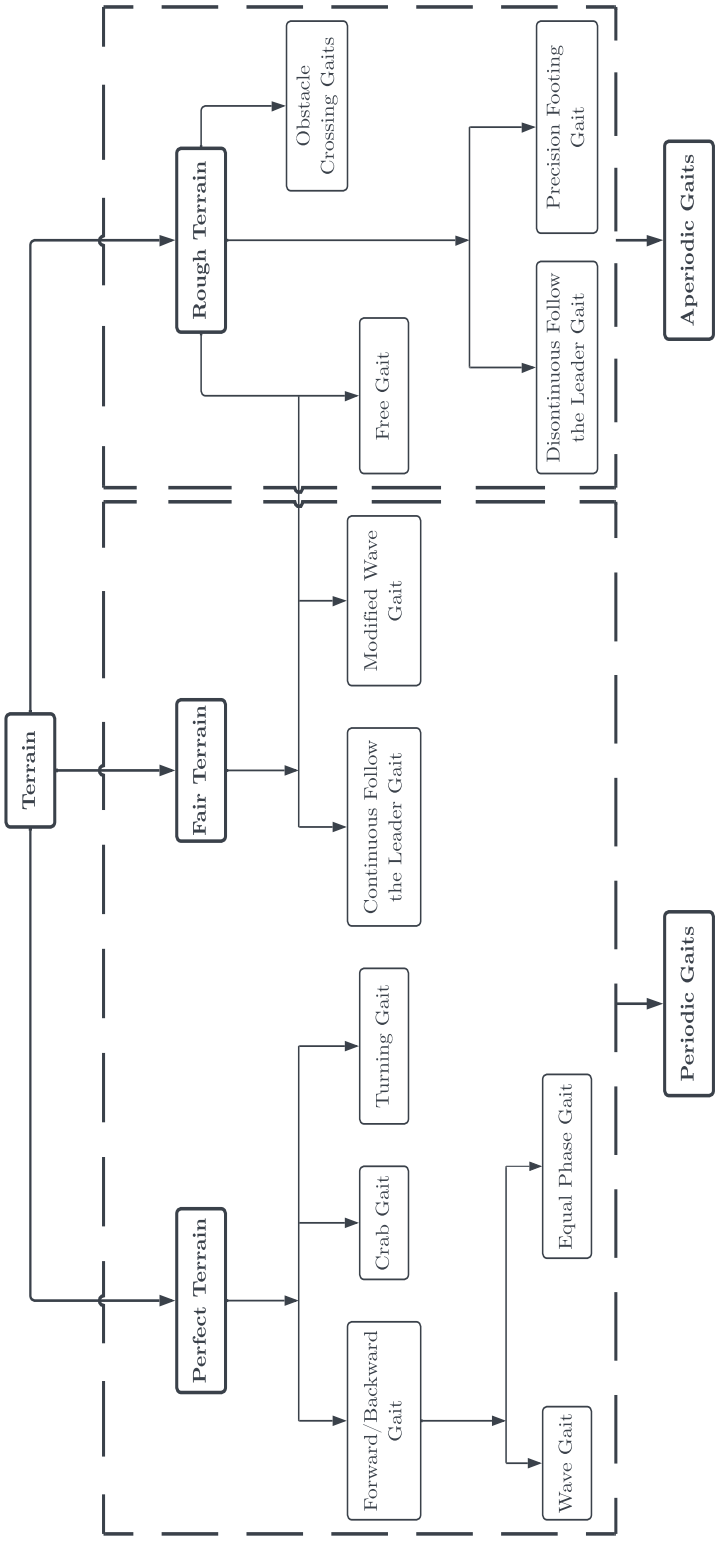


Figure 2.2: Possible gait patterns of a legged robot over various types of terrains

to the unevenness of the terrain.

Many types of level-walking gaits are used for different kinds of terrains [54]. Over a perfectly smooth terrain, level-walking gaits are of three types: forward/backward, crab, and turning gaits. All three types of gaits are related since forward/backward gait is a particular case of crab gait, and crab gait is a particular case of turning gait. Forward/backward gait enables a legged robot to move forward or backward, while the angle between the global vertical axis and the robot body's longitudinal axis does not change. Typically, multi-legged robots use either wave or equal phase gait to move. Through a wave gait, a multi-legged robot periodically moves its legs, resembling a wave propagating through the stepping actions irrespective of the status of its leg's placement over the terrain. Whereas, through equal phase gait, a multi-legged robot equally distributes the lifting and placing events of the legs over a locomotion cycle. Further, literature [55] also shows that wave gait is the most stable of the regular gaits where the stability margin is optimum under certain conditions. In both wave gait and equal phase gait, the feet of the robot can only be adjusted to a small degree to avoid small forbidden areas, hence making them unsuitable for use on terrains with many irregularities. Crab gait, as the word "crab" implies, lets the robot move in a crab-like fashion where the direction of travel is diagonal and at an angle from the global vertical axis. Typically, the angle between the robot body's longitudinal and global vertical axis remains unchanged during crab gait. Further, turning gaits enable the robot to turn in the desired direction. Typically, turning is achieved by differential motion of the legs on either side of the robot, wherein the legs on one side move faster than the other.

Legged robots can still utilize periodic gaits over a reasonably smooth terrain with irregularities to ensure smooth operation. However, the terrain geometry must be known before the robot travels. The terrain parameters can be calculated manually, through a terrain scanner, tactile sensors, or sensing via compliant legs. Once the terrain irregularities are known, the legged robot can traverse a path free of terrain irregularities by knowing the locations of all the secure footholds. Through follow the leader gait, robot operators select secure foothold positions for the front two legs of the robot along the path of travel. The remaining legs are placed at the previous positions of the legs ahead of them. Through this gait mode, the foothold identification and selection process is minimized while the robot maintains a high speed of motion through periodicity. Further, a wave gait is a specific case of continuous follow the leader gait. Hence, to traverse a path with many irregularities, a wave gait can be modified [56] to adjust the foothold positions

of the remaining legs based on the secure foothold locations given for the first two legs to ensure smooth travel of the robot. Another class of gait algorithms, free gaits, do not follow the periodic flow of leg movements or placing/lifting events while ensuring positive stability at all times. Free gaits also require prior knowledge of terrain geometry.

Most level-walking gaits cannot be used for rough terrains with many irregularities. However, a discontinuous follow the leader gait can be used to negotiate travel on such terrains. Discontinuous follow the leader gait ensures good stability at all times, and no more than one leg is lifted at a time. The principle of travel remains the same as that of the continuous one. The footholds for the first two legs of the robot are selected on the path. Another approach to allow a robot to travel safely while maintaining level is to manually declare the foothold position trajectory for all the legs of the robot and guide the legs through them. However, this approach, termed precision footing gait, is cumbersome since the possible options are many, and the process gets complicated as the path length increases. Hence, using obstacle-crossing gaits [57] is the best option for travel on rough terrains while optimizing stability and speed. Obstacle-crossing gaits can also cross over obstacles more prominent than the robot's size.

2.2.3 Metachronal Wave Gait

Metachronal wave gait is a type of wave gait that can be realized in legged robots by actuating each leg of the robot sequentially with a constant desired phase shift between contiguous legs [58, 59]. Metachronal rhythm as a means of locomotion for organisms in both fluid and terrestrial environments can be observed over a vast spectrum of species ranging from the swimming of plankton to the walking of millipedes. There are also terrestrial robots that make use of metachronal coordination for locomotion. For instance, research in [60] presents a soft robotic system that utilizes metachronal waves generated over a magnetically actuated artificial ciliary carpet to perform a crawling motion similar to a millipede. Forward locomotion of a legged robot through metachronal wave gait can be achieved through two methods based on the initial conditions of the legs of the robot. If all the legs of the robot are initially touching the terrain, the robot lifts its rear-most leg and only starts lifting the leg in front of the actuated leg when a desired phase difference between them is achieved. This lifting process that achieves the desired phase shift between the contiguous legs of the robot begins from the posterior and proceeds towards the

anterior as each leg of the robot, in turn, is lifted. Once the forward-most leg of the robot has been lifted, all contiguous legs have a constant phase shift between them, resulting in a metachronal rhythm. The constant phase shift between contiguous legs also results in a traveling sinusoidal wave propagating through the legs of the robot. Since all the legs of the robot are touching the terrain initially, the robot achieves the highest possible initial static stability because all the legs of the robot support the weight of the robot's body. However, the initial forward movement of the robot would be slow because the metachronal coordination between all the legs of the robot would only begin when the forward-most leg of the robot is lifted off the ground. Until the forward-most leg is lifted off the terrain, at least one leg of the robot would be in continuous contact with the terrain, slowing down the forward locomotion process [61].

Another way to achieve forward locomotion through metachronal wave gait is to assume all the contiguous legs of the robot to be initially out of phase by the desired constant phase difference and actuate all the legs simultaneously instead of actuating the legs sequentially. This would result in a metachronal wave from the instant all the legs are actuated since all the contiguous legs are already at a fixed phase lag from one another. This gait in insects is referred to as a diagonal tripod gait which is a form of metachronal wave gait [62–64]. From a control perspective, this method is much easier to implement since it involves simultaneously actuating each leg of the robot, as opposed to the other method, which involves sequentially actuating each leg of the robot. The only drawback of this method is that it initially has a lowered static stability due to a decreased number of legs in contact with the terrain. However, in both methods, the actuation inputs to each leg must be controlled and ensure that contiguous legs are moving with a constant phase difference.

There are many ways to implement the lifting cycle of the legs of a robot utilizing metachronal wave gait. One way is to rotate all the legs such that the tips of the legs of the robot trace a circular trajectory, with each leg tracing a different point on the circle. By enabling the tip of each leg to trace a different point on the circle, a phase difference between the legs can be established, and a metachronal wave propagating through the legs of the robot can be observed. Each leg movement comprises two kinematic stages during the forward movement: dense phase and sparse phase. During the dense phase, the leg is in contact with the ground and rotates backward to generate a thrust that propels the robot forward. The thrust is generated due to the friction between the legs and the terrain. During the sparse

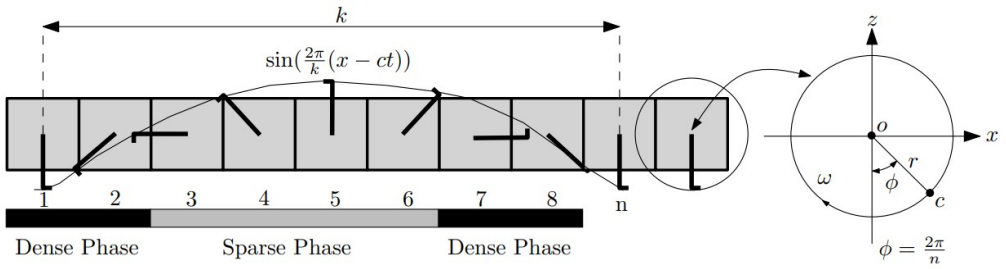


Figure 2.3: Metachronal wave propagation in Millipede locomotion (replicated from [65])

phase, the leg is not in contact with the ground and rotates forward to recover to a position where it would be in contact with the terrain for the dense phase. During the backward motion of the robot, the legs would rotate in the opposite direction for each of the phases. To move left/right, the legs on one side of the robot act as a pivot while the legs on the other side rotate. Further, robots with a body that undulate laterally aid in sharper and quicker turns.

Metachronal coordination among the legs of a multi-legged robot provides increased stability and decreased energy expenditure. Further, the phase difference between each leg can be reduced to increase the robot's speed, but the robot's overall stability is reduced. The traveling wave phenomenon is also observed in myriapods such as millipedes and centipedes that utilize numerous legs to locomote. Since myriapods possess hundreds of legs, they can have enough legs in the dense space to support their body weight at any given instant. The remaining legs are in a sparse phase with a constant phase difference between contiguous legs. The typical locomotion pattern followed by an n-legged millipede is shown in Figure 2.3 [65].

The tips of the legs of a millipede can be approximated to lie on a propagating sinusoidal wave. The position of a millipede's leg can be predicted by assuming that the leg trajectory traces a circular path if suspended freely in the air [66]. By that assumption, when a millipede walks, the ground disrupts the circular path, leading to forward propulsion. Though realistically, millipedes do not rotate their legs along a circular path due to joint constraints. Instead, they move their legs to and fro periodically. However, the circular reference model is a reasonable simplification and allows for easy calculation of the leg's position at any given time when specific parameters, such as the angular velocity ω of the leg, are known. Further, the profile of the millipede legs' extremities can be described by a metachronal wave propagating along the axis x that is aligned with the axis of the body when it is

rectilinear using [65]

$$\sin\left(\frac{2\pi}{k}(x-ct)\right) \tag{2.1}$$

where k and c are the wavelength and the frequency of the propagating wave, respectively. Additionally, the phase difference between contiguous legs is expressed as [65]

$$\phi = \frac{2\pi}{n} \tag{2.2}$$

where n is the number of legs comprised in one wave.

In this thesis, the proposed multi-legged robot will use metachronal wave gait to traverse over a flat and even terrain. The robot's movement is enabled by applying a moment at the base of each leg to rotate the leg. Further, the contiguous legs of the robot are initially assumed to be out of phase by a desired phase difference. Hence, actuating all the legs of the robot simultaneously while ensuring constant rotational frequency would result in metachronal coordination between the legs of the robot, enabling the locomotion of the robot.

2.2.4 Multi-legged Robots with Flexible Elements

Robot modeling is generally based on theoretical solid mechanics, with different degrees of complexity and sophistication of the models based on the device to be described. For example, cases for which lumped parameter modeling can be sufficient are treated as particle masses; on the other hand, articulated systems of rigid bodies are used to model robotic manipulators with fixed and mobile bases, and flexible continua models are involved for soft mechanisms with a large number of degrees of freedom, which can be appropriately described as distributed parameter systems. Once the framework is determined, a suitable theory from classical mechanics, such as Hamilton's principle, can be applied to obtain a linear/nonlinear system described by partial differential equations if continuous elements are involved. Exact solutions for the partial differential equations cannot be often determined. Instead, numerical methods must be used in such cases. One such numerical method is the finite element method, which involves discretizing the solution domain and eliminating the spatial dependence of the partial differential equations to obtain a system of linear/nonlinear ordinary differential equations with only time-dependent parameters. Hence, modeling a robot assuming that each

component has a degree of flexibility is demanding and requires high computational effort. In addition, stability analysis and designing reliable control techniques might be required for the flexible robot to execute desired tasks.

Though the process is complex, there are added benefits when utilizing flexible elements for multi-legged robots. Flexible elements of the robot designed using soft mechanical structures that provide a high degree of flexibility can make the robot highly compliant, resulting in better locomotion performance over unstructured terrains, improved energy efficiency, enhanced gait control, natural shock absorption capabilities, higher speed of locomotion, and better obstacle avoidance capabilities [14]. For instance, [67] proposes a realized semi-autonomous multi-legged robot with a flexible body and legs along with experimental results. The results indicate improved robot mobility over debris and through narrow spaces compared to conventional robots. The improvement was attributed to the flexibility of the robotic system.

Often, flexible robots are referred to as hyper-redundant robots due to the large or infinite degrees of freedom. A source of inspiration for designing flexible legs comes from nature. Since most organisms have evolved over many generations, the locomotion principles used by these organisms in their respective environments are often highly energy efficient and optimized for locomotion speed. Recent literature [14] has shown that biologically inspired compliant-legged robots designed using elastic elements offer improved dynamic performances compared to rigid-legged counterparts. Notable advantages include improved energy efficiency, enhanced gait control, natural shock absorption capabilities, higher speed of locomotion, and better obstacle avoidance capabilities.

In this thesis, we model the legs of the robot as flexible structures to capitalize on the compliancy and likely improve traversal over complex terrains. The inspiration for the design comes from the flexible axonemal structure of protrusions present on the surface of many fluid-based microorganisms. Overall, the axoneme behaves as an elastic rod-like structure with a high degree of flexibility and possesses an internal set of shear stresses that allow it to move and bend. In biological protrusions, these stresses result from the concurrent effect of dynein motor proteins and passive elastic connections within the structure. Distributed motors and passive mechanical links can generate the same effect in a realistic implementation. In this work, moments are applied at the bases of the legs to drive the robot. The internal system of shear forces acts as a secondary means of actuation and is included in the modeling of the system presented here. However, the internal forces are set to

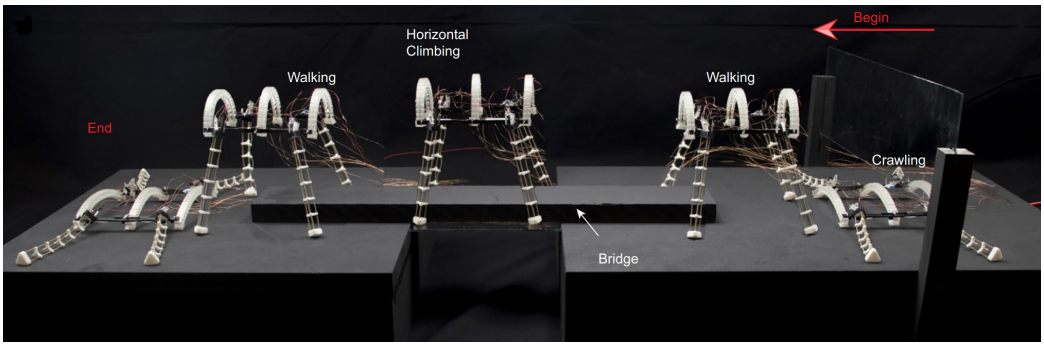


Figure 2.4: Morphing capabilities of an adaptive robot designed using thermal shape memory alloys [68]

zero during the simulation to demonstrate metachronal coordination by applying moments at the bases.

The axonemal structure is specifically chosen due to its extreme flexibility and morphing characteristics. The ability to transform into different complex shapes while being compliant and adaptive on unstructured terrains is highly desirable to tackle environmental constraints. For instance, [68] proposes a flexible robot framework that utilizes thermally activated shape memory polymers to enable the morphing capabilities of the robot. Figure 2.4 depicts an experimental result presented in [68] where the robot performs multimodal terrestrial locomotion due to its extreme flexibility. Similarly, biologically active axonemes of small-scale biological entities have demonstrated these desired flexible structure qualities [69, 70]. The highly flexible nature of the axonemal structure, the internal actuation mechanism, and the potential induction of metachronal waves through basal mechanical coupling make it an extremely desirable candidate for the flexible framework of the robot legs modeled in this thesis. The modeling and results presented in this work show the viability of the proposed locomotion mechanisms and pave the way to address hardware implementation constraints.

2.2.5 Terrain Contact Mechanics

An important aspect to consider while formulating the dynamics of any terrestrial robot is to conceptualize and mathematically model how the robot interacts with the terrain. Contact mechanics are important because the robot can only move on the terrain due to the contact forces that are generated when the robot comes into contact with the terrain. The moment when contact occurs between the robot and the terrain is also called an impact event [71]. From a geometrical perspective,

contact between the leg of the robot and the stationary terrain can be considered as a restriction on the motion of the leg in specific directions when the conditions for an impact event are met. Significantly, the kinematic constraints in the direction of movement of a rotating leg occur both in the normal and tangential directions, which require both normal and frictional forces that act to prevent the motion of the leg along the respective directions. Here, the normal force prevents the leg from penetrating the terrain, while the frictional force prevents the leg from slipping. Therefore, appropriate functions are required to model the impact event based on the geometry of the terrain and the resultant contact forces. Apart from the restriction of movement, the contact forces and impact events also affect mechanical elements such as vibrations, wave propagation, fatigue, wear, and the appearance of cracks [72]. These phenomena are complex to model and occur due to impulsive changes in the direction of motion of the system, resulting in the appearance of high-magnitude reactive forces. Since this work only focuses on checking the viability of the proposed locomotion mechanism and does not involve analyzing the robot from a design perspective, the contact mechanics is limited to modeling the contact forces and conditions for the impact event.

The contact formulations can be obtained using standard methods such as the Lagrange multipliers method and the Penalty method [71]. The Lagrange multipliers method involves using additional unknown variables called Lagrange multipliers to enforce constraints on a system. Therefore, the system states and unknown Lagrange multipliers must be determined to solve the system's dynamics. Though it is an additional complication to solve for Lagrange multipliers, the method allows for exact enforcement of the constraints on the system (i.e., the Lagrange multipliers ensure that a constraint is always fully satisfied). The penalty method does not require the use of additional unknowns. Instead, a penalty function is formulated using the system constraints and penalty parameters that ensure that the imposed constraints are approximately satisfied. Theoretically, the penalty function can be interpreted as a spring-damper system that restricts the motion between two deformable bodies in contact [71, 72]. The spring restricts the motion of one body into the other by applying a reactive force, and the elasticity of the spring models the deformation of the bodies that could occur during the impact event. The damper models the energy dissipation when the bodies come into contact. Here, energy dissipation represents the energy lost by both bodies during the contact process. The energy loss could be visualized in a bouncing ball problem, where the ball loses energy with every bounce in the form of heat, sound, and air friction, which leads

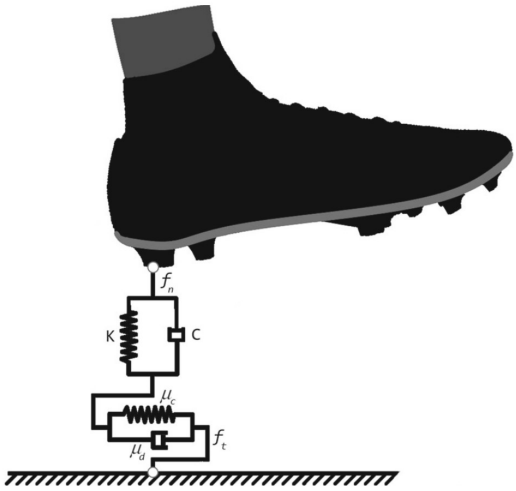


Figure 2.5: Visual depiction of a spring-damper system that models the contact mechanics between a foot and the ground [72]

to a lower bounce height. For instance, Figure 2.5 represents the contact mechanics between a foot and the ground in the form of penalty functions. The normal force f_n is a penalty function with spring stiffness K and damping ratio C that prevents the penetration of the foot into the terrain and represents the energy loss during the impact event. Similarly, the frictional force f_t is modeled with stiffness μ_c and damping coefficient μ_d that prevents slipping of the foot and depicts the energy dissipation during the contact event.

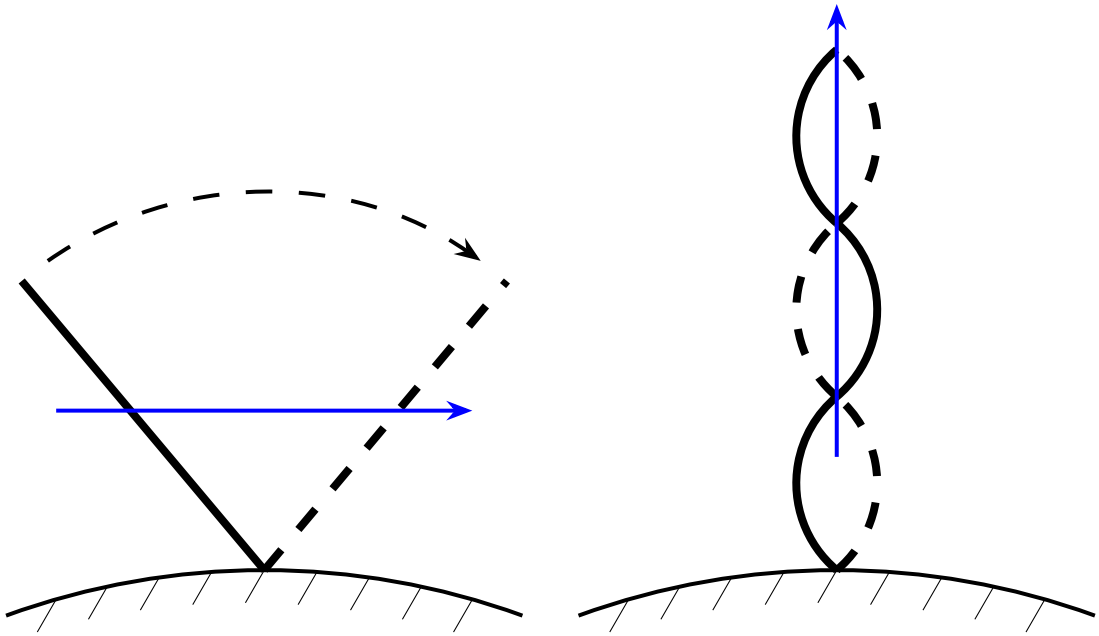
In simulations, the penalty method is preferred due to its simplicity and computational efficiency [72]. Typically, large values are used for penalty parameters that lead to acceptable satisfaction of the system constraints. However, there are drawbacks in the form of identifying penalty parameters for different types of terrains and inclusion of high frequencies in the system due to large penalty parameters for the spring stiffness[72]. There are works present in the literature that specifically address the problems arising due to the penalty methods in different situations [72]. Though the constraint is not precisely satisfied using the penalty method, the advantage is that it does not require solving for additional variables as in the case of the Lagrange multipliers method. This can reduce the computational time needed to solve the differential equations describing the motion of a complex system [73–75].

For this thesis, the proposed multi-legged robot is simulated over a flat and even terrain. The contact mechanics between the robot and the terrain are derived using the penalty method. The penalty functions are formulated using a linear Hooke

contact model where the elastic contact of the robot's payload and tips of the legs with the terrain are represented by linear spring elements acting in normal and tangential directions [71, 76]. At the time of the impact event, the linear springs apply reactive forces in the normal and tangential directions at the tips of the legs, preventing them from penetrating through the terrain while ensuring no slippage. However, there is a modeling simplification considered for the contact mechanics between the robot's body and the terrain. At the time of the impact event between the body and terrain, the penalty model only applies a normal contact force to meet the non-penetration condition. The frictional force between the body and terrain during contact is assumed to be non-existent. Penalty methods such as the Kelvin-Voigt contact model that use an energy dissipator and advanced penalty contact models [72] are not considered in this work since the objective is to portray the qualitative results and show the viability of the locomotion mechanism. Since the robot is simulated over a flat terrain, using advanced penalty contact models seems unessential. Also, the energy dissipation phenomena during impact events are disregarded because the results are not analyzed from an energy consumption perspective.

2.3 Biomechanics of Cilia and Flagella

Many cells and microorganisms possess slender hair-like protrusions called cilia or flagella on the surface of their bodies. The protrusions present on microorganisms do not have a standard size or sparsity. Each organism has a different number of protrusions of different lengths and spacing suited to serve a biological purpose. These protrusions can either act as motile organelles that enable the motion of the organism in fluid environments or as non-motile sensory organelles that detect chemical and mechanical gradients [77]. In most present-day organisms, these protrusions can sense and actuate, regulating their beat cycles in response to chemical cues or mechanical and hydrodynamical stimuli [78]. In general, the normal functions of motile protrusions involve moving fluid relative to the position they are attached to the organism's body. Hence, these protrusions can serve as an aquatic locomotion mechanism for various microorganisms. Arrays of cilia or flagella are also often found as linings in the tracts of digestive, respiratory, excretory, reproductive, and coelomic circulatory systems where they are attached to the tract surface, and they pump the fluid carrying essential materials over the tract by oscillating in specific beat cycles. For instance, ciliary arrays present on the linings of human lungs allow



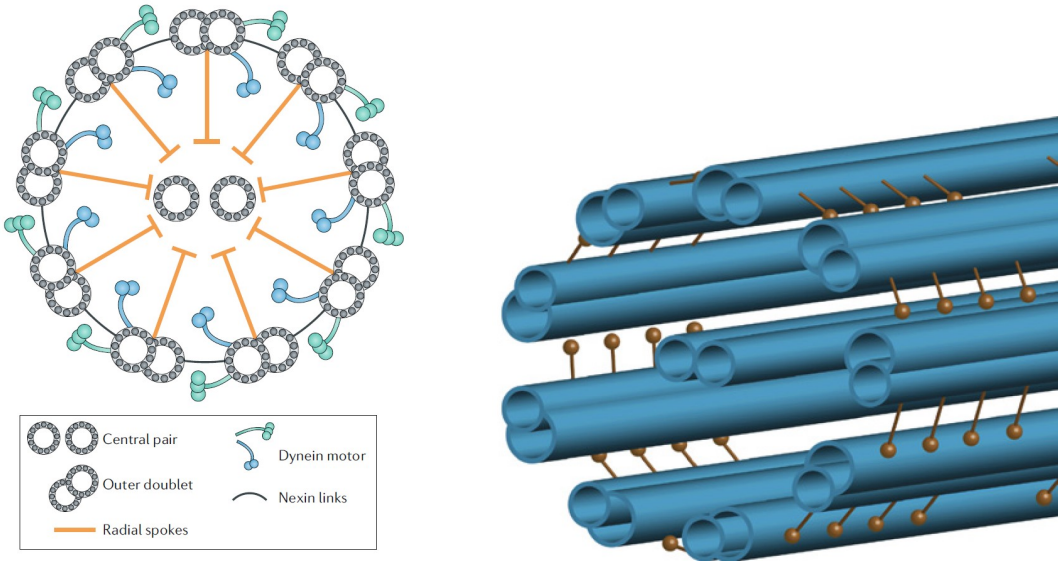
(a) Movement of a cilium (b) Movement of a flagellum

Figure 2.6: Visualization of characteristic movement of a cilium and a flagellum. The solid blue arrow indicates the direction of fluid propulsion, and the dotted lines show the movement of the organelles (reproduced from [77])

movement of mucus [79]. Cilia and flagella interchangeably refer to protrusions because there is no structural distinction. However, the movement and the direction in which a protrusion pushes a fluid serve as a differentiating factor that results in using two different biological terms. Flagella drive the fluid continuously in parallel to the flagellar axis. In contrast, cilia push the fluid orthogonally or tangentially to the ciliary axis in a pulsatile manner during the power strokes [77]. The difference in fluid movement is diagrammatically represented in Figure 2.6. Also, flagella typically refer to sparse or isolated protrusions used by unicellular organisms for locomotion, as opposed to cilia, which refers to a close collective group or a carpet of protrusions that can both stationarily propagate fluid over the surface or enable locomotion of an organism.

2.3.1 Structure of Cilia and Flagella

Despite the biological function differences, a cilium and flagellum have an identical fundamental ultrastructure called the axoneme. They can oscillate due to an internal system of shear stresses generated within this structure. The axoneme is an elastic rod-like structure typically composed of doublets of hollow elastic protein cylinders



(a) Cross-sectional view of the axoneme (b) Three-dimensional view of the axoneme

Figure 2.7: Cross-sectional and three-dimensional view of an axoneme with '9+2' arrangement of microtubule doublets (replicated from [20, 81])

called microtubules or filaments arranged cylindrically and mechanically coupled via flexible links. The elastic nature of these filaments gives bending capabilities to the axonemal structure. However, the filaments are inextensible. The microtubule assembly can also be found in the cytoplasm of many cells and allows the cells to maintain their shape and internal arrangements. The number of microtubule doublets within an axoneme can vary between cilia and flagella of different organisms [80]. However, the canonical structure that is most researched consists of nine microtubule doublets that surround a central pair of microtubules, also known as a '9+2' arrangement. This axoneme structure is illustrated in Figure 2.7.

The elastic links that bundle outer cylindrical pairs into a uniform rod-like structure are called nexin links. All the outer pairs are connected to the central pair of microtubules via radial spokes. Furthermore, the point at which the axoneme is attached to the cell is called the basal end, while the free end of the axoneme is termed the distal end. The central pair of microtubules define the bending axis of the axoneme. The beating of the axoneme is powered by a large motor protein called dynein that is anchored at regular intervals along the length of the outer doublets. The dyneins convert the chemical energy of Adenosine Triphosphate through hydrolysis, which results in the movement of dynein arms towards the basal end of the doublets, thereby generating a sliding force that can drive the doublets apart if the crosslinks between them are removed [82, 83]. The bending of

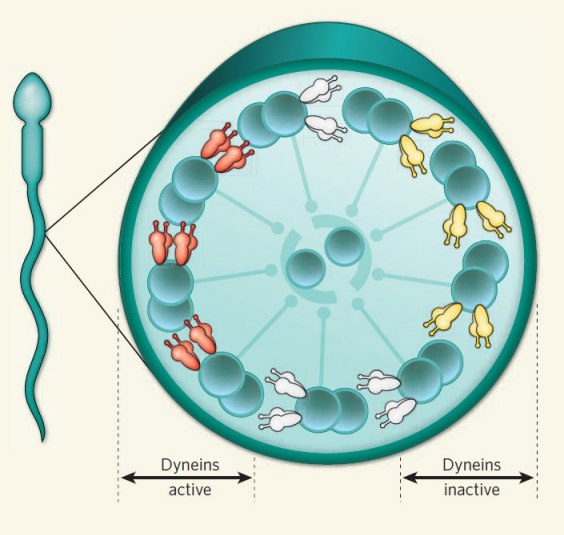


Figure 2.8: Cross-section of an axoneme from a spermatozoon tail. Microtubule doublets are shown as overlapping circles. Active dyneins are represented with the color red, and inactive dyneins are represented with the color yellow (replicated from [84])

the axoneme through dynein motors can also be visualized as a system of internal shear forces that cause the bending of an elastic tube at any point along its neutral axis [70]. If the sliding is permitted globally, the filaments separate without bending. However, the elastic crosslinks themselves are not enough to restrict the sliding. By rigidly connecting one of the two ends of the axoneme, the filaments undergo a bending deformation under the application of a sliding force, and sliding occurs at the other free end. The bending process of the axoneme is depicted in Figure 2.8, where the cross-section of an axoneme from spermatozoa is considered.

In Figure 2.8, the bending of the axoneme is achieved when active dyneins on one side move towards the basal end of the axoneme, while the inactive dyneins on the other side move passively towards the distal end. Since the basal end is rigidly connected, the bending of the system occurs due to the sliding force that results in a sliding displacement between the filaments at the tail. To propagate a beat through the axoneme, activity states of the dyneins on the two sides must switch in a spatially and temporally controlled manner. This aspect of regulating the sliding of the dynein along the microtubule doublets to generate bending of the axoneme is still an active field of research.

There are several hypothesized mechanisms for how the sliding of dynein is regulated to produce axonemal beating. One hypothesis is the 'tug-of-war' model, in which dyneins antagonistically slide on opposite ends of the axoneme, causing

longitudinal forces to continuously build up until dyneins on one side are forced to detach, breaking symmetry and causing the axoneme to bend [69]. Another hypothesis termed the 'geometric clutch' model revolves around the assumption that the axoneme naturally bends due to its shape and external forces. Under that assumption, the axoneme can bend in any desired direction by engaging and disengaging dynein motors based on the amount of alteration of inter-doublet spacing [85]. However, the hypothesis that has gained more traction and experimental support involves a direct coupling between the local curvature of the axoneme and the attachment of the dynein molecules. In this model, the continuous transverse motion of dynein motors along filaments gradually increases the elastic stress in the system, which eventually stalls the motors and leads to a reversal in their walking direction [69, 86].

2.3.2 Axonemal Biomechanics

Before approaching the mathematical modeling of axonemes, there needs to be an overall generalization of the axonemal structure. The structure involves a cylindrical arrangement of elastic filament pairs crosslinked via flexible links and motor proteins. The filaments are highly resistant to stretching and compression but offer no resistance to bending. Hence, the axonemal structure is inextensible and can undergo high bending deformations. The concurrent action of dynein arms generates an internal system of shear stresses that results in the filaments tending to slide with respect to each other. Since the filaments are rigidly connected at the base, the sliding forces cause the axonemal structure to bend instead of the filaments separating away from each other. Also, because the distal end of the structure is free, the sliding forces cause a relative sliding displacement at the tip of the axonemes. Further, the elastic properties of the crosslinks between the filament pairs characterize the resistance to bending of the axonemal structure.

In this thesis, two-dimensional modeling of the axonemal structure is considered for implementing it as the legs of a planar robot's locomotion mechanism. Since the axonemal system can undergo large elastic bending deformations and shear displacements at the tip, the Euler-Bernoulli model cannot explain the dynamics. In [87], the authors show that classical theories like the Timoshenko model and Cosserat rod theories cannot explain certain phenomena and features observed in biological axonemes because of the inconsistency between the traditional shearing mechanics and shearing effects observed in crosslinked filament bundles. In the

context of hydrodynamics, the mechanics of a beating axoneme has been studied by several works in the literature [20, 69, 81, 84, 87–92]. The common theme among these works is that the axoneme itself is not modeled as an inextensible, unshearable, elastic beam or as a general shearable rod. Instead, each filament doublet is modeled as constrained Euler elastica. The filaments are inextensible and inseparable from the bundle of filaments. Also, the bundle of filaments is rigidly coupled at the base, and the internal stresses drive the bending of the axonemal structure with relative sliding displacements at the tip. To capture the complex dynamics of axonemal bending, each model presented in the literature conceptualizes different mechanisms to generate oscillatory deformation patterns of the axoneme.

This thesis only focuses on capturing the desirable high-bending capabilities and the internal actuation mechanism of the axonemal structure for use in future work related to the hardware implementation of the proposed locomotion mechanism. Further, the lightweight and compliant feature of the filaments, as described by constrained Euler elastica, is a highly beneficial feature for legged robots. In contrast, using bulky deformable beam or rod structures as the legs of the robot modeled using existing Timoshenko or Cosserat theories would only add to the inertia of the robotic system. Therefore, instead of looking for specific models that delve into the complex behavior of biological axonemes, this work takes inspiration from [69], where a general class of systems called internally driven filaments characterizes the behavior of biological axonemes. The planar version of this class of systems is shown in Figure 2.9 where two elastic filaments arranged in parallel with a constant separation, effectively describing a flexible rod, are rigidly attached on one end (the basal end of the axoneme) where they are not permitted to slide with respect to each other. Everywhere else, sliding is possible, and deformation of this rod-like structure leads to local sliding displacements between the filaments. The system of filaments is internally driven by a system of shear forces that corresponds to an internal force density that tends to slide the two filaments apart. Also, the elastic filaments are inextensible. In Figure 2.9, the constant distance between the filaments is represented by the quantity a, the neutral axis represented by the vector $\mathbf{r}$ is parameterized by the arclength s, internal distributed system of shear forces dependent on the arclength are represented by $f(s)$, the shapes of the filaments parameterized by the arclength are represented by the quantities $\mathbf{r}_1$ and $\mathbf{r}_2$, and the sliding displacement at the distal end is represented by the quantity $\Delta(s)$.

The planar hydrodynamic equations for a beating axoneme derived in [69] can be solved to determine the shapes of bending waves, swimming velocity, and

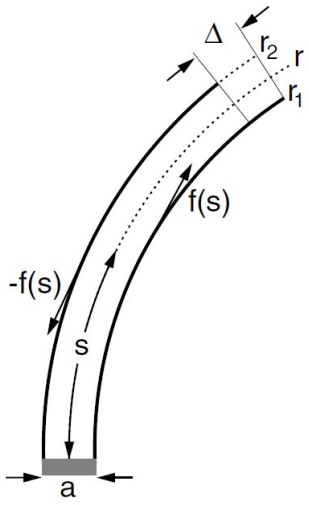


Figure 2.9: Planar model of the axoneme (replicated from [69]). The solid lines represent the elastic filaments. The dotted line represents the neutral axis of the axoneme

tension profile of the axoneme. Recent studies also involve three-dimensional modeling and simulation of a cilium in an oscillating flow to better understand the hydrodynamics of axonemal beating [89]. Other studies also involve the effect of surrounding fluid properties on axonemal beating [93], role of nonlinearity in stable beating [94], and beat cycle stabilization through nonlinearity of dynein recruitment and detachment [95].

2.3.3 Beating Patterns of Axonemal Arrays

The aspect of cilia and flagella that is more interesting is their collective behavior and ability to synchronize and display various beating patterns despite external forces and disturbances arising in the surrounding fluid. These beating patterns emerge over multiple scales ranging from synchronization of flagellar beats of two swimming sperms [96] to metachronal wave formation in ciliary carpets [97]. By assuming that each axoneme executes a circular motion around an axis perpendicular to the surface from which it protrudes, a simple phase oscillator model can describe the motion of an axoneme and the synchronization phenomenon in axonemal arrays. Several forms of synchronization or beat patterns emerge in axonemal arrays. They include in-phase and antiphase synchronization (where all the oscillators have identical phases or antiphases), phase locking (where all the oscillators have identical phase velocities), entrainment (where all the oscil-

lators have identical mean phase velocities), and metachronal wave formation (where all the oscillators are phase-locked with a constant phase difference between each other) [17]. The feature that makes the synchronous behavior of coupled axonemal arrays more interesting is that the beating patterns are not neuronally controlled. All the organisms that possess these axonemes lack a nervous system and can execute beating patterns to produce a range of complex behaviors such as gait switching during micro-swimming and run-and-tumble foraging strategy [98]. Hence, the synchronization dynamics are independently managed within the axonemes through feedback-dependent modulation of the motion of dynein motors along the axonemal axis. The feedback mechanism relies on external cues generated through variations in the hydrodynamic, chemical, and mechanical properties of the surrounding fluid.

Several studies were conducted to understand the origin of synchronization better, and in the process, many theoretical models were derived to formulate the beating dynamics of flagellated microorganisms. Through these theoretical models, the research studies intended to find the reasoning behind synchronization among beating axonemes despite external disturbances in the surrounding fluid. Initially, these studies pointed towards hydrodynamic coupling as the reason behind synchronization phenomena. Through hydrodynamic coupling, the hydrodynamic flow generated by one oscillating axoneme exerts a force on the neighboring oscillating axoneme, which couples them together. One particularly influential study [99] that proved this hypothesis involved modeling hydrodynamic fields generated by two cilia modeled as a pair of spheres undergoing elliptical orbits near a boundary. The resultant synchronization phenomena due to hydrodynamic coupling depended on the distance between cilia and characteristic parameters of effective and recovery strokes of ciliary beat. Further, models of ciliary chains also supported the hypothesis that phase locking and metachronal coordination arise due to hydrodynamic interactions when specific parameters such as beat wavelength, axoneme rigidity, and interciliary distance were within an acceptable threshold [17]. However, the whole dynamics of ciliary coupling are more complex than phase synchronization alone. Numerical simulations of beating axonemes show that other higher-order effects, such as mechanical bistability and load-dependent decoupling, can also occur [100, 101].

More recent studies have shown that in organisms with sparsely spaced axonemes, such as flagella of Chlamydomonas, the hydrodynamic forces alone do not explain the coupling. Since the axonemes are sparse, the far-field hydrodynamic

forces are an order of magnitude lower than those required for hydrodynamic coupling in physiological conditions. Instead, the synchronization can be due to a feature called basal coupling. Here, basal coupling refers to mechanical substructures that couple the bases of the axonemes. These basal substructures or internal fibers act as a direct mechanical linkage between the bases of the axonemes which allows the transfer of forces from one oscillating axoneme to another. This hypothesis was experimentally proven by comparing the flagellar beat cycles of wild Chlamydomonas and mutant Chlamydomonas that are deficient in filamentary connections between basal bodies of the flagella [21, 102]. Even in arrays of mammalian cilia where the axonemal arrays are dense, ciliary roots and basal feet structures continue to provide additional resistance to fluid stresses [103].

2.3.4 Arrays of Beating Axonemes for Terrestrial Locomotion

To a large extent, the biomechanics of axonemal beating and beating patterns of axonemal arrays have been widely explored in the context of fluid environments since they are the fundamental mechanisms involved in micro-swimming and micro-pumping of fluids by organisms. The recent work in [21] proposed that the synchronization of beat patterns and the emergence of metachronal waves in axonemal arrays is primarily due to the basal coupling between axonemes in addition to lower order effects of hydrodynamic coupling. This also suggests that if the surrounding fluid is removed, beating patterns in axonemal arrays can be achieved by pure basal coupling. Pairs of oscillating axonemes in a low-viscosity environment (such as air) that beat synchronously through basal coupling is similar to Huygens clocks [104]. In the Huygens clock experiment, two oscillating pendula tend towards synchrony or anti-synchrony if attached to a common support whose flexibility provides the necessary coupling. The flexible support enables the transfer of mechanical vibrational forces by traveling waves from one pendulum to another. The traveling waves couple the oscillator and enable synchronization. This also hints toward using pure basal coupling in axonemal arrays for metachronal coordination.

In this thesis, we intend to utilize the research on axonemal structures and beating patterns of axonemal arrays in fluid environments for terrestrial robot locomotion. Specifically, each axoneme can act as an individual leg of a multi-legged robot. The flexible yet stiff structure of the axoneme makes it an excellent candidate for the legs of the robot. Further, the axonemes can also improve the compliance

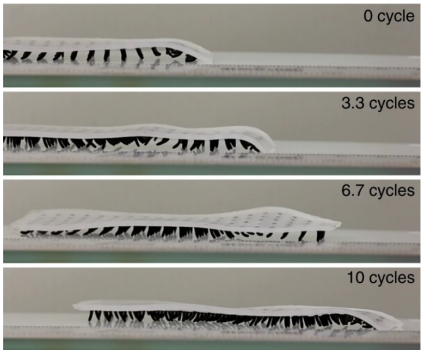


Figure 2.10: Demonstration of terrestrial locomotion using metachronal coordination in a magnetic ciliary carpet

and robustness of the terrestrial robot, as its flexibility allows the axonemes to adapt and morph to unstructured terrains, making it easier for the legged robot to navigate a variety of terrains. Also, the internal forces can act as an alternate source of the actuation mechanism for the robot, as opposed to applying moments at the joints of the legs of the robot. From a mechanical design point of view, an engineered version of the axoneme-inspired legs would have a flexible structure within which a system of internal shear forces is generated through passive and active links reproduced via distributed motors and viscoelastic links.

The basal coupling between the axonemes can aid metachronal coordination, an extremely optimal gait for multi-legged robot locomotion. Also, instead of actuating and controlling each axoneme to follow a specific oscillation cycle for metachronal coordination, the basal coupling can reduce the actuation inputs to each axoneme, thereby minimizing energy consumption. Basal coupling can be realized using mechanical coupling elements such as linear/nonlinear springs, dampers, or a combination of those elements.

In [105], planar dynamics of a basally coupled array of axonemes have been derived. The work also includes exploring different feedback mechanism schemes for basal coupling, such as rotation, curvature, and tension of the axonemes. The mathematical modeling of the robot presented in this thesis is inspired by the modeling approach used in [69, 105]. Another notable research work using metachronal coordination in ciliary carpets for various soft robotic applications, including terrestrial locomotion, is [60]. An experimental result of [60] is shown in Figure 2.10 where an artificially fabricated magnetic ciliary carpet is actuated through an externally generated magnetic field. The controlled actuation generated metachronal wave patterns, resulting in the forward movement of the robot.

Chapter 3

Robot Locomotion Mechanism Modeling

3.1 Proposed Robot Model

The model of the locomotion mechanism is inspired by the mobility principles utilized by microorganisms as discussed in Chapter 2. Specifically, the interest is in using surface protrusions to possibly generate beating patterns through basal coupling. Here basal coupling of protrusions refers to a mechanical linkage that connects the bases of these protrusions. Further, these protrusions have the structure of an axoneme, which is cylindrical, flexible, and may have internal forces. The objective of the thesis is to lay the foundation for using an array of basally coupled flexible elements for the terrestrial locomotion of mobile robots, leveraging on emergent coordinated patterns that are suitable for the purpose. The possibility of sustaining coordinated patterns via basal coupling is explored, suggesting that this is a possible actuation strategy complementary to distributed actuation along the filaments. Future work includes studying the connection between actuation and emergent coordination patterns for control design.

Hence, the axonemal structure of cilia or flagella is used to model the individual legs. Further, to sustain metachronal coordination, each leg of the robot is coupled at the base using linear springs. Also, to simplify the dynamics, the body is assumed to be rigid. The generalized robot model is schematized in Figure 3.1. The basis vectors $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ form the global coordinate frame for the environment and the robot. The robot model is presented for N legs. In total, there are $2N + 1$ springs connecting the bases of the legs and the robot's body. For simplicity, the linear

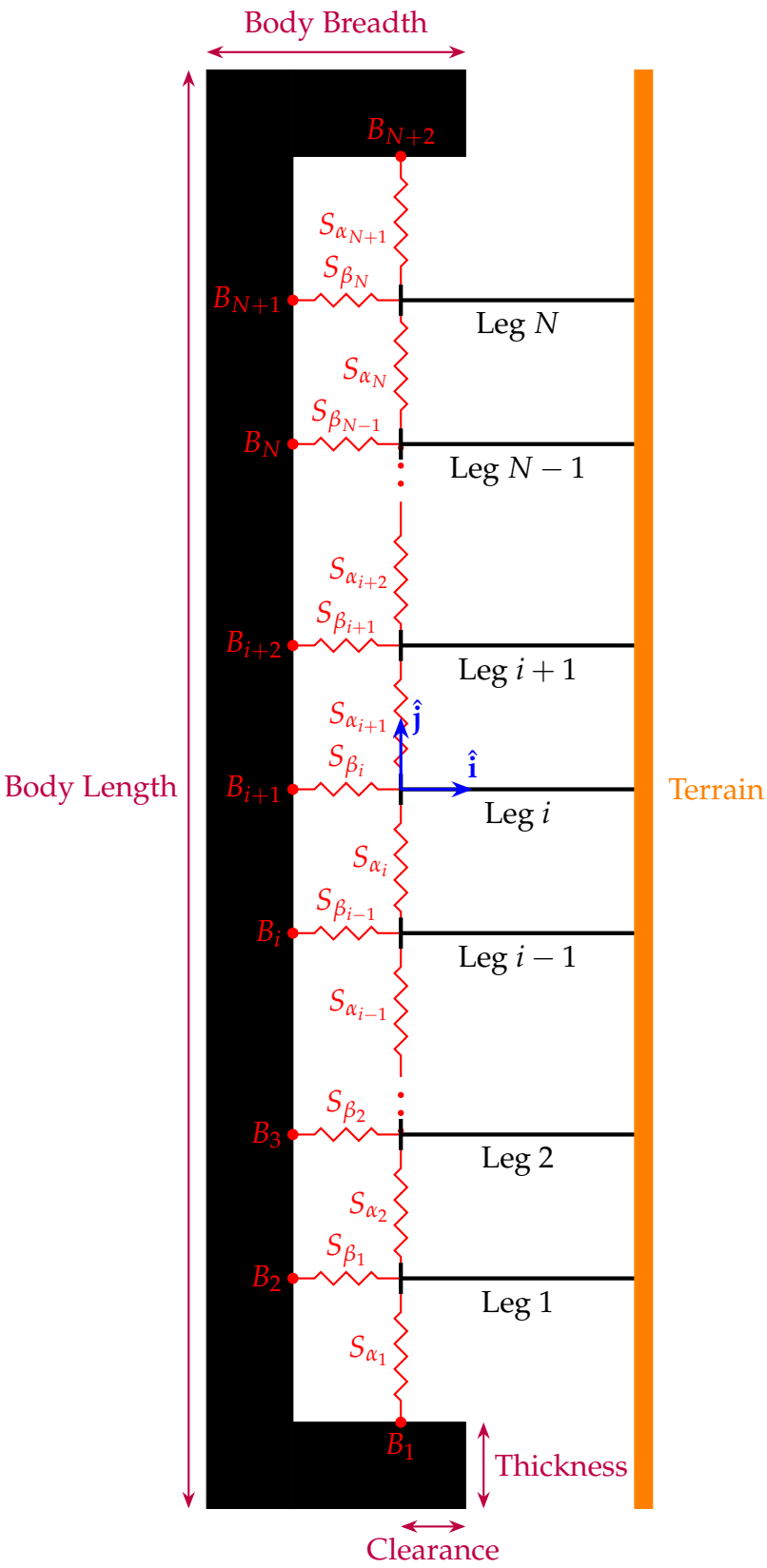
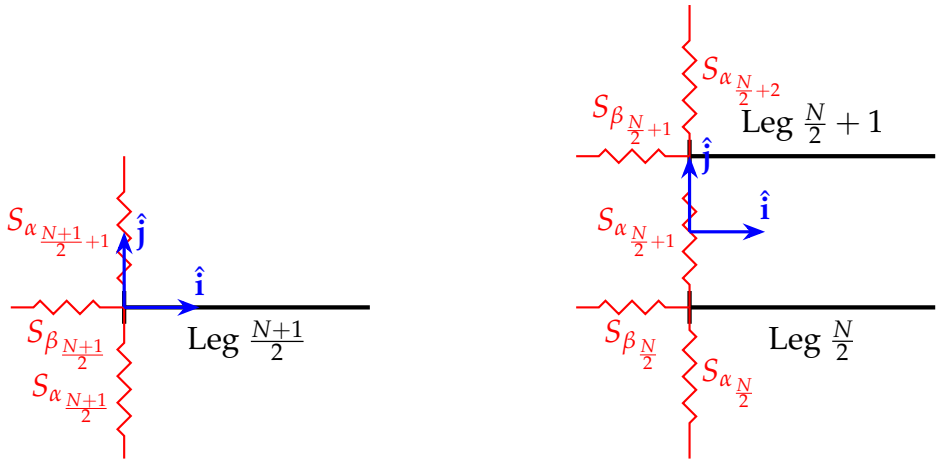


Figure 3.1: Schematic of the robot model



(a) For odd number of legs (b) For even number of legs

Figure 3.2: Location of the basis vectors based on the number of legs

springs $S_{\alpha_1}, \ldots, S_{\alpha_i}, \ldots, S_{\alpha_{N+1}}$ have the same spring constants α and the same initial lengths L_α. Similarly, the linear springs $S_{\beta_1}, \ldots, S_{\beta_i}, \ldots, S_{\beta_N}$ have the same spring constants β and the same initial lengths L_β. The springs S_{α_1}, S_{β_1}, S_{β_2}, $\ldots$, S_{β_i}, $\ldots$, $S_{\beta_{N-1}}$, S_{β_N}, $S_{\alpha_{N+1}}$ are attached to $N+2$ points B_1, B_2, B_3, $\ldots$, B_{i+1}, $\ldots$, B_N, B_{N+1}, B_{N+2} on the body and all the springs are initially undeformed and present in the configuration represented in Figure 3.1.

In order to simplify calculations of the coordinates, the global coordinate frame is positioned such that the robot's body is symmetrical about the basis vector $\hat{\mathbf{i}}$ and the bases of all the legs are along the basis vector $\hat{\mathbf{j}}$. As depicted in Figure 3.2, the global coordinate frame is initially placed at the base of the $(\frac{N+1}{2})^{th}$ leg if N is odd, or midway between the bases of the $(\frac{N}{2})^{th}$ and $(\frac{N}{2}+1)^{th}$ leg if N is even. The robot's payload has a U-shape with uniform thickness as schematized in Figure 3.1. The dimensional labels of body length, breadth, and thickness describe the body's shape. The dimensional label clearance describes the distance between the point B_1 and the edge of the body nearest to B_1 facing the terrain. Since the body is symmetric, the clearance is the same for the distance between the point B_{N+2} and the edge of the body nearest to B_{N+2} facing the terrain.

The robot's movement is simulated on a flat and even terrain. The body's orientation is initially considered to be parallel to the terrain. Since the terrain is a straight line parallel to the basis vector $\hat{\mathbf{j}}$, the equation describing the shape of the terrain is $x = X_{terr}$, where X_{terr} is the parallel distance between the basis vector $\hat{\mathbf{j}}$ and the terrain. The normal and frictional forces arising due to the contact between the tip of the legs and the terrain are modeled through penalty functions. The

robot's movement is enabled by applying time-varying moments at the base of each leg. The moments rotate the leg to execute a circular motion at the desired frequency. Also, the applied moments are controlled to provide a desired phase shift among contiguous legs. Due to the generated contact forces between the rotating leg and the terrain, the robot moves in the desired direction. Since this thesis is the first iteration of this robot locomotion model, we do not explore the extent to which the basal mechanical coupling due to springs aids in metachronal coordination. Instead, this thesis shows that the proposed robot model is a viable terrestrial locomotion solution by applying suitable moments at the base of the legs to ensure the desired phase shift in rotation cycles of the legs, thereby enabling metachronal coordination.

The fundamental blocks of the model are:

- Dynamics of individual flexible legs
- Dynamics of the robot's rigid body
- Contact mechanics between the terrain and the legs

3.2 Generalized Coordinates of the Robot's Body

Before the dynamics of the legs can be derived, we require the generalized coordinates and properties of the robot's body in order to calculate the coordinates of the points $B_1, \ldots, B_{N+2}$ and determine the reactive spring forces acting on the base of each leg. Let the mass of the robot's body be m. The body has three degrees of freedom since it is a planar rigid body in a two-dimensional space. The generalized coordinates of the body are represented by the vector $\mathbf{q}_{body}(t) = (X_{body}(t), Y_{body}(t), \theta(t))$ where, X_{body} and Y_{body} are the Cartesian coordinates of the body's center of mass, and θ is the orientation of the body about the center of mass, positive in the anti-clockwise direction with respect to the vertical basis vector $\hat{\mathbf{j}}$. The robot's body is depicted in Figure 3.3 along with the generalized coordinates and the dimensional labels. The length (τ_L) and breadth (τ_B) of the robot's body are determined from the body's thickness (τ_T), body's clearance (τ_C), and the initial lengths of the springs L_α and L_β. The length, breadth, and area (Π_A) of the robot's body are

$$\tau_L = 2\tau_T + (N+1)L_\alpha \tag{3.1a}$$

$$\tau_B = \tau_T + L_\beta + \tau_C \tag{3.1b}$$

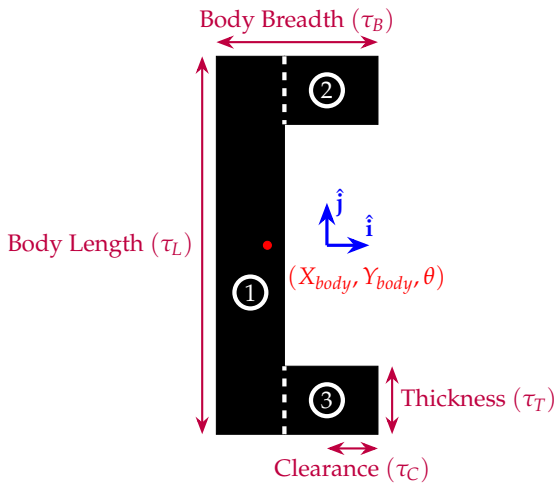


Figure 3.3: Schematic of the robot's body. The body is divided into three regions to determine the coordinates of the center of mass

$$\Pi_A = \tau_L \tau_T + 2(\tau_B - \tau_T)(\tau_T) \tag{3.1c}$$

The body is divided into three regions as represented in Figure 3.3 to calculate the initial coordinates of the center of mass. Since the springs are undeformed initially, the coordinates are

$$X_{body}(0) = \frac{\left(-L_\beta - \frac{\tau_T}{2}\right)(\tau_L)(\tau_T) + 2\left(\frac{-L_\beta + \tau_C}{2}\right)(\tau_B - \tau_T)(\tau_T)}{\Pi_A} \tag{3.2a}$$

$$Y_{body}(0) = 0 \tag{3.2b}$$

Let the coordinates of the points $B_1, \ldots, B_{N+2}$ at time t be collected in the vector $\mathbf{r}_B(t)$. Since the body can rotate by θ radians in the anticlockwise direction about the center of mass $(X_{body}(t), Y_{body}(t))$, the coordinates of point B_i, i.e., $\mathbf{r}_{B_i}(t)$ is determined using the following

$$\begin{aligned} \mathbf{r}_{B_i}(t) &= \mathbf{r}_C(t) + \mathbf{R}(\theta(t))\mathbf{r}_{B_iC}(0) \\ &= \begin{bmatrix} X_{body}(t) \\ Y_{body}(t) \end{bmatrix} + \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) \end{bmatrix} \mathbf{r}_{B_iC(0)} \end{aligned} \tag{3.3}$$

where $\mathbf{r}_C(t) = (X_{body}(t), Y_{body}(t))$ is the coordinates of the center of mass of the body at time t, $\mathbf{R}(\theta(t))$ is a rotation matrix, and $\mathbf{r}_{B_iC}(0)$ is the coordinates of point B_i with respect to the center of mass of the body at time $t = 0$. The process of determining the coordinates of point B_i is illustrated in Figure 3.4.

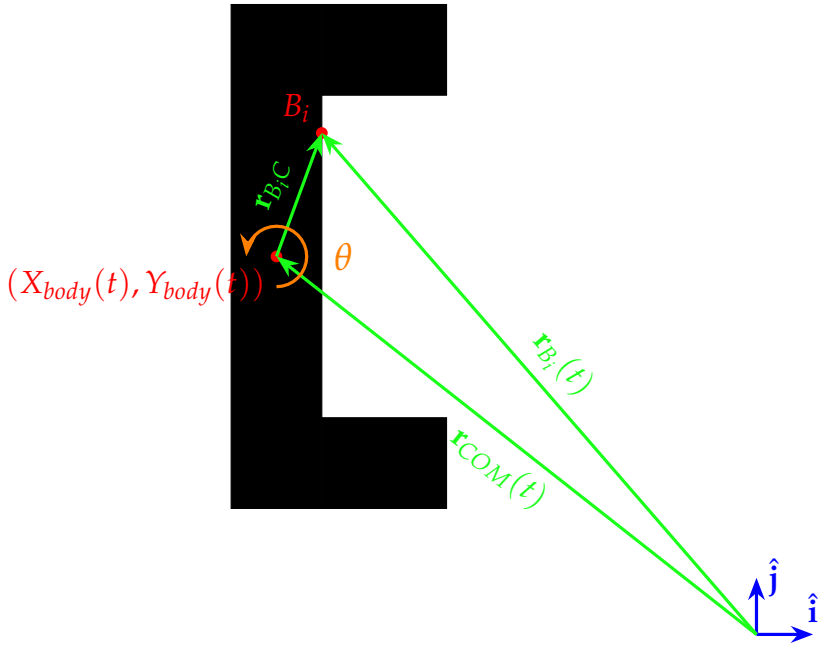


Figure 3.4: Illustration of the calculation of coordinates of point B_i

The contents of the vector $\mathbf{r}_{B_iC}(0)$ are

$$\mathbf{r}_{B_iC}(0) = \begin{cases} \left(-X_{body}(0), -\frac{N+1}{2}L_\alpha\right) & \text{for } i = 1 \\ \left(-X_{body}(0) - L_\beta, -(\frac{N+1}{2} - (i-1))L_\alpha\right) & \text{for } 2 \leq i \leq N+1 \\ \left(-X_{body}(0), \frac{N+1}{2}L_\alpha\right) & \text{for } i = N+2 \end{cases} \tag{3.4}$$

3.3 Leg Dynamics

The planar structure and mathematical modeling of the legs presented in this thesis are inspired by [69, 105]. The axoneme is approximately a cylindrical surface with a constant radius that is flexible and can be actuated by internal forces. As explained in Section 2.3.2, a suitable simplification of the axonemal structure in a two-dimensional space would involve two flexible yet inextensible filaments separated by a constant distance that possess internal forces that can bend the structure. The simplified two-dimensional model of the axoneme is illustrated in Figure 3.5 [69, 105].

The two filaments are separated by approximately a constant distance a. The arclength s spans the neutral axis of the axoneme, and the position vector $\mathbf{r}(s) = (X(s), Y(s))$ parameterized by the arclength describes the position of all points on

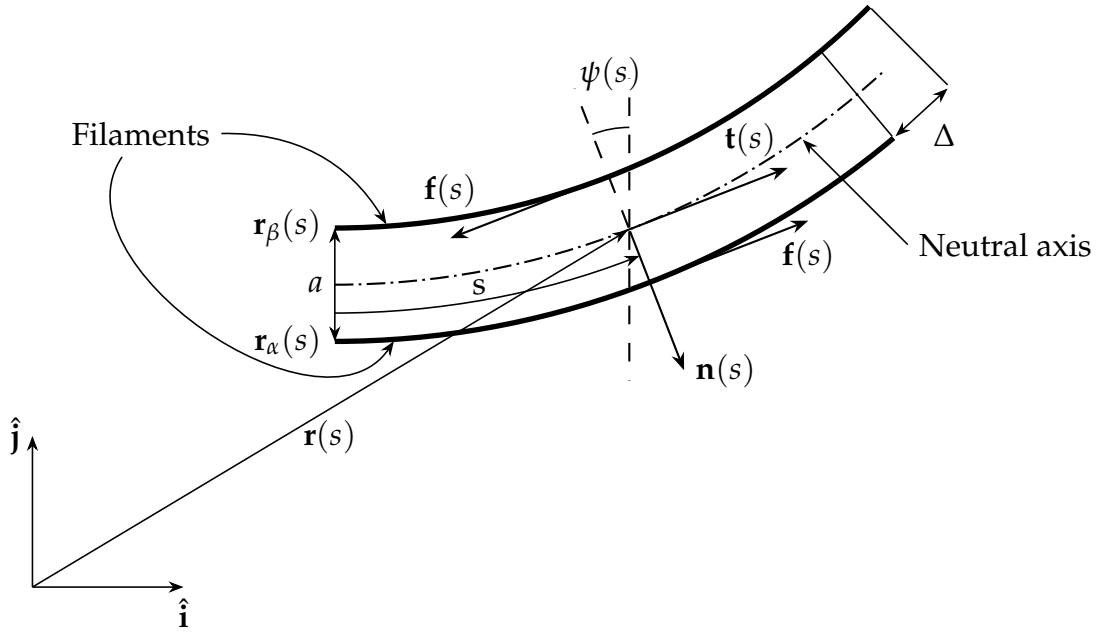


Figure 3.5: Schematic of the simplified two-dimensional axonemal structure (reproduced from [105])

the neutral axis with respect to the global coordinate system. Here X and Y are the $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ components of the position vector $\mathbf{r}(s)$ respectively. The elastic filaments are described using the position vector of the neutral axis as

$$\mathbf{r}_\alpha(s) = \mathbf{r}(s) + \frac{a}{2}\mathbf{n}(s) \tag{3.5a}$$

$$\mathbf{r}_\beta(s) = \mathbf{r}(s) - \frac{a}{2}\mathbf{n}(s) \tag{3.5b}$$

where $\mathbf{n}(s)$ is the unit normal vector and $\mathbf{t}(s)$ is the unit tangent vector to the neutral axis at the arclength s. The local tangent angle to the neutral axis, with respect to the global coordinate frame, is given by $\psi(s)$. The bending of the system is driven by an internal system of shear forces $\mathbf{f}(s)$, which is equivalent to an internal couple or bending moment of magnitude $a\mathbf{f}$. The magnitude of $\mathbf{f}(s)$, which equals $f(s)$, can be interpreted as a force per unit length acting at position s in opposing directions on the two microtubules [69]. This internal force density corresponds to a shear stress within the system of filaments that induces a local strain in the form of relative sliding displacement $\Delta(s)$ between the filaments. Since the arclength s defines the neutral axis, the local geometry can be characterized by the Frenet-Serret frame relations [106]

$$\mathbf{r}' = \mathbf{t} \tag{3.6a}$$

$$\mathbf{t}' = \kappa \mathbf{n} \tag{3.6b}$$

$$\mathbf{n}' = -\kappa \mathbf{t} \tag{3.6c}$$

where κ is the curvature of the neutral axis. Throughout this thesis, the prime symbol denotes derivatives with respect to the arclength s, i.e., $\mathbf{r}' \equiv \partial \mathbf{r}/\partial s$. Since, both the normal and tangent vectors are unit vectors, $\mathbf{t}' = \kappa \mathbf{n} \implies \|\mathbf{r}''\| = \kappa$. Hence, $\kappa = \|\mathbf{r}''\| = \sqrt{X''^2 + Y''^2}$. Since $\psi(s)$ represents the angle between the tangent vector and the basis vector $\hat{\mathbf{i}}$, the value of the local tangent angle is $\psi = atan2(X', Y')$. Here, $atan2$ is the two-argument arctangent, which returns the angle in the range $(-\pi, \pi]$. The relative sliding displacement Δ of the filaments is related to the curvature of the neutral axis by [69]

$$\Delta(s) = a \int_0^s \kappa(\sigma) d\sigma \tag{3.7}$$

The equations of motion can be derived using the extended Hamilton's principle. Let $\mathbf{r}_i(s_i, t) = (X_i(s_i, t), Y_i(s_i, t))$ be the generalized coordinates describing the neutral axis of the i^{th} leg with $s_i \in [0, \ell]$ and ℓ being the length.

3.3.1 Kinetic Energy and Potential Energy

The kinetic energy of the i^{th} leg is given by

$$\mathcal{K}_i = \frac{1}{2} \int_0^\ell \rho \dot{\mathbf{r}}_i \cdot \dot{\mathbf{r}}_i ds_i \tag{3.8}$$

where ρ represents the linear mass density. Throughout this thesis, the superimposed dot means the time derivative, i.e., $\dot{\mathbf{r}} \equiv \partial \mathbf{r}/\partial t$. The leg is modeled as a nonlinear rod, governed by curvature with a distributed system of internal shear forces $\mathbf{f}_i(s, t)$ and external forces and moments. Since the internal system of shear stresses causes the relative sliding displacement of the filaments, and the curvature governs the bending of the system, the strain energy of the i^{th} filament is [69, 107]

$$\mathcal{E}_{strain_i} = \int_0^\ell \left(\frac{K_M}{2} \kappa_i^2 + f_i \Delta_i \right) ds_i \tag{3.9}$$

where K_M denotes the total bending rigidity of the filaments and $f_i = \|\mathbf{f}_i\|$ is the magnitude of $\mathbf{f}_i$. Note that the curvature $\kappa_i(s)$ denotes the local strain caused by the bending of the filament. Therefore, for an infinitesimal section ds_i, the elastic potential energy due to the bending of the filaments is $\frac{1}{2} K_M \kappa_i^2 ds_i$. Hence, the line integral of that expression gives the elastic energy of one filament due to bending

[107]. Also, the internal force density or force per until length f_i corresponding to the shear stress is the dual of the sliding displacement Δ_i, contributing by the term $\int f_i \Delta_i ds_i$ towards the strain energy of the system [69]. The flexible filaments representing the legs are inextensible. The local inextensibility condition is given by $\mathbf{r}_i' \cdot \mathbf{r}_i' = 1$ [69, 107] which ensures that s_i is the arclength. Typically, the constraint is enforced by introducing a Lagrange multiplier. Specifically, there would be N additional Lagrange multiplier variables to enforce the inextensibility constraints. In this thesis, we use a penalty-based approach [108]. The penalty-based method is chosen over the Lagrange multiplier method to avoid the introduction of additional variables to the equations and thereby reduce the computational complexity. A drawback of the penalty-based method is that it brings forth the problem of tuning the penalty parameters. Large penalty numbers can ensure convergence of the constraint at the cost of poor numerical conditioning and very large round-off errors [108]. In this thesis, the penalty numbers were tuned through trial and error, and the convergence of the constraint will be illustrated by plotting the solutions of the equations. The constraint equation of the i^{th} filament is

$$\int_0^\ell \left(\mathbf{r}_i' \cdot \mathbf{r}_i' - 1\right) ds_i \tag{3.10}$$

The gravitational potential energy of the i^{th} filament is

$$\begin{aligned} \mathcal{E}_{grav_i} &= \int_0^\ell \left(\rho g (X_{terr} - X_i)\right) ds_i \\ &= \int_0^\ell \left(\rho g (X_{terr} - \mathbf{r}_i \cdot \hat{\mathbf{i}})\right) ds_i \end{aligned} \tag{3.11}$$

Therefore the total potential energy of the i^{th} filament is

$$\mathcal{E}_i = \int_0^\ell \left(\frac{K_M}{2}\kappa_i^2 + f_i \Delta_i + \rho g (X_{terr} - \mathbf{r}_i \cdot \hat{\mathbf{i}})\right) ds_i \tag{3.12}$$

Using Equation (3.7), after partial integration of Equation (3.12) we obtain

$$\mathcal{E}_i = \int_0^\ell \left(\frac{K_M}{2}\kappa_i^2 - aF_i\kappa_i + \rho g (X_{terr} - \mathbf{r}_i \cdot \hat{\mathbf{i}})\right) ds_i \tag{3.13}$$

where $F_i(s) = -\int_s^\ell f_i(\sigma) d\sigma$ [69]. Note that the strain energy due to the deformation of the springs has not been included in the potential energy equation. Instead, the reactive forces generated due to the deformed springs acting on the bases of the

legs will be introduced through the principle of virtual work for ease of calculation.

3.3.2 Damping Forces

The Rayleigh dissipation function is used to model the damping of each filament and can be written as [109]

$$\mathcal{R}_i = \frac{\zeta}{2}\int_0^{\ell} \dot{\mathbf{r}}_i \cdot \dot{\mathbf{r}}_i ds_i \tag{3.14}$$

where ζ is a constant dissipation constitutive parameter. The damping force, which is non-conservative, is given by the negative velocity gradient of the dissipation function [109]. Hence, the damping force on the i^{th} filament is given by

$$\mathbf{F}_{D_i} = -\frac{\partial \mathcal{R}_i}{\partial \dot{\mathbf{r}}_i} = -\zeta \int_0^{\ell} \dot{\mathbf{r}}_i ds_i \tag{3.15}$$

Then, the virtual work done by the damping forces on the i^{th} filament is

$$\delta \overline{W}_{D_i} = \mathbf{F}_{D_i} \cdot \delta \mathbf{r}_i = -\zeta \int_0^{\ell} \dot{\mathbf{r}}_i \cdot \delta \mathbf{r}_i ds_i \tag{3.16}$$

3.3.3 External Forces and Moments

For each leg, there is a reactive spring force acting at the base, a time-varying moment applied at the base, a normal force due to the terrain normal contact at the tip, and a frictional force due to the terrain tangential contact acting at the tip. The reactive spring forces acting on different legs of the robot is illustrated in Figure 3.6. Let the change in the length of the spring S_{α_i} be $\Delta x_{\alpha_i} = L_\alpha - \gamma_{\alpha_i}$ where

$$\gamma_{\alpha_i} = \begin{cases} \|\mathbf{r}_1(0,t) - \mathbf{r}_{B_1}(t)\| & \text{for } i = 1 \\ \|\mathbf{r}_i(0,t) - \mathbf{r}_{i-1}(0,t)\| & \text{for } 2 \leq i \leq N \\ \|\mathbf{r}_N(0,t) - \mathbf{r}_{B_{N+2}}(t)\| & \text{for } i = N+1 \end{cases} \tag{3.17}$$

with L_α being the initial length of the springs $S_{\alpha_1}, \ldots, S_{\alpha_{N+1}}$. Let the change in the length of the spring S_{β_i} be $\Delta x_{\beta_i} = L_\beta - \gamma_{\beta_i}$ where

$$\gamma_{\beta_i} = \begin{cases} \|\mathbf{r}_1(0,t) - \mathbf{r}_{B_2}(t)\| & \text{for } i = 1 \\ \|\mathbf{r}_i(0,t) - \mathbf{r}_{B_{i+1}}(t)\| & \text{for } 2 \leq i \leq N-1 \\ \|\mathbf{r}_N(0,t) - \mathbf{r}_{B_{N+1}}(t)\| & \text{for } i = N \end{cases} \tag{3.18}$$

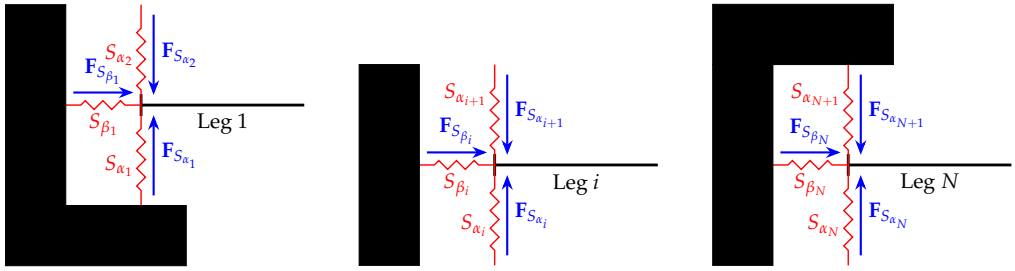


Figure 3.6: Depiction of reactive spring forces acting at the base of the legs

with L_β being the initial length of the springs $S_{\beta_1}, \ldots, S_{\beta_N}$. Note that there are $N+1$ S_α and N S_β springs. The reactive spring forces acting at the base of the i^{th} leg is given by

$$\mathbf{F}_{S_i} = \begin{cases} \mathbf{F}_{S_{\alpha_1}} + \mathbf{F}_{S_{\beta_1}} + \mathbf{F}_{S_{\alpha_2}} & \text{for } i = 1 \\ \mathbf{F}_{S_{\alpha_i}} + \mathbf{F}_{S_{\beta_i}} + \mathbf{F}_{S_{\alpha_{i+1}}} & \text{for } 2 \leq i \leq N-1 \\ \mathbf{F}_{S_{\alpha_N}} + \mathbf{F}_{S_{\beta_N}} + \mathbf{F}_{S_{\alpha_{N+1}}} & \text{for } i = N \end{cases} \tag{3.19}$$

Substituting the related reactive spring force terms gives

$$\mathbf{F}_{S_i} = \begin{cases} \dfrac{\alpha \Delta x_{\alpha_1}}{\gamma_{\alpha_1}}(\mathbf{r}_1(0,t) - \mathbf{r}_{B_1}(t)) \\ + \dfrac{\beta \Delta x_{\beta_1}}{\gamma_{\beta_1}}(\mathbf{r}_1(0,t) - \mathbf{r}_{B_2}(t)) & \text{for } i = 1 \\ + \dfrac{\alpha \Delta x_{\alpha_2}}{\gamma_{\alpha_2}}(\mathbf{r}_1(0,t) - \mathbf{r}_2(0,t)) \\ \\ \dfrac{\alpha \Delta x_{\alpha_i}}{\gamma_{\alpha_i}}(\mathbf{r}_i(0,t) - \mathbf{r}_{i-1}(0,t)) \\ + \dfrac{\beta \Delta x_{\beta_i}}{\gamma_{\beta_i}}(\mathbf{r}_i(0,t) - \mathbf{r}_{B_{i+1}}(t)) & \text{for } 2 \leq i \leq N-1 \\ + \dfrac{\alpha \Delta x_{\alpha_{i+1}}}{\gamma_{\alpha_{i+1}}}(\mathbf{r}_i(0,t) - \mathbf{r}_{i+1}(0,t)) \\ \\ \dfrac{\alpha \Delta x_{\alpha_N}}{\gamma_{\alpha_N}}(\mathbf{r}_N(0,t) - \mathbf{r}_{N-1}(0,t)) \\ + \dfrac{\beta \Delta x_{\beta_N}}{\gamma_{\beta_N}}(\mathbf{r}_N(0,t) - \mathbf{r}_{B_{N+1}}(t)) & \text{for } i = N \\ + \dfrac{\alpha \Delta x_{\alpha_{N+1}}}{\gamma_{\alpha_{N+1}}}(\mathbf{r}_N(0,t) - \mathbf{r}_{B_{N+2}}(t)) \end{cases} \tag{3.20}$$

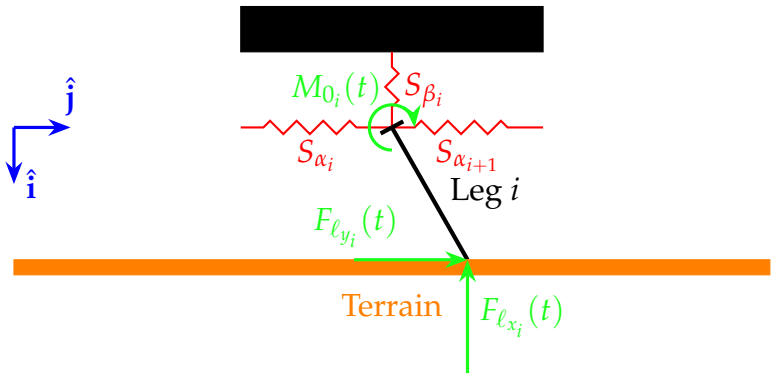


Figure 3.7: Depiction of external forces and moments acting on the i^{th} leg

Let a time-varying moment applied at the base of the i^{th} leg be $M_{0_i}(t)$. As outlined in Section 2.2.3, to execute a metachronal gait, each contiguous leg of the robot is initially assumed to be phase-shifted by a constant phase difference. Therefore, metachronal coordination and forward movement of the robot is enforced by applying moments simultaneously to the bases of all the legs and rotating them at a constant frequency in the clockwise direction. By rotating the legs at a constant desired frequency through controlled moments, the initial phase difference between contiguous legs would remain approximately the same throughout the locomotion process. Let the desired frequency of rotation be ω. Hence, the moment to be applied at the base of the i^{th} leg to rotate it at the desired frequency is

$$\begin{aligned} M_{0_i}(t) &= K_p\left(-\omega - \dot{\psi}_i(0,t)\right) \\ &= K_p\left(-\omega - \frac{\partial(atan2(X_i'(0,t), Y_i'(0,t)))}{\partial t}\right) \end{aligned} \tag{3.21}$$

where K_p is the proportional gain and $\dot{\psi}_i(0,t)$ represents the rate of change of the local tangent angle at the base of the i^{th} leg. $\psi_i(0,t)$ is also referred to as the basal angle of the i^{th} leg. The negative sign before ω indicates a negative rate of change of basal angle, thereby rotating the leg in the desired clockwise direction.

During the rotation, when the tip of an individual leg touches the terrain, it experiences contact forces in the form of frictional force and normal force as illustrated in Figure 3.7. Here, the direction of the normal force is opposite to the direction of the basis vector $\hat{\mathbf{i}}$ and since the leg rotates clockwise, the direction of the frictional force is along the direction of the basis vector $\hat{\mathbf{j}}$. The normal force and the frictional force acting on the i^{th} leg are represented by the quantities $F_{\ell_{x_i}}(t)$ and $F_{\ell_{y_i}}(t)$ respectively. They are modeled via penalty functions that act on the tip of a leg whenever there

is contact [71]. Since the terrain is flat, the tip of the i^{th} leg touches the terrain if $X_i(\ell, t) = X_{terr}$. The penalty functions for the normal and frictional forces are [71]

$$F_{\ell_{x_i}}(t) = \begin{cases} \epsilon \left(X_{terr} - X_i(\ell, t)\right) & \text{if } X_i(\ell, t) \geq X_{terr} \\ 0 & \text{otherwise} \end{cases} \tag{3.22a}$$

$$F_{\ell_{y_i}}(t) = \begin{cases} \nu \left(-\dot{Y}_i(\ell, t)\right) & \text{if } X_i(\ell, t) \geq X_{terr} \\ 0 & \text{otherwise} \end{cases} \tag{3.22b}$$

where ϵ and ν are penalty constants. Here, the penalty functions act as springs with very high stiffness that prevent the movement of the tip of the legs when in contact with the terrain [71]. For the normal force, whenever the i^{th} leg touches the terrain, a force $\epsilon \left(X_{terr} - X_i(\ell, t)\right)$ is applied at the tip to prevent the leg from penetrating the terrain. Similarly, the frictional force prevents slipping of the tip of the i^{th} leg, i.e., $Y_i(\ell, t) = constant \implies \dot{Y}_i(\ell, t) = 0$ when the leg is in contact with the terrain. Therefore, a frictional force of $\nu \left(-\dot{Y}_i(\ell, t)\right)$ is applied at the tip of the i^{th} leg when it makes contact with the terrain. Note that the negative sign for the frictional force indicates that it is applied opposite to the velocity of the tip of the i^{th} filament. Let $\mathbf{F}_{\ell_i} = (F_{\ell_{x_i}}, F_{\ell_{y_i}})$ denote the contact forces acting at the tip of the i^{th} leg. Then, the virtual work done by the moment and the reactive spring forces at the base, along with the virtual work done by the contact forces on the tip of the i^{th} leg, is given by

$$\delta\overline{W}_{Ext_i} = M_{0_i}\delta\psi_i(0, t) + \mathbf{F}_{S_i} \cdot \delta\mathbf{r}_i(0, t) + \mathbf{F}_{\ell_i} \cdot \delta\mathbf{r}_i(\ell, t) \tag{3.23}$$

Note that, the variation of the curvature is $\delta\kappa_i = \delta(\mathbf{n}_i \cdot \mathbf{r}_i'') = \mathbf{n}_i \cdot \delta\mathbf{r}_i''$, since from Equation (3.6), it can be derived that $\mathbf{r}_i'' \cdot \delta\mathbf{n}_i = \kappa_i \mathbf{n}_i \cdot \delta\mathbf{n}_i = 0$ [105]. Using, $\delta\kappa_i = \mathbf{n}_i \cdot \delta\mathbf{r}_i''$, the variation of the local tangent angle is derived as

$$\begin{aligned} \psi_i = \int \kappa_i ds \implies \delta\psi_i = \int \delta\kappa_i ds = \int \mathbf{n}_i \cdot (\delta\mathbf{r}_i')' ds \\ = \mathbf{n}_i \cdot \delta\mathbf{r}_i' - \int \mathbf{n}_i' \cdot \delta\mathbf{r}_i' ds = \mathbf{n}_i \cdot \delta\mathbf{r}_i' + \int \kappa \mathbf{t}_i \cdot \delta\mathbf{t}_i ds = \mathbf{n}_i \cdot \delta\mathbf{r}_i' \\ (\text{Since } \mathbf{t}_i \cdot \mathbf{t}_i = 1 \implies \mathbf{t}_i \cdot \delta\mathbf{t}_i = 0) \\ \text{Then, } \delta\psi_i = \mathbf{n}_i \cdot \delta\mathbf{r}_i' = \frac{\mathbf{r}_i''}{\|\mathbf{r}_i''\|} \cdot \delta\mathbf{r}_i' \end{aligned} \tag{3.24}$$

Since $\mathbf{t}_i = (X_i', Y_i')$, $\mathbf{n}_i = (-Y_i', X_i')$ [106], we have $\delta\psi_i = -Y_i'\delta X_i' + X_i'\delta Y_i'$. Therefore, the virtual work done by external forces and moments is

$$\delta\overline{W}_{Ext_i} = M_{0_i}\frac{\mathbf{r}_i''(0,t)}{\|\mathbf{r}_i''(0,t)\|}\cdot\delta\mathbf{r}_i'(0,t) + \mathbf{F}_{S_i}\cdot\delta\mathbf{r}_i(0,t) + \mathbf{F}_{\ell_i}\cdot\delta\mathbf{r}_i(\ell,t) \tag{3.25}$$

3.3.4 Extended Hamilton's Principle

The Lagrangian of the i^{th} leg is denoted by

$$\mathcal{L}_i = \mathcal{K}_i - \mathcal{E}_i \tag{3.26}$$

By using the extended Hamilton's principle, the variation of the action from the Lagrangian between two arbitrary time instants t_1 and t_2, along with the virtual work done by the damping forces, external forces and moments gives

$$\int_{t_1}^{t_2}\left(\delta\mathcal{K}_i - \delta\mathcal{E}_i + \delta\overline{W}_{D_i} + \delta\overline{W}_{ext_i}\right)dt = 0 \tag{3.27}$$

Integrating the variation of the kinetic energy between the time instants gives

$$\int_{t_1}^{t_2}\delta\mathcal{K}_i dt = -\int_{t_1}^{t_2}\left(\int_0^{\ell}\rho\ddot{\mathbf{r}}_i\cdot\delta\mathbf{r}_i ds_i\right)dt \tag{3.28}$$

The variation of the potential energy gives

$$\begin{aligned}\delta\mathcal{E}_i = &\int_0^{\ell}\left(K_M\mathbf{r}_i^{\mathrm{IV}} - \left(\frac{aF_i}{\|\mathbf{r}_i''\|}\mathbf{r}_i''\right)'' - \rho g\hat{\mathbf{i}}\right)\cdot\delta\mathbf{r}_i ds_i \\ &+ \left(K_M - \frac{aF_i}{\|\mathbf{r}_i''\|}\right)\mathbf{r}_i''\cdot\delta\mathbf{r}_i'\Big|_0^{\ell} - \left(K_M\mathbf{r}_i^{\mathrm{III}} - \left(\frac{aF_i}{\|\mathbf{r}_i''\|}\mathbf{r}_i''\right)'\right)\cdot\delta\mathbf{r}_i\Big|_0^{\ell}\end{aligned} \tag{3.29}$$

Details of calculations for the variations of the kinetic energy and potential energy are presented in Appendix A.1. Substituting the variations of kinetic energy and potential energy along with the virtual work done by the damping forces (3.16), external forces and moments (3.25) in Equation (3.27), we get the equations of motion of the i^{th} leg as

$$\rho\ddot{\mathbf{r}}_i + \zeta\dot{\mathbf{r}}_i + K_M\mathbf{r}_i^{\mathrm{IV}} - \left(\frac{aF_i}{\|\mathbf{r}_i''\|}\mathbf{r}_i''\right)'' - \rho g\hat{\mathbf{i}} = \mathbf{0} \tag{3.30}$$

with

$$\int_0^{\ell}\left(\mathbf{r}_i'\cdot\mathbf{r}_i' - 1\right)ds_i = 0 \tag{3.31}$$

as the inextensibility constraint of the filament. The boundary conditions are

$$s = \ell : \begin{cases} \mathbf{F}_{\ell_i} = -K_M \mathbf{r}_i^{\mathrm{III}}(\ell, t) + \left(\dfrac{aF_i}{\|\mathbf{r}_i''\|} \mathbf{r}_i'' \right)' \Big|_{s=\ell} \\ 0 = K_M \|\mathbf{r}_i''(\ell, t)\| \end{cases} \tag{3.32a}$$

$$s = 0 : \begin{cases} \mathbf{F}_{S_i} = K_M \mathbf{r}_i^{\mathrm{III}}(0, t) - \left(\dfrac{aF_i}{\|\mathbf{r}_i''\|} \mathbf{r}_i'' \right)' \Big|_{s=0} \\ M_{0_i} = -K_M \|\mathbf{r}_i''(0, t)\| + aF_i(0, t) \end{cases} \tag{3.32b}$$

where $\mathbf{F}_{\ell_i}$, $\mathbf{F}_{S_i}$, and M_{0_i} are the vector of contact forces, vector of spring forces, and external moment applied on the i^{th} leg. The boundary conditions are all dynamic or von Neumann, accounting for the actions of the restoring springs and the actuating moment at the base. Also, the free-end or contact-with-terrain condition at the tip, assuming that the moment is zero also when in contact, contributes to the dynamic boundary conditions. Note that the initial conditions of the partial differential equations shown in Equation (3.30) cannot be stated yet since it is a generalized equation for N legs and would require defining geometric parameters of the legs in their initial state. Further, the ad hoc choice of starting the robot from a metachronal state where each contiguous leg is phase-shifted by a certain desired amount makes generalizing the initial condition for an N-legged robot difficult. Therefore, the initial conditions will be introduced in Chapter 4 where certain parameters are defined for the robotic system to be simulated.

3.3.5 Weak Form

The numerical solution of the boundary value problem posed in Equation (3.30) and Equation (3.32) can be found via the weak formulation of the problem, then applying the finite element method to determine an approximate solution. The weak formulation turns the differential equation into an integral equation through which a solution with less stringent requirements (for instance, the solution can satisfy lower-order derivative conditions) can be found. To obtain the weak form, we multiply the partial differential equation by the virtual displacement $\delta \mathbf{r}_i$ and integrate it over the spatial domain $[0, \ell]$. Then, we apply integration by parts to

different terms and substitute the boundary conditions to obtain

$$\begin{aligned}\int_0^{\ell} \Bigg(& \left(\rho \ddot{\mathbf{r}}_i + \zeta \dot{\mathbf{r}}_i - \rho g \hat{\mathbf{i}}\right) \cdot \delta \mathbf{r}_i + \left(K_M - \frac{aF_i}{\|\mathbf{r}_i''\|}\right) \mathbf{r}_i'' \cdot \delta \mathbf{r}_i'' \Bigg) ds_i \\ & - \mathbf{F}_{\ell_i} \cdot \delta \mathbf{r}_i(\ell, t) - \mathbf{F}_{S_i} \cdot \delta \mathbf{r}_i(0, t) - M_{0_i} \frac{\mathbf{r}_i''(0,t)}{\|\mathbf{r}_i''(0,t)\|} \cdot \delta \mathbf{r}_i'(0, t) = 0\end{aligned} \tag{3.33}$$

Details regarding the calculations to obtain the weak form can be found in Appendix A.2. From Equation (3.24), we have

$$\frac{\mathbf{r}_i''(0,t)}{\|\mathbf{r}_i''(0,t)\|} \cdot \delta \mathbf{r}_i'(0,t) = \delta \psi_i(0,t) = -Y_i'(0,t)\delta X_i'(0,t) + X_i'(0,t)\delta Y_i'(0,t) \tag{3.34}$$

Substituting $\mathbf{r}_i = (X_i, Y_i)$ and Equation (3.34) in Equation (3.33), we get

$$\begin{aligned}\int_0^{\ell} \Bigg(& \rho \left(\ddot{X}_i \delta X_i + \ddot{Y} \delta Y_i\right) + \zeta \left(\dot{X}_i \delta X_i + \dot{Y}_i \delta Y_i\right) - \rho g \delta X_i \\ & + \left(K_M - \frac{aF_i}{\sqrt{X_i''^2 + Y_i''^2}}\right) \left(X_i'' \delta X_i'' + Y_i'' \delta Y_i''\right) \Bigg) ds_i \\ & - F_{\ell_{x_i}}(t)\delta X_i(\ell, t) - F_{\ell_{y_i}}(t)\delta Y_i(\ell, t) - F_{S_{x_i}}(t)\delta X_i(0,t) - F_{S_{y_i}}(t)\delta Y_i(0,t) \\ & - M_{0_i}(t)\left(-Y_i'(0,t)\delta X_i'(0,t) + X_i'(0,t)\delta Y_i'(0,t)\right) = 0\end{aligned} \tag{3.35}$$

where $F_{S_{x_i}}$ and $F_{S_{y_i}}$ are the x and y components of the reactive spring force vector $\mathbf{F}_{S_i}$.

Therefore, in the context of numerical approximation methods, there are in total $2N$ trial solution fields (X_i, Y_i) to which we associate the $2N$ test function fields $(\delta X_i, \delta Y_i)$, or in the context of mechanical systems, the test functions are virtual displacements [110].

3.3.6 Finite Element Discretization

To obtain a numerical solution of Equation (3.35), the set of trial solutions X_i and Y_i and the related test functions δX_i and δY_i are projected on finite element basis functions. The domain of each i^{th} leg $[0, \ell]$ is discretized by n nodes with coordinates $s_{i_j} \in [0, \ell]$, for $i = 1, \ldots, N$ and $j = 1, \ldots, n$. Trial functions and test functions are

approximated by the expansions [110, 111]

$$X_i(s_i,t) = \sum_{k=1}^{n} \left(\phi_{\alpha_k}(s_i)a_{i_k}(t) + \phi_{\beta_k}(s_i)b_{i_k}(t)\right) \tag{3.36a}$$

$$\delta X_i(s_i,t) = \sum_{k=1}^{n} \left(\phi_{\alpha_k}(s_i)\delta a_{i_k}(t) + \phi_{\beta_k}(s_i)\delta b_{i_k}(t)\right) \tag{3.36b}$$

$$Y_i(s_i,t) = \sum_{k=1}^{n} \left(\phi_{\alpha_k}(s_i)c_{i_k}(t) + \phi_{\beta_k}(s_i)d_{i_k}(t)\right) \tag{3.36c}$$

$$\delta Y_i(s_i,t) = \sum_{k=1}^{n} \left(\phi_{\alpha_k}(s_i)\delta c_{i_k}(t) + \phi_{\beta_k}(s_i)\delta d_{i_k}(t)\right) \tag{3.36d}$$

where $\phi_{\alpha_1},\ldots,\phi_{\alpha_n}$ and $\phi_{\beta_1},\ldots,\phi_{\beta_n}$ are the interpolating polynomials and the coefficient sets $a_{i_1},\ldots,a_{i_n}$, $b_{i_1},\ldots,b_{i_n}$, $c_{i_1},\ldots,c_{i_n}$, and $d_{i_1},\ldots,d_{i_n}$ are the unknown time varying amplitudes. Note that the approximations shown in Equation (3.36) are formulated to meet continuity requirements. Since the weak form shown in Equation (3.35) consists of integrals involving terms X_i'' and Y_i'', the second derivative of the interpolating polynomials must be square-integrable, i.e., $\int_0^\ell (\phi_{\alpha_j}'')^2 ds < \infty$ and $\int_0^\ell (\phi_{\beta_j}'')^2 ds < \infty$ for all $j = 1,\ldots,n$. Interpolating polynomials satisfying such requirements are often termed Lagrange polynomials in H^2 [110]. Note that the set of Lagrange polynomials used for both the trial solutions X_i and Y_i are the same since they have to satisfy the same continuity requirements. To choose a suitable interpolating polynomial, an approach that is used to tackle a similar problem in the application of the Galerkin weighted-residual method for weak formulation of the Euler-Bernoulli method is applied in this thesis. To satisfy the compatibility and continuity requirements, a third-order interpolating polynomial is required, which consists of four unknown constants[110, 111]. The following properties of the Lagrange polynomials are used to determine the unknown constants of each interpolating polynomial [110]

$$\phi_{\alpha_j}(s_{i_k}) = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases}, \quad \phi'_{\alpha_j}(s_{i_k}) = 0;\ j,k = 1,\ldots,n \tag{3.37a}$$

$$\phi'_{\beta_j}(s_{i_k}) = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases}, \quad \phi_{\beta_j}(s_{i_k}) = 0;\ j,k = 1,\ldots,n \tag{3.37b}$$

In other words, evaluating the trial solutions at the node locations gives $X_i(s_{i_k}, t) = a_{i_k}(t)$, $X_i'(s_{i_k}, t) = b_{i_k}(t)$, $Y_i(s_{i_k}, t) = c_{i_k}(t)$, and $Y_i'(s_{i_k}, t) = d_{i_k}(t)$. Example plots of the finite element basis functions are shown in Appendix A.3. The Lagrange polynomials are collected into vectors $\boldsymbol{\phi}_\alpha = [\phi_{\alpha_1}, \ldots, \phi_{\alpha_n}]$, $\boldsymbol{\phi}_\beta = [\phi_{\beta_1}, \ldots, \phi_{\beta_n}]$, and $\boldsymbol{\phi} = [\boldsymbol{\phi}_\alpha, \boldsymbol{\phi}_\beta]$. Similarly, the unknown time-varying coefficients are collected into vectors $\mathbf{a}_i = [a_{i_1}, \ldots, a_{i_n}]$, $\mathbf{b}_i = [b_{i_1}, \ldots, b_{i_n}]$, $\mathbf{c}_i = [c_{i_1}, \ldots, c_{i_n}]$, $\mathbf{d}_i = [d_{i_1}, \ldots, d_{i_n}]$, $\mathbf{q}_{X_i} = [\mathbf{a}_i, \mathbf{b}_i]$, $\mathbf{q}_{Y_i} = [\mathbf{c}_i, \mathbf{d}_i]$, and $\mathbf{q}_i = [\mathbf{q}_{X_i}, \mathbf{q}_{Y_i}]$. Therefore, the trial functions and test functions can be rewritten in a simpler form as

$$X_i(s_i, t) = \boldsymbol{\phi}\mathbf{q}_{X_i}^T \tag{3.38a}$$

$$\delta X_i(s_i, t) = \delta\mathbf{q}_{X_i}\boldsymbol{\phi}^T \tag{3.38b}$$

$$Y_i(s_i, t) = \boldsymbol{\phi}\mathbf{q}_{Y_i}^T \tag{3.38c}$$

$$\delta Y_i(s_i, t) = \delta\mathbf{q}_{Y_i}\boldsymbol{\phi}^T \tag{3.38d}$$

Substituting the projections (3.38) in the weak form (3.35), we obtain the finite element discrete form, which has to hold for all virtual displacements $\delta\mathbf{q}_i$ as

$$\rho\mathbf{M}_i\ddot{\mathbf{q}}_i + \zeta\mathbf{M}_i\dot{\mathbf{q}}_i + \mathbf{N}_i\mathbf{q}_i - \mathbf{U}_i = 0 \tag{3.39}$$

where $\mathbf{q}_i$ is a list of length $4n$ consisting of all unknown coefficients, $\mathbf{M}_i$ is the mass matrix of dimensions $4n \times 4n$, $\mathbf{N}_i$ is a $4n \times 4n$ nonlinear matrix that accounts for the nonlinear dynamics of the leg, and $\mathbf{U}_i$ is the input vector of length $4n$ that depends on the external forces and moments acting on the leg. The details of the elements of the matrices can be found in Appendix A.4.

To enforce the inextensibility constraint on the filaments, we consider the penalty system formulated in [108], where a fictitious mass-spring-damper system based on the constraints is added to the system's Lagrangian. We first introduce the projected constraint along the finite element basis functions by substituting (3.38) in (3.10). For the i^{th} filament, we obtain

$$\begin{aligned}
\mathcal{V}_i &= \int_0^\ell \left(\mathbf{r}_i'.\mathbf{r}_i' - 1\right) ds_i \\
&= \int_0^\ell \left(X_i'^2 + Y_i'^2 - 1\right) ds_i \\
&= \mathbf{q}_i \begin{bmatrix} \int_0^\ell \left(\boldsymbol{\phi}'^T\boldsymbol{\phi}'\right) ds_i & \mathbf{0} \\ \mathbf{0} & \int_0^\ell \left(\boldsymbol{\phi}'^T\boldsymbol{\phi}'\right) ds_i \end{bmatrix} \mathbf{q}_i^T - \ell
\end{aligned} \tag{3.40}$$

In [108], the Lagrangian is augmented by the following term

$$\frac{\lambda}{2}\left(\left(\frac{d\mathcal{V}_i}{dt}\right)^2 - 4\eta\mu\left(\frac{d\mathcal{V}_i}{dt}\right)\mathcal{V}_i - \eta^2\mathcal{V}_i^2\right) \tag{3.41}$$

where λ, η, and μ are penalty parameters, with λ being a large constant, and η and μ representing the natural frequency and damping ratio of the fictitious penalty oscillator. In Equation (3.41), the first term is analogous to kinetic energy, the second term is a dissipator, and the last one is a fictitious storage term. Since the constraint $\mathcal{V}_i(\mathbf{q}_i)$ is holonomic in the finite element nodal coordinates $\mathbf{q}_i$, the penalty system adds the following term to the discretized system's dynamics [108]

$$\mathbf{L}_i = \lambda\left(\frac{d^2\mathcal{V}_i}{dt^2} + 2\eta\mu\frac{d\mathcal{V}_i}{dt} + \eta^2\mathcal{V}_i\right)\frac{\partial\mathcal{V}_i}{\partial\mathbf{q}_i}^T \tag{3.42}$$

Substituting Equation (3.42) into Equation (3.39), the finite element discrete form of the i^{th} leg becomes

$$\rho\mathbf{M}_i\ddot{\mathbf{q}}_i + \zeta\mathbf{M}_i\dot{\mathbf{q}}_i + \mathbf{N}_i\mathbf{q}_i + \mathbf{L}_i - \mathbf{U}_i = 0 \tag{3.43}$$

In total, there are $4n \times N$ equations describing the dynamics of N legs.

3.4 Robot's Body Dynamics

3.4.1 Kinetic Energy and Potential Energy

The kinetic energy of the body is given by

$$\mathcal{K}_{body} = \frac{1}{2}m\left(\dot{X}_{body}^2 + \dot{Y}_{body}^2\right) + \frac{1}{2}\mathcal{I}_{body}\dot{\theta}^2 \tag{3.44}$$

where

$$\mathcal{I}_{body} = \frac{1}{12}m\left(\left(\tau_L^2 + \tau_B^2\right) - \left((\tau_L - 2\tau_T)^2 + (\tau_B - \tau_T)^2\right)\right) \tag{3.45}$$

is the moment of inertia of the robot's body about its center of mass, with τ_L, τ_B, and τ_T being the length, breadth, and thickness of the body, respectively. The potential energy is given by

$$\mathcal{E}_{body} = mg\left(X_{terr} - X_{body}\right) \tag{3.46}$$

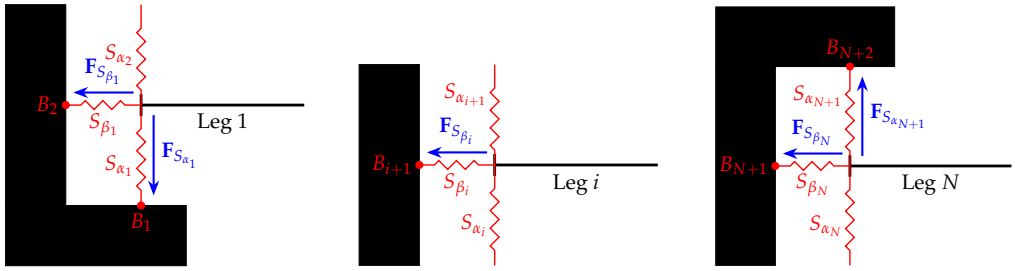


Figure 3.8: Depiction of reactive spring forces acting on the body of the robot

Note that the strain energy due to the deformation of the springs has not been included in the potential energy. Instead, similar to the approach in modeling the legs, we consider the virtual work done by the reactive spring forces.

3.4.2 Damping Forces

The Rayleigh dissipation function is used to model the damping of the robot's body and can be written as

$$\begin{aligned} \mathcal{R}_{body} &= \frac{\zeta}{2}\left(\dot{\mathbf{q}}_{body} \cdot \dot{\mathbf{q}}_{body}\right) \\ &= \frac{\zeta}{2}\left(\dot{X}_{body}^2 + \dot{Y}_{body}^2 + \dot{\theta}^2\right) \end{aligned} \tag{3.47}$$

where ζ is the constant dissipation constitutive parameter. The damping force, which is non-conservative, is given by the negative velocity gradient of the dissipation function. Hence, the generalized damping force vector acting on the body is given by

$$\mathbf{Q}_{D_{body}} = -\frac{\partial \mathcal{R}_{body}}{\partial \dot{\mathbf{q}}_{body}} \tag{3.48}$$

3.4.3 Reactive Spring Forces

The reactive forces due to the springs S_{α_1}, S_{β_1}, S_{β_2}, ..., S_{β_N}, and $S_{\alpha_{N+1}}$ act on the points B_1, B_2, B_3, ..., B_{N+1}, and $B_N + 2$ respectively, as shown in Figure 3.8. Let the reactive spring forces be collected in the list $\mathbf{F}_{S_{body}}(t)$. The reactive spring force

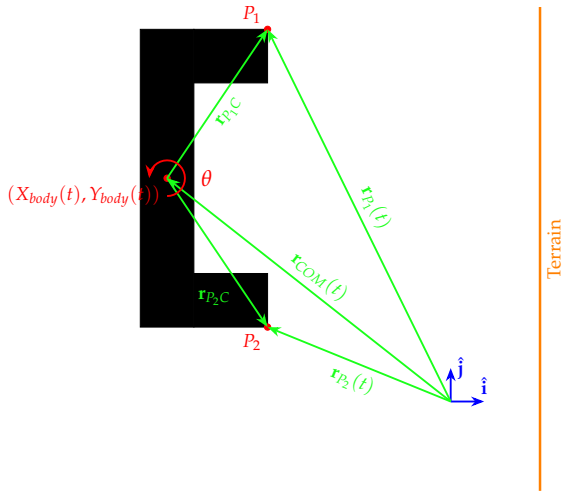


Figure 3.9: Illustration of the calculation of coordinates of points P_1 and P_2

acting on the point B_i is given by

$$\mathbf{F}_{S_{body_i}} = \begin{cases} \dfrac{\alpha \Delta x_{\alpha_1}}{\gamma_{\alpha_1}} (\mathbf{r}_{B_1}(t) - \mathbf{r}_1(0,t)) & \text{for } i = 1 \\ \dfrac{\beta \Delta x_{\beta_{i-1}}}{\gamma_{\beta_{i-1}}} (\mathbf{r}_{B_i}(t) - \mathbf{r}_{i-1}(0,t)) & \text{for } 2 \leq i \leq N+1 \\ \dfrac{\alpha \Delta x_{\alpha_{N+1}}}{\gamma_{\alpha_{N+1}}} (\mathbf{r}_{B_{N+2}}(t) - \mathbf{r}_N(0,t)) & \text{for } i = N+2 \end{cases} \tag{3.49}$$

The generalized reactive spring force vector is

$$\mathbf{Q}_{S_{body}} = \sum_{i=1}^{N+2} \mathbf{F}_{S_{body_i}} \cdot \frac{\partial \mathbf{r}_{B_i}}{\partial \mathbf{q}_{body}} \tag{3.50}$$

3.4.4 Normal Force

When the robot's body touches the terrain, it experiences a normal force. Note that, for simplicity, the frictional force is neglected when the body makes contact with the terrain. The shape of the payload is chosen such that points P_1 and P_2, as shown in Figure 3.9, can describe the contact boundary. Firstly, the coordinates of points P_1 and P_2 are calculated as

$$\mathbf{r}_{P_1}(t) = \begin{bmatrix} X_{body}(t) \\ Y_{body}(t) \end{bmatrix} + \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) \end{bmatrix} \begin{bmatrix} |X_{body}(0)| + clc \\ \frac{N+1}{2} L_\alpha + tns \end{bmatrix} \tag{3.51a}$$

$$\mathbf{r}_{P_2}(t) = \begin{bmatrix} X_{body}(t) \\ Y_{body}(t) \end{bmatrix} + \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) \end{bmatrix} \begin{bmatrix} |X_{body}(0)| + clc \\ -\frac{N+1}{2}L_\alpha - tns \end{bmatrix} \tag{3.51b}$$

The action of contact normal forces acting on points P_1 and P_2 are simulated using penalty functions that are formulated as follows [71]

$$F_{N_{P_{1_x}}}(t) = \begin{cases} \begin{cases} \epsilon(X_{terr} - \mathbf{r}_{P_{1_x}}(t)) & \text{if } \mathbf{r}_{P_{1_x}}(t) \geq X_{terr} \\ 0 & \text{otherwise} \end{cases} & \text{if } \theta(t) \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.52a}$$

$$F_{N_{P_{2_x}}}(t) = \begin{cases} \begin{cases} \epsilon(X_{terr} - \mathbf{r}_{P_{2_x}}(t)) & \text{if } \mathbf{r}_{P_{2_x}}(t) \geq X_{terr} \\ 0 & \text{otherwise} \end{cases} & \text{if } \theta(t) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.52b}$$

where $\mathbf{r}_{P_{1_x}}(t)$ and $\mathbf{r}_{P_{2_x}}(t)$ are the X-components of the vectors $\mathbf{r}_{P_1}(t)$ and $\mathbf{r}_{P_2}(t)$ respectively. Note that when $\theta(t) = 0$ and the body makes contact with the terrain, the points P_1 and P_2 touch the terrain. However, when $\theta(t) < 0$ and $\mathbf{r}_{P_{1_x}}(t) = X_{terr}$, the body can only make contact with the terrain through point P_1. Similarly, when $\theta(t) > 0$ and $\mathbf{r}_{P_{2_x}}(t) = X_{terr}$, the body can only make contact with the terrain through point P_2. This describes the choice of the shape of the robot's body to ease the process of deriving contact mechanics. The penalty function $F_{N_{P_{1_x}}}(t)$ is the X-component of the normal force vector $\mathbf{F}_{N_{P_1}} = (F_{N_{P_{1_x}}}, 0)$ acting on point P_1. Similarly, the penalty function $F_{N_{P_{2_x}}}(t)$ is the x-component of the normal force vector $\mathbf{F}_{N_{P_2}} = (F_{N_{P_{2_x}}}, 0)$ acting on point P_2. The generalized normal force vector is

$$\mathbf{Q}_{N_{body}} = \mathbf{F}_{N_{P_1}} \cdot \frac{\partial \mathbf{r}_{P_1}}{\partial \mathbf{q}_{body}} + \mathbf{F}_{N_{P_2}} \cdot \frac{\partial \mathbf{r}_{P_2}}{\partial \mathbf{q}_{body}} \tag{3.53}$$

In total, the vector of generalized forces for the robot's body is given by

$$\mathbf{Q}_{body} = \mathbf{Q}_{D_{body}} + \mathbf{Q}_{S_{body}} + \mathbf{Q}_{N_{body}} \tag{3.54}$$

where $\mathbf{Q}_{D_{body}}$ and $\mathbf{Q}_{S_{body}}$ are defined in Equation (3.48) and Equation (3.50) respectively.

3.4.5 Euler-Lagrange Equations

The Lagrangian for the robot's body is given by

$$\mathcal{L}_{body} = \mathcal{K}_{body} - \mathcal{E}_{body} \tag{3.55}$$

The Euler-Lagrange equations describing the dynamics of the robot's body are given by

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_{body}}{\partial \dot{\mathbf{q}}_{body}}\right) - \frac{\partial \mathcal{L}_{body}}{\partial \mathbf{q}_{body}} = \mathbf{Q}_{body} \tag{3.56}$$

which results in three coupled ordinary differential equations describing the time evolution of the quantities $X_{body}(t)$, $Y_{body}(t)$, and $\theta(t)$.

3.5 Robot Dynamics

The equations of motion of the robot are described using

$$\rho \mathbf{M}_i \ddot{\mathbf{q}}_i + \zeta \mathbf{M}_i \dot{\mathbf{q}}_i + \mathbf{N}_i \mathbf{q}_i + \mathbf{L}_i - \mathbf{U}_i = 0 \tag{3.57a}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_{body}}{\partial \dot{\mathbf{q}}_{body}}\right) - \frac{\partial \mathcal{L}_{body}}{\partial \mathbf{q}_{body}} = \mathbf{Q}_{body} \tag{3.57b}$$

In total, $4nN + 3$ ordinary differential equations are required to describe the dynamics of the robot.

Chapter 4

Simulation Results and Discussion

To demonstrate the features of the robot locomotion model presented above, a simulation of a nine-legged robot is conducted, with the equations of motion generated and solved using a code written in Mathematica 13. Each leg's finite element is discretized by five uniformly spaced nodes in the range $[0, \ell]$. The resulting system of differential equations, along with the equations of motion for the robot's body, is solved using the Mathematica built-in solver NDSolve, which utilizes multi-step methods with local convergence control and adaptive time steps to solve a wide range of ordinary and partial differential equations. The "AccuracyGoal" and "PrecisionGoal" options of NDSolve are used to control the absolute and relative local errors in the solution, with the solution being automatically halted if it does not converge to the specified tolerances. By adjusting these control parameters, solutions with varying levels of error and simulation run-time can be obtained while ensuring convergence. In this thesis, both parameters are set to 3 digits to significantly reduce the time required to solve the equations while maintaining satisfactory solution errors. Using these parameters, it took Mathematica approximately 11 hours to solve the system of 183 coupled ordinary differential equations that describe the nine-legged robot's dynamics over a 50-second simulation. The long computational time could be attributed to the symbolic computational feature of Mathematica. The simulation is executed without any simplification of the differential equations presented here. Instead, the symbolic computation of Mathematica is used to evaluate most expressions, which are long and complex to evaluate by hand.

The initial condition of the robotic system is depicted in Figure 4.1, where contiguous legs are set to be phase-shifted by 90 degrees. The initial velocity equals zero, and in the given configuration, three legs are in contact with the terrain.

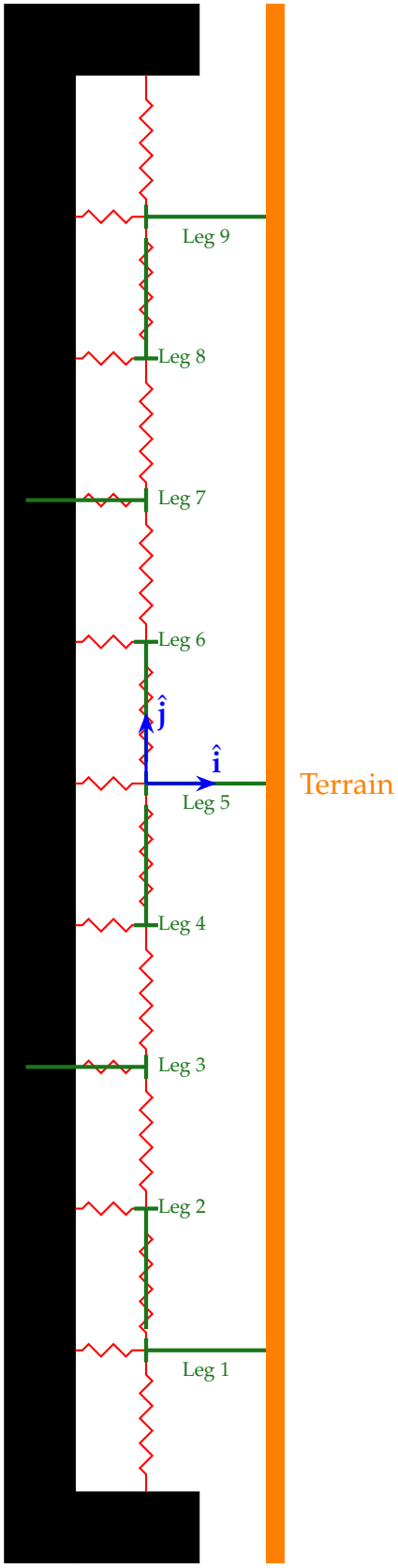


Figure 4.1: Schematic of the initial conditions. The legs are depicted in green

The initial configurations of the filaments in terms of the arclength are given by

$$\mathbf{r}_1(s,0) = s\hat{\mathbf{i}} - 4\hat{\mathbf{j}} \tag{4.1a}$$

$$\mathbf{r}_2(s,0) = (-3-s)\hat{\mathbf{j}} \tag{4.1b}$$

$$\mathbf{r}_3(s,0) = -s\hat{\mathbf{i}} - 2\hat{\mathbf{j}} \tag{4.1c}$$

$$\mathbf{r}_4(s,0) = (s-1)\hat{\mathbf{j}} \tag{4.1d}$$

$$\mathbf{r}_5(s,0) = s\hat{\mathbf{i}} \tag{4.1e}$$

$$\mathbf{r}_6(s,0) = (1-s)\hat{\mathbf{j}} \tag{4.1f}$$

$$\mathbf{r}_7(s,0) = -s\hat{\mathbf{i}} + 2\hat{\mathbf{j}} \tag{4.1g}$$

$$\mathbf{r}_8(s,0) = (s+3)\hat{\mathbf{j}} \tag{4.1h}$$

$$\mathbf{r}_9(s,0) = s\hat{\mathbf{i}} + 4\hat{\mathbf{j}} \tag{4.1i}$$

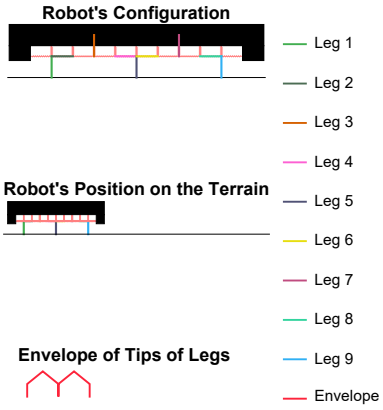
To sustain metachronal coordination and enable robot movement, all of its legs are driven simultaneously at the same angular velocity. The internal forces of the filaments are set to zero in the simulation to demonstrate metachronal coordination through the application of moments at the bases. The robot's parameters were tuned to obtain stable results and are provided in Table 4.1. Using these parameters, the initial coordinates of the center of mass of the robot's body are $(X_{body}, Y_{body}) \approx (-0.9112, 0)$ which can be calculated using Equation (3.2).

The filaments are set to have high bending rigidity to provide support for the body and to facilitate propulsion by enabling the legs to push against the terrain. Additionally, the springs that connect the legs to the robot's body are set to be stiff to ensure steady movement of the bases of the legs and the maintenance of a consistent gait cycle. If the springs were more flexible, the base of the legs oscillate more frequently with high displacements from their initial positions, disrupting the robot's gait cycle. However, highly flexible springs can enhance the transfer of forces between oscillating legs through reactive spring forces. This can improve the energy efficiency of locomotion by reducing the moment required to oscillate the legs. Future research will focus on the design of the basal coupling mechanics to achieve and sustain desired coordination patterns with respect to walking-type locomotion.

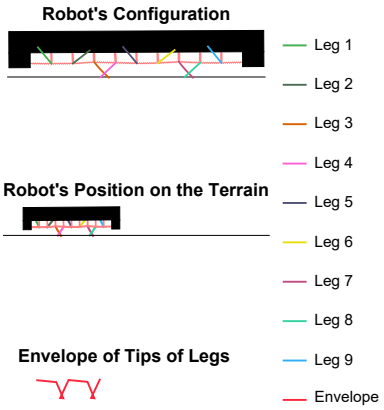
Parameter	Symbol	Value	Units
Length of each leg	ℓ	1	m
Distance between filaments	a	0.05	m
Linear mass density of each leg	ρ	1	kgm^{-1}
Bending rigidity of each leg	K_M	50	kgm^3s^{-2}
Mass of the robot's body	m	0.01	kg
Acceleration due to gravity	g	9.81	ms^{-2}
Damping coefficient	ζ	5.0	Nsm^{-1}
Integration time	tf	50	s
Number of legs	N	9	–
Number of nodes	n	5	–
Spring constants of $S_{\alpha_1}, \ldots, S_{\alpha_{10}}$	α	500	Nm^{-1}
Initial lengths of springs $S_{\alpha_1}, \ldots, S_{\alpha_{10}}$	L_α	1	m
Spring constants of $S_{\beta_1}, \ldots, S_{\beta_9}$	β	500	Nm^{-1}
Initial lengths of springs $S_{\beta_1}, \ldots, S_{\beta_9}$	L_β	0.5	m
Thickness of the robot's body	τ_T	1	m
Clearance of the robot's body	τ_C	0.2	m
Parallel distance between the basis vector $\hat{j}$ and the terrain	X_{terr}	1	m
Frequency of rotation of the legs	ω	0.5	s^{-1}
Proportional gain for the moment applied at the base of the legs	K_p	500	–
Penalty constant of the penalty function on the normal force	ϵ	1000	–
Penalty constant of the penalty function on the frictional force	ν	200	–
Penalty constant of the penalty system enforcing the inextensibility constraint on the filaments	λ	10000	–
Natural frequency of the penalty system enforcing the inextensibility constraint on the filaments	η	175	–
Damping ratio of the penalty system enforcing the inextensibility constraint on the filaments	μ	50	–

Table 4.1: Parameters for the simulation of the proposed robot model

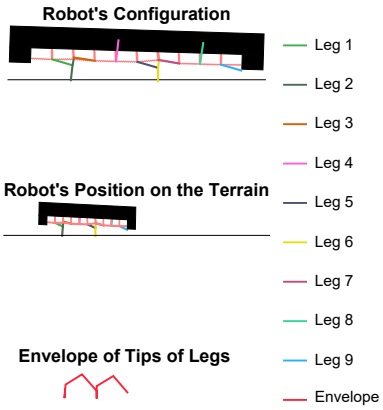
Appendix B presents the Mathematica code utilized for simulating the robot, while Figure 4.2 displays snapshots of the robot's configuration at different time instants during the simulation. The simulation results demonstrate the feature of moving across a flat terrain, made possible by the coordinated motion of the legs, which maintain a 90° phase shift to propel the robot forward, with three contact points with the terrain.



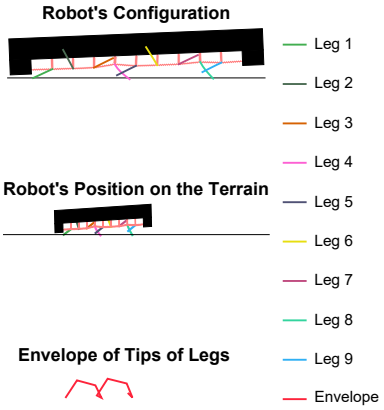
(a) t = 0 s



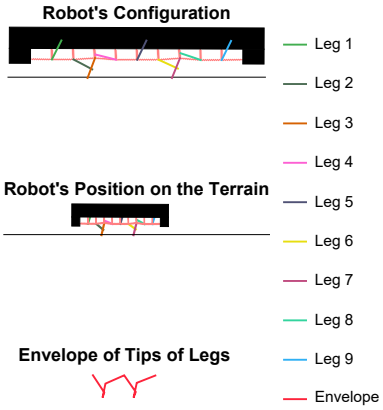
(b) t = 5 s



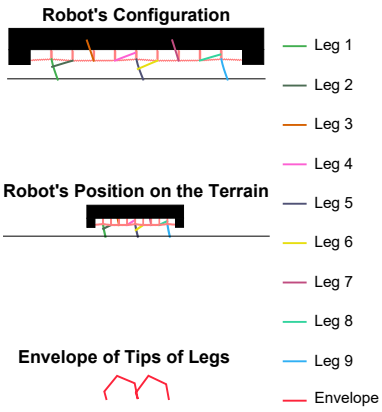
(c) t = 10 s



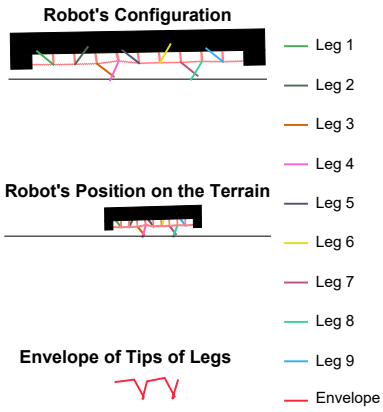
(d) t = 15 s



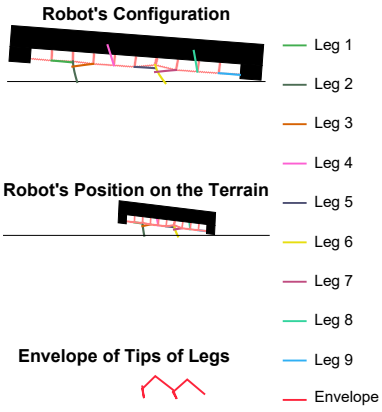
(e) t = 20 s



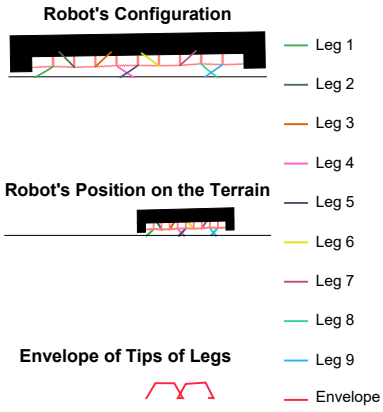
(f) t = 25 s



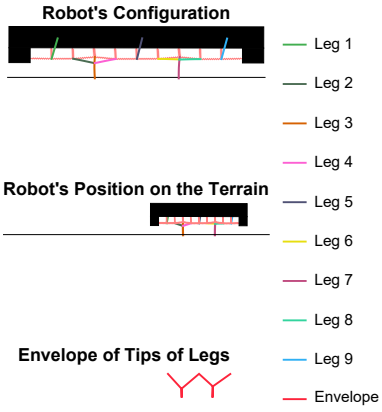
(g) t = 30 s



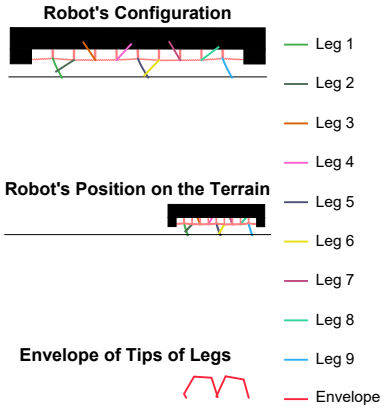
(h) t = 35 s



(i) t = 40 s



(j) t = 45 s



(k) t = 50 s

Figure 4.2: Snapshots of the animation of the robot at different time instants

In Figure 4.2, eleven snapshots are presented, each taken at five-second intervals, depicting the position on the path, the configuration, and the envelope of the tips of the legs. The envelope is represented by joining their tips with straight lines. The forward motion of the robot on the terrain can be inferred by plotting the X and Y coordinates of the body's center of mass, shown in Figure 4.3. The Y coordinate of the center of mass steadily increases, indicating the forward motion. In contrast, the X coordinate remains almost constant with small oscillations during the simulation, suggesting that the body is bobbing during the forward motion while maintaining a close vicinity to the terrain. The reactive spring forces generate the oscillations during traversal. Also, the forward motion indicates that the penalty functions in the form of normal forces, and frictional forces act as expected on the tips of the legs.

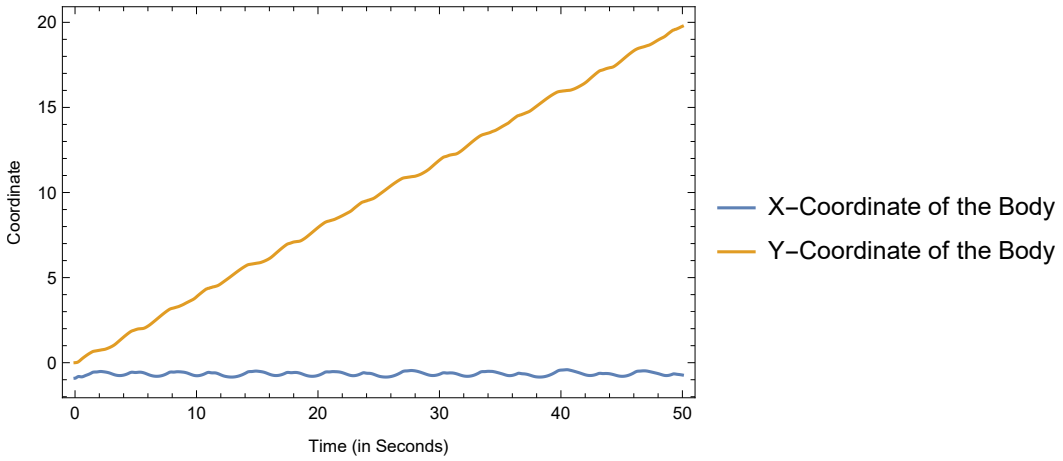


Figure 4.3: Evolution of the coordinates of the center of mass of the robot's body

In order to visualize the metachronal coordination, the basal angles of each leg are plotted in Figure 4.4. The results show that the applied moments at the bases of the legs ensure the phase difference between contiguous legs to be 90°. Another visualization of metachronal coordination is the envelope of the tips of the legs. As shown in Figure 4.2, the envelope follows a forward propagating sinusoidal wave. The resolution of this sinusoidal wave can be enhanced by utilizing more legs for robot locomotion and increasing the distance between the base of the legs beyond the length of the legs.

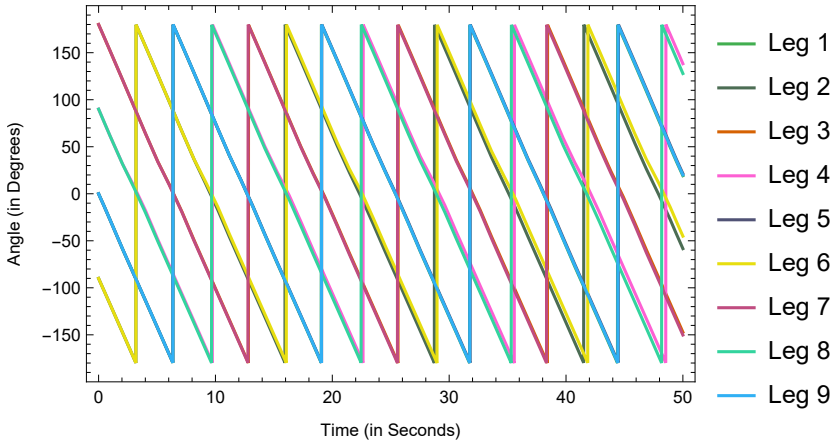
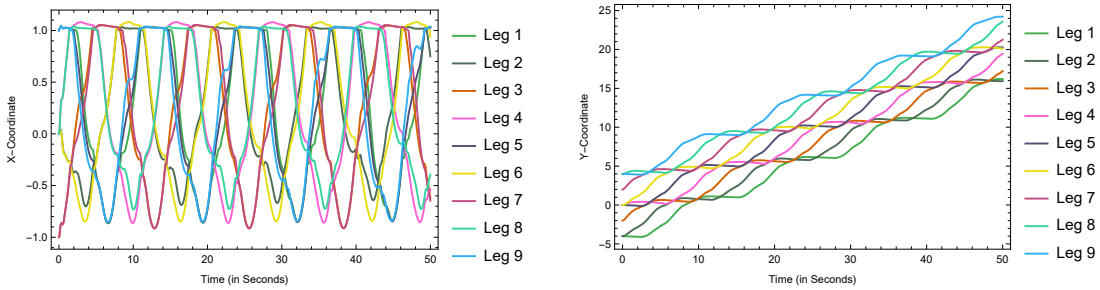


Figure 4.4: Evolution of the basal angles of the legs

The evolution of the coordinates of the tips of the filaments $X_i(\ell, t)$ and $Y_i(\ell, t)$ is shown in Figure 4.5, which exhibit an oscillatory behavior induced by the rotation of the filaments. Furthermore, the periodic nature of this oscillatory behavior can be attributed to the metachronal coordination between filaments.



(a) X-coordinate of the tips

(b) Y-coordinate of the tips

Figure 4.5: Evolution of the tip displacements of the legs

It can also be observed from Figure 4.2 that the legs bend when pushing against the terrain during the locomotion process due to their flexibility. The penalty function enforces the inextensibility constraint, forcing the legs to bend instead of changing their length when an external force is applied. The inextensibility condition $\mathcal{V}_i = \int_0^\ell (\mathbf{r}_i'.\mathbf{r}_i' - 1) ds_i = \int_0^\ell (X_i'^2 + Y_i'^2 - 1) ds_i$ is plotted for all the filaments during the robot locomotion in Figure 4.6.

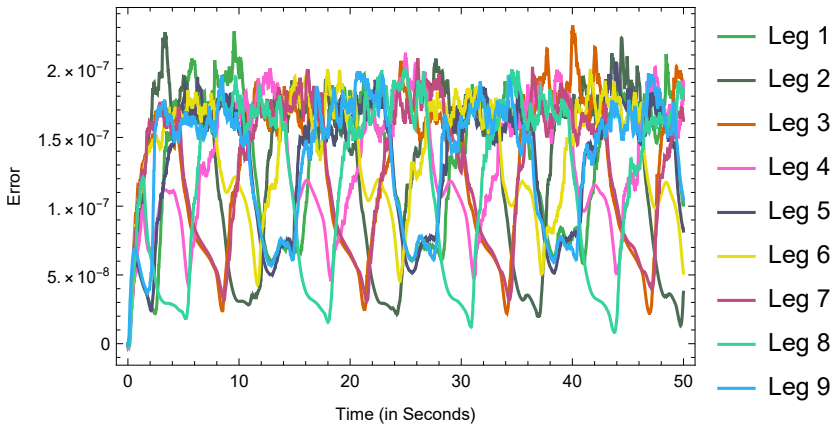


Figure 4.6: Evolution of the inextensibility error of the filaments

In the simulation, the error is less than 10^{-7} for all the legs during the locomotion. Compared to the length of the stiff filaments, which is one, the error is small, indicating the effectiveness of the penalty function in enforcing the inextensibility constraint on the legs. The Lagrange multiplier approach might be better than the penalty function method at enforcing the inextensibility constraint since it does not require additional parameter tuning. However, for the simulations presented here, the approximated penalty method is sufficient to illustrate the dynamics of the model.

In this thesis, mesh sensitivity analysis is not conducted for the modeled robotic system, since the primary focus is to provide results that describe the performance, prove the viability of the proposed locomotion mechanism, and generating a stable solution with respect to locomotion. Though highly resource-costly, mesh refinement could be necessary in future work to obtain grid-converged solutions and study certain bending deformation factors of individual legs. However, for this work, it is expected that the solutions will be within certain acceptable bounds, and the qualitative aspects of the provided results will not vary with mesh refinement.

Chapter 5

Conclusions and Future Work

Terrestrial robots have a wide spectrum of applications, ranging from non-destructive environmental inspection to military and defense applications. New locomotion mechanisms are required to be researched in order to improve performance of terrestrial robots and to use them in more complex environments. Improved robot performance can accommodate for navigation in unknown environments, locomotion over rough and uneven terrains and optimization of power consumption. This thesis contributes towards the development of a multi-legged locomotion strategy with potential benefits in the form of terrain adaptive motion and lower power consumption when compared to rigid multi-legged robots.

The biological inspiration for the locomotion mechanism comes from the mobility principle used by microorganisms to travel in fluid environments. These small-scale entities utilize flexible protrusions called axonemes present on their body to push the surrounding fluid via internal electrochemical motors and move in a desirable direction. While the protrusions are actuated, coordinated beating patterns emerge due to a combination of mechanical coupling through the bases of the axonemes, and hydrodynamic interaction between the flowing fluid and the axonemes. The mechanical linkage between the bases of the protrusions via flexible inter-cellular connections is also referred to as basal coupling. One beating pattern of interest is metachronal coordination which requires synchronous oscillation of all the filaments with a constant phase shift between contiguous ones. This suggests the hypothesis that metachronal coordination can be achieved by pure basal coupling, as it can also be observed in terrestrial animals. For instance, millipedes possess hundreds of legs and utilize metachronal coordination to move. Hence, the fluid based locomotion principle utilized by microorganisms could also be utilized for terrestrial locomotion of robots by assuming the legs to be an array

of flexible filaments coupled at the base. From a design perspective, an array of flexible protrusions is a robust solution as they can adapt and morph to a variety of terrains enabling locomotion of a multi-legged robot over unstructured terrains. Furthermore, metachronal coordination via basal coupling could minimize the actuation input to each leg and maximize the energy efficiency of the robot.

To illustrate the proposed locomotion solution, this thesis presents the mathematical modeling and simulation of a two-dimensional model of a robot, which utilizes flexible filaments as its legs, that are coupled at the base using linear springs. The robot's body has a U-shape with uniform thickness that encloses the bases of the legs of the robot. Each leg is modeled as a two-dimensional non-linear rod with internal forces by using constrained Euler elastica. The constraint prevents the axial elongation while allowing to bend. The constraint is enforced using a penalty based approach. The modeling results in a system of nonlinear partial differential equations which are solved using the finite element method. The robot's body is assumed to be rigid to simplify the dynamics, and focus on the impact of utilizing flexible filaments as legs. The dynamics of the body is derived through Euler-Lagrange equations. The robot's movement is induced by actuating each leg with a moment at the base. The moments applied at the bases of the legs ensure a desirable phase shift between contiguous legs and sustain metachronal coordination. The robot is simulated over a flat and even terrain to illustrate locomotion. The contact forces between the terrain and the legs in the form of normal and frictional forces are also included in the model. The internal forces on the legs are set to zero for the purpose of the simulation. Furthermore, each is initially assumed to be out of phase by a desired phase difference, to start from metachronal coordination. The finite element space of each filament is discretized by five uniformly spaced nodes. Simulation results illustrate the motion across a flat terrain through the use of metachronal coordination. The contact forces in the form of normal and frictional forces generated by the interaction between the robot and the terrain enable the forward movement of the robot.

In this thesis, metachronal coordination is not an emerging pattern from basal coupling. The understanding of the connection between legs' elastic coupling and eventual emergence of coordinated stable states is the object of ongoing work, with implications for robot design. The topology of the coupling basal elements could play an important role in inducing coordinated patterns. Current research directions are exploring ideas from pinning controllability of networked systems, to obtain existence and stability conditions of coordinated states, and eventual

minimal requirements in terms of actuation.

The results presented indicate that the proposed model is capable of describing legged terrestrial locomotion on a flat terrain, with sustained metachronal coordination of the system of legs. This thesis is the first step. Future work includes

- **Minimizing actuation inputs to the legs to improve overall energy efficiency of the robot**: In this thesis, stiff linear springs were used as basal coupling elements to ensure smooth operation of the robot. The actuation inputs can be minimized by using better mechanical coupling elements such as nonlinear springs, or a combination of springs and dampers that induce metachronal wave patterns. These mechanical coupling elements must enable transfer of forces among legs to induce metachronal coordination.

- **Utilizing internal forces of the filaments for actuation**: In this thesis, the internal forces of the filaments were set to zero to depict how the robot moves when only moments are applied at the bases of the legs. The internal forces could be used as actuation, consistently with the mechanics of several organisms.

- **Testing the robot on unstructured terrains**: This thesis only demonstrates the capabilities and potential over a flat terrain. Since the legs are flexible, the legs of the robot can adapt and morph to a variety of terrain. Hence, the proposed robot model is an attractive solution to deploy over unstructured environments.

- **Flexibility of the robot's body**: In this thesis, the robot's body is considered to be rigid to simplify the mathematical modeling process. By considering a highly flexible body, the locomotion could be aided [112]. Further, the flexibility of the body could be a requirement for specific missions.

Appendix A

Calculations, Simplifications and Important Plots

A.1 Variation of Kinetic Energy and Potential Energy

The variation of the kinetic energy of the i^{th} axoneme yields

$$\begin{aligned}\delta\mathcal{K}_i &= \delta\left(\frac{1}{2}\int_0^\ell \rho\dot{\mathbf{r}}_i\cdot\dot{\mathbf{r}}_i ds_i\right)\\ &= \int_0^\ell \rho\dot{\mathbf{r}}_i\cdot\delta\dot{\mathbf{r}}_i ds_i\\ &= \int_0^\ell \rho\dot{\mathbf{r}}_i\cdot\frac{d}{dt}\left(\delta\mathbf{r}_i\right) ds_i\end{aligned} \tag{A.1}$$

Integrating the variation of the kinetic energy between the time instants t_1 and t_2, applying integration by parts, and using the condition $\delta\mathbf{r}_i(t_1) = \delta\mathbf{r}_i(t_2) = 0$ from Hamilton's principle, we get

$$\begin{aligned}\int_{t_1}^{t_2}\mathcal{K}_i dt &= \int_0^\ell \rho\dot{\mathbf{r}}_i\cdot\delta\mathbf{r}_i\Big|_{t_1}^{t_2} ds_i - \int_{t_1}^{t_2}\left(\int_0^\ell \rho\ddot{\mathbf{r}}_i\cdot\delta\mathbf{r}_i ds_i\right)dt\\ &= -\int_{t_1}^{t_2}\left(\int_0^\ell \rho\ddot{\mathbf{r}}_i\cdot\delta\mathbf{r}_i ds_i\right)dt\end{aligned} \tag{A.2}$$

Since $\delta\kappa_i = \mathbf{n}_i\cdot\delta\mathbf{r}_i''$, the variation of the potential energy of the i^{th} axoneme yields

$$\begin{aligned}
\delta\mathcal{E}_i &= \delta\left(\int_0^\ell \left(\frac{K_M}{2}\kappa_i^2 - aF_i\kappa_i + \rho g(X_{terr} - \mathbf{r}_i\cdot\hat{\mathbf{i}})\right) ds_i\right) \\
&= \int_0^\ell \left(\left(K_M\kappa_i - aF_i\right)\delta\kappa_i - \rho g\hat{\mathbf{i}}\cdot\delta\mathbf{r}_i\right) ds_i \\
&= \int_0^\ell \left(\left(K_M\|\mathbf{r}_i''\| - aF_i\right)\frac{\mathbf{r}_i''}{\|\mathbf{r}_i''\|}\cdot\delta\mathbf{r}_i'' - \rho g\hat{\mathbf{i}}\cdot\delta\mathbf{r}_i\right) ds_i \\
&= \int_0^\ell \left(\left(K_M\mathbf{r}_i'' - aF_i\frac{\mathbf{r}_i''}{\|\mathbf{r}_i''\|}\right)\cdot\delta\mathbf{r}_i'' - \rho g\hat{\mathbf{i}}\cdot\delta\mathbf{r}_i\right) ds_i
\end{aligned} \tag{A.3}$$

Applying integration by parts twice yields

$$\begin{aligned}
\delta\mathcal{E}_i = &\int_0^\ell \left(K_M\mathbf{r}_i^{\mathrm{IV}} - \left(\frac{aF_i}{\|\mathbf{r}_i''\|}\mathbf{r}_i''\right)'' - \rho g\hat{\mathbf{i}}\right)\cdot\delta\mathbf{r}_i ds_i \\
&+ \left(K_M - \frac{aF_i}{\|\mathbf{r}_i''\|}\right)\mathbf{r}_i''\cdot\delta\mathbf{r}_i'\Big|_0^\ell - \left(K_M\mathbf{r}_i^{\mathrm{III}} - \left(\frac{aF_i}{\|\mathbf{r}_i''\|}\mathbf{r}_i''\right)'\right)\cdot\delta\mathbf{r}_i\Big|_0^\ell
\end{aligned} \tag{A.4}$$

A.2 Calculations for the Weak Form

Multiplying the partial differential equation 3.30 with the virtual displacement $\delta\mathbf{r}_i$ and integrating it over the spatial domain $[0, \ell]$ gives

$$\int_0^\ell \left(\left(\rho\ddot{\mathbf{r}}_i + \zeta\dot{\mathbf{r}}_i + K_M\mathbf{r}_i^{\mathrm{IV}} - \left(\frac{aF_i}{\|\mathbf{r}_i''\|}\mathbf{r}_i''\right)'' - \rho g\hat{\mathbf{i}}\right)\cdot\delta\mathbf{r}_i\right) ds_i = 0 \tag{A.5}$$

Applying integration by parts twice, we get

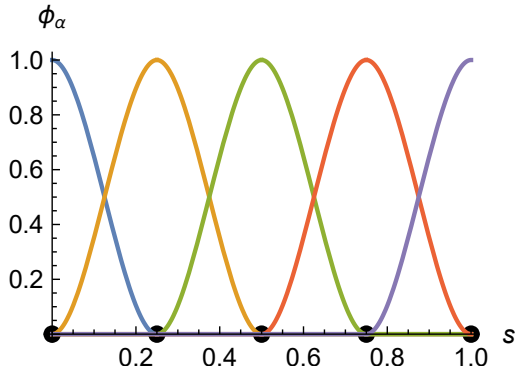
$$\begin{aligned}
\int_0^\ell &\left(\left(\rho\ddot{\mathbf{r}}_i + \zeta\dot{\mathbf{r}}_i - \rho g\hat{\mathbf{i}}\right)\cdot\delta\mathbf{r}_i + \left(K_M - \frac{aF_i}{\|\mathbf{r}_i''\|}\right)\mathbf{r}_i''\cdot\delta\mathbf{r}_i''\right) ds_i \\
&- \left(K_M - \frac{aF_i}{\|\mathbf{r}_i''\|}\right)\mathbf{r}_i''\cdot\delta\mathbf{r}_i'\Big|_0^\ell + \left(K_M\mathbf{r}_i^{\mathrm{III}} - \left(\frac{aF_i}{\|\mathbf{r}_i''\|}\mathbf{r}_i''\right)'\right)\cdot\delta\mathbf{r}_i\Big|_0^\ell
\end{aligned} \tag{A.6}$$

Substituting the boundary conditions 3.32, we get

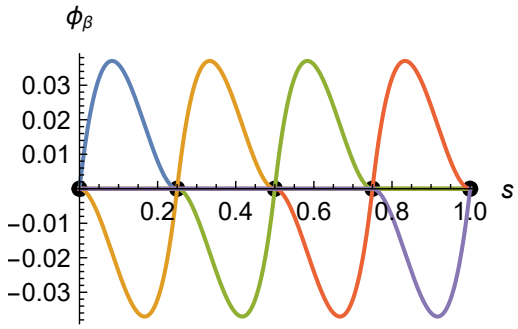
$$\int_0^{\ell} \left(\left(\rho \ddot{\mathbf{r}}_i + \zeta \dot{\mathbf{r}}_i - \rho g \hat{\mathbf{i}}\right) \cdot \delta \mathbf{r}_i + \left(K_M - \frac{aF_i}{\|\mathbf{r}_i''\|} \right) \mathbf{r}_i'' \cdot \delta \mathbf{r}_i'' \right) ds_i \\ - \mathbf{F}_{\ell_i} \cdot \delta \mathbf{r}_i(\ell, t) - \mathbf{F}_{S_i} \cdot \delta \mathbf{r}_i(0, t) - M_{0_i} \frac{\mathbf{r}_i''(0,t)}{\|\mathbf{r}_i''(0,t)\|} \cdot \delta \mathbf{r}_i'(0, t) = 0 \tag{A.7}$$

A.3 Sample Plots of Finite Element Basis Functions

For a mesh of five uniformly spaced nodes in the domain $[0, \ell]$, the finite element basis functions $\boldsymbol{\phi}_\alpha$ and $\boldsymbol{\phi}_\beta$ and their derivatives are constructed using the conditions shown in Equation (3.37). The following are the plots

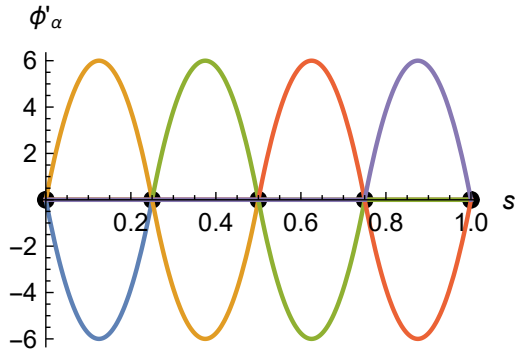


(a) Plot of $\boldsymbol{\phi}_\alpha$

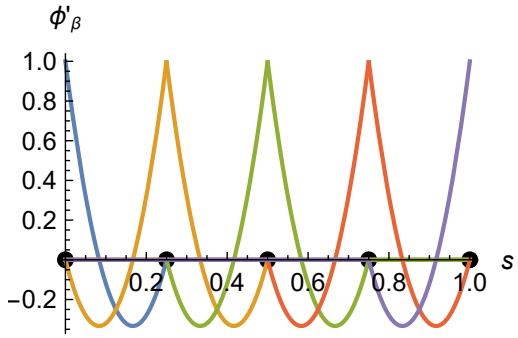


(b) Plot of $\boldsymbol{\phi}_\beta$

Figure A.1: Plots of finite element basis functions $\boldsymbol{\phi}_\alpha$ and $\boldsymbol{\phi}_\beta$ when $n = 5$



(a) Plot of $\boldsymbol{\phi}'_\alpha$



(b) Plot of $\boldsymbol{\phi}'_\beta$

Figure A.2: Plots of the derivatives of the finite element basis functions $\boldsymbol{\phi}'_\alpha$ and $\boldsymbol{\phi}'_\beta$ when $n = 5$

A.4 Structures of Matrices Appearing in the Finite Element Discrete Form

The following are the elements of different matrices appearing in the finite element discrete form:

- Mass matrix $\mathbf{M}_i$:

$$\mathbf{M}_i = \begin{bmatrix} \int_0^\ell \left(\boldsymbol{\phi}^T \boldsymbol{\phi}\right) ds_i & \mathbf{0} \\ \mathbf{0} & \int_0^\ell \left(\boldsymbol{\phi}^T \boldsymbol{\phi}\right) ds_i \end{bmatrix} \tag{A.8}$$

- Nonlinear matrix $\mathbf{N}_i$:

$$\mathbf{N}_i = \begin{bmatrix} \int_0^\ell \left(\mathcal{Q}_i \boldsymbol{\phi}''^T \boldsymbol{\phi}''\right) ds_i & \mathbf{0} \\ \mathbf{0} & \int_0^\ell \left(\mathcal{Q}_i \boldsymbol{\phi}''^T \boldsymbol{\phi}''\right) ds_i \end{bmatrix} \tag{A.9}$$

  where

$$\mathcal{Q}_i = K_M - \frac{aF_i}{\sqrt{X_i''^2 + Y_i''^2}} = K_M - \frac{aF_i}{\sqrt{(\boldsymbol{\phi}'' \mathbf{q}_{X_i}^T)^2 + (\boldsymbol{\phi}'' \mathbf{q}_{Y_i}^T)^2}} \tag{A.10}$$

- Input vector $\mathbf{U}_i$:

$$\begin{aligned} \mathbf{U}_i = & \begin{bmatrix} F_{\ell_{x_i}}(t) \boldsymbol{\phi}^T(\ell) \\ F_{\ell_{y_i}}(t) \boldsymbol{\phi}^T(\ell) \end{bmatrix} + \begin{bmatrix} F_{S_{x_i}}(t) \boldsymbol{\phi}^T(0) \\ F_{S_{y_i}}(t) \boldsymbol{\phi}^T(0) \end{bmatrix} \\ & + M_{0_i}(t) \begin{bmatrix} -\boldsymbol{\phi}'^T(0).\boldsymbol{\phi}'(0) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\phi}'^T(0).\boldsymbol{\phi}'(0) \end{bmatrix} \begin{bmatrix} \mathbf{q}_{Y_i} \\ \mathbf{q}_{X_i} \end{bmatrix} \\ & + \rho g \begin{bmatrix} \int_0^\ell \boldsymbol{\phi}^T ds_i \\ \mathbf{0} \end{bmatrix} \end{aligned} \tag{A.11}$$

Appendix B

Robot Simulation Code in Mathematica

The code below is a script written in Wolfram Language and implemented in Mathematica 13. The code simulates the proposed robot model and demonstrates its locomotion capability over a flat terrain. The following source code can also be downloaded at the link to the Mathematica notebook.

Common Material, Geometric, and Time Integration Parameters

```
(*Length of each leg*)L = 1;
(*Distance between filaments*)a = L / 20.;
(*Density of each leg*)ρ = 1;
(*Bending rigidity of each leg*)kM = 50;
(*Number of legs*)nf = 9;
(*Mass of the robot's body*)m = 0.01;
(*Acceleration due to gravity*)g = 9.81;
(*Damping coefficient*)ζ = 5.0;
(*Integration Time*)tf = 50;
```

Finite Element Basis Functions

Finite Element Mesh Properties

```
(*Number of nodes*)Nn = 5;
(*Number of elements*)Ne = Nn - 1;
(*Number of nodes for each element*)Nm = 2;
(*Global nodes*)n = Table[i, {i, 0, L, L / (Nn - 1)}] // N;
```

Coefficients of the Interpolating Polynomials

```
(*Direct map*)x[s_, i_] := (n[[i]] + n[[i + 1]] + (n[[i + 1]] - n[[i]]) s) / 2;
(*Inverse map*)s[x_] = Table[s /. First[Solve[x[s, i] == x, s]], {i, Nn - 1}];
(*Polynomials*)NN[i_, x_] := α0 + Sum[αj (x - n[[i]])^j, {j, 3}];
(*Calculate coefficients of the interpolating polynomials*)
cond10 = {NN[1, x[-1, 1]] == 1, NN[1, x[1, 1]] == 0,
    (∂s NN[1, x[s, 1]] /. s → -1) == 0, (∂s NN[1, x[s, 1]] /. s → 1) == 0} // Chop;
sol10 = Solve[cond10, Table[αi, {i, 0, 3}]] // Flatten;
cond20 = {NN[1, x[-1, 1]] == 0, NN[1, x[1, 1]] == 1,
   (∂s NN[1, x[s, 1]] /. s → -1) == 0, (∂s NN[1, x[s, 1]] /. s → 1) == 0};
sol20 = Solve[cond20, Table[αi, {i, 0, 3}]] // Flatten;
cond11 = {NN[1, x[-1, 1]] == 0, NN[1, x[1, 1]] == 0,
   (∂s NN[1, x[s, 1]] /. s → -1) == 1, (∂s NN[1, x[s, 1]] /. s → 1) == 0};
sol11 = Solve[cond11, Table[αi, {i, 0, 3}]] // Flatten;
cond21 = {NN[1, x[-1, 1]] == 0, NN[1, x[1, 1]] == 0,
   (∂s NN[1, x[s, 1]] /. s → -1) == 0, (∂s NN[1, x[s, 1]] /. s → 1) == 1};
sol21 = Solve[cond21, Table[αi, {i, 0, 3}]] // Flatten;
sol = {{sol10, sol20}, {sol11, sol21}};

(*Polynomials with coefficients in the master element*)
polycoeff[a_, s_] := Flatten[Table[NN[1, x[s, 1]] /. sol[[i + 1, j]], {j, 2}, {i, 0, 1}]][[a]];
```

Basis Functions

```
(*Polynomials in the global domain*)
ϕ[x_] = Module[{ϕa, ϕb},
    ϕa = Join[{Which[n[[1]] ≤ x ≤ n[[2]], Evaluate[polycoeff[1, s[x][[1]]]], True, 0]},
      Table[Which[n[[i - 1]] ≤ x ≤ n[[i]], Evaluate[polycoeff[3, s[x][[i - 1]]]], Evaluate[
          n[[i]] ≤ x ≤ n[[i + 1]]], Evaluate[polycoeff[1, s[x][[i]]]], True, 0], {i, 2, Nn - 1}],
      {Which[n[[-2]] ≤ x ≤ n[[-1]], Evaluate[Expand[polycoeff[3, s[x][[Nn - 1]]]]], True, 0]}];
    ϕb = Join[{Which[n[[1]] ≤ x ≤ n[[2]], Evaluate[(n[[2]] - n[[1]])/2 polycoeff[2, s[x][[1]]]],
        True, 0]}, Table[Which[n[[i - 1]] ≤ x ≤ n[[i]],
        Evaluate[(n[[i]] - n[[i - 1]])/2 polycoeff[4, s[x][[i - 1]]]], Evaluate[n[[i]] ≤ x ≤ n[[i + 1]]],
        Evaluate[(n[[i + 1]] - n[[i]])/2 polycoeff[2, s[x][[i]]]], True, 0], {i, 2, Nn - 1}], {Which[
        n[[-2]] ≤ x ≤ n[[-1]], Evaluate[(n[[-1]] - n[[-2]])/2 polycoeff[4, s[x][[Nn - 1]]]], True, 0]}];
    Join[ϕa, ϕb]] // Chop;

(*Derivatives*)
Dϕ[x_] = D[ϕ[x], x];
DDϕ[x_] = D[ϕ[x], {x, 2}];
DDDϕ[x_] = D[ϕ[x], {x, 3}];

(*Plot of ϕα*)
figphialpha = Show[{Graphics[{PointSize[.04], Point[({n, Table[0, Nn]} // Transpose)]}],
    Plot[Table[ϕ[x][[i]], {i, 1, Nn}] // Evaluate, {x, 0, L},
     PlotStyle → Thickness[0.01], PlotRange → All]}, Axes → True, ImageSize → Small,
   AxesLabel → {"\!\(\*StyleBox[\"s\",FontSlant->\"Italic\"]\)", "ϕα"},
   AspectRatio → 1 / 1.5];(*Plot of ϕβ*)
```

```
figphibeta = Show[{Graphics[{PointSize[.04], Point[({n, Table[0, Nn]} // Transpose)]}],
    Plot[Table[ϕ[x][[i]], {i, Nn + 1, 2 Nn}] // Evaluate, {x, 0, L},
     PlotStyle → Thickness[0.01], PlotRange → All]}, Axes → True, ImageSize → Small,
   AxesLabel → {"\!\(\*StyleBox[\"s\",FontSlant->\"Italic\"]\)", "ϕβ"},
   AspectRatio → 1 / 1.5];
(*Plot of ϕα'*)
figphialphaprime =
  Show[{Graphics[{PointSize[.04], Point[({n, Table[0, Nn]} // Transpose)]}],
    Plot[Table[Dϕ[x][[i]], {i, 1, Nn}] // Evaluate, {x, 0, L},
     PlotStyle → Thickness[0.01], PlotRange → All]}, Axes → True, ImageSize → Small,
   AxesLabel → {"\!\(\*StyleBox[\"s\",FontSlant->\"Italic\"]\)", "ϕ'α"},
   AspectRatio → 1 / 1.5];
(*Plot of ϕβ'*)
figphibetaprime =
  Show[{Graphics[{PointSize[.04], Point[({n, Table[0, Nn]} // Transpose)]}],
    Plot[Table[Dϕ[x][[i]], {i, Nn + 1, 2 Nn}] // Evaluate, {x, 0, L},
     PlotStyle → Thickness[0.01], PlotRange → All]}, Axes → True, ImageSize → Small,
   AxesLabel → {"\!\(\*StyleBox[\"s\",FontSlant->\"Italic\"]\)", "ϕ'β"},
   AspectRatio → 1 / 1.5];
```

Robot Geometry and System Variables

Filament Parameters

Filament State Vectors

```
(*Collection of coefficients of vector a for all legs*)
qa[t_] =
  Table[ToExpression[StringJoin["a", ToString[i], ToString[j], "[t]"]], {i, nf}, {j, Nn}];
 (*Collection of coefficients of vector b for all legs*)
qb[t_] =
  Table[ToExpression[StringJoin["b", ToString[i], ToString[j], "[t]"]], {i, nf}, {j, Nn}];
(*Collection of coefficients of vector c for all legs*)
qc[t_] =
  Table[ToExpression[StringJoin["c", ToString[i], ToString[j], "[t]"]], {i, nf}, {j, Nn}];
(*Collection of coefficients of vector d for all legs*)
qd[t_] =
  Table[ToExpression[StringJoin["d", ToString[i], ToString[j], "[t]"]], {i, nf}, {j, Nn}];
(*Collection of all unknown coefficients*)
q[t_] = Join[qa[t], qb[t], qc[t], qd[t], 2];
(*Collection of all unknown coefficients of X(s,t)*)
qx[t_] = Join[qa[t], qb[t], 2];
(*Collection of all unknown coefficients of Y(s,t)*)
qy[t_] = Join[qc[t], qd[t], 2];
```

Filament Initial Conditions

```
(* r(s,0) i.e. the initial shape of each leg described using arclength parameter 's'*)
initialr[s_] = {{s, -4.0}, {0, -3.0 - s}, {-s, -2.0},
   {0, s - 1.0}, {s, 0}, {0, 1.0 - s}, {-s, 2.0}, {0, s + 3.0}, {s, 4.0}};
(* t(s,0) i.e. the unit tangent vector of
```

```
 each leg described using arclength parameter 's'*)
initialt[s_] = D[initialr[s], s];
```

Connecting Springs Parameters

```
(*Spring constants of Sα springs*)α = 500.0;
(*Undeformed Length of each Sα spring*)Lα = 1.0 ;
(*Spring constants of Sβ springs*)β = 500.0;
(*Undeformed Length of each spring*)Lβ = 0.5;
```

Robot Body Parameters

Generalized Coordinates and Dimensional Parameters

```
(*Generalized Coordinates of the robot's body*)qbody[t_] = {xbody[t], ybody[t], θ[t]};
(*Thickness of the robot's body*)tns = 1.0;
(*Clearance of the robot's body*)clc = 0.2;
```

Center of Mass Calculation

```
(*Length of the robot's body*)len = 2 * tns + (nf + 1) * Lα;
(*Breadth of the robot's body*)brd = tns + Lβ + clc;
(*Area of the robot's body*)area = (len) (tns) + 2 (brd - tns) (tns);
(*Initial X-coordinate of the center of mass of the robot's body*)
xcom = ((-Lβ - tns / 2) (len) (tns) + 2 ((-Lβ + clc) / 2) (brd - tns) (tns)) / area;
(*Initial Y-coordinate of the center of mass of the robot's body*)
ycom = 0;
```

Other Parameters

```
(*Rotation transformation function on the coordinates B1, B2, ...,
BN+2 for rotation by θ[t] about the point (xbody[t],ybody[t])*)
rtran = RotationTransform[qbody[t][[3]], {qbody[t][[1]], qbody[t][[2]]}];
(*Coordinates of points B1, B2, ..., BN+2*)
fpt[t_] = Join[{{qbody[t][[1]] - xcom, qbody[t][[2]] - ((nf + 1) / 2) Lα}},
   Table[{qbody[t][[1]] - xcom - Lβ, qbody[t][[2]] - (((nf + 1) / 2) - i) Lα}, {i, 1, (nf + 1) / 2}],
   Table[{qbody[t][[1]] - xcom - Lβ, qbody[t][[2]] + (i) Lα}, {i, 1, ((nf + 1) / 2) - 1}],
   {{qbody[t][[1]] - xcom, qbody[t][[2]] + ((nf + 1) / 2) Lα}}];
(*coordinates of fixed points after applying the transformation function*)
mfpt[t_] = rtran /@ fpt[t] // FullSimplify;

(*Penalization parameters for penalty function on in-extesibility constraint*)
λ = 10000;
η = 175;
μ = 50;
(*Penalization Parameter for penalty function on normal force*)ϵ = 1000;
(*Penalization Parameter for penalty function on friction force*)ν = 200;
(*Parallel distance between basis vector ĵ and the terrain*)Xterr = 1;
(*Proportional gain for the moment applied at the base of the axonemes*)Kp = 500;
(*Frequency of rotation of the legs*)ω = 0.5;
```

Projected Solutions and Evaluations

```
(*Approximate solutions for X(s,t),X'(s,t),X''(s,t)*)
Xa[x_, t_] = Map[ϕ[x].# &, qx[t]];
DXa[x_, t_] = Map[Dϕ[x].# &, qx[t]];
DDXa[x_, t_] = Map[DDϕ[x].# &, qx[t]];

(*Approximate solutions for Y(s,t),Y'(s,t),Y''(s,t)*)
Ya[x_, t_] = Map[ϕ[x].# &, qy[t]];
DYa[x_, t_] = Map[Dϕ[x].# &, qy[t]];
DDYa[x_, t_] = Map[DDϕ[x].# &, qy[t]];
```

Robot Initial Conditions

```
(* Initial conditions for unknown coefficients aij *)
ic1 = Table[Xa[n[[i]], 0][[j]], {j, nf}, {i, Length[n]}] -
      Table[initialr[n[[i]]][[ ;; , 1]][[j]], {j, nf}, {i, Nn}] // Chop // Flatten;
(* Initial conditions for unknown coefficients bij *)
ic2 = Table[DXa[n[[i]], 0][[j]], {j, nf}, {i, Length[n]}] -
      Table[initialt[n[[i]]][[ ;; , 1]][[j]], {j, nf}, {i, Nn}] // Chop // Flatten;
(* Initial conditions for unknown coefficients cij *)
ic3 = Table[Ya[n[[i]], 0][[j]], {j, nf}, {i, Length[n]}] -
      Table[initialr[n[[i]]][[ ;; , 2]][[j]], {j, nf}, {i, Nn}] // Chop // Flatten;
(* Initial conditions for unknown coefficients dij *)
ic4 = Table[DYa[n[[i]], 0][[j]], {j, nf}, {i, Length[n]}] -
      Table[initialt[n[[i]]][[ ;; , 2]][[j]], {j, nf}, {i, Nn}] // Chop // Flatten;
ic5 = q'[0] // Flatten;
(*Initial conditions for C.O.M of robot's body*)
ic6 = qbody[0] // Flatten;
ic7 = qbody'[0] // Flatten;

(*List of all initial conditions for the system*)
icvalues = Join[Thread[Join[ic1, ic2, ic3, ic4, ic5] == 0],
    Thread[ic6 == {xcom, ycom, 0}], Thread[ic7 == 0]] // Chop;

(*Threaded list of all initial conditions for the system*)
icthread =
  Join[Chop[Thread[Flatten[q[0]] → Flatten[Join[Table[initialr[n[[i]]][[ ;; , 1]][[j]],
          {j, nf}, {i, Nn}], Table[initialt[n[[i]]][[ ;; , 1]][[j]], {j, nf}, {i, Nn}],
        Table[initialr[n[[i]]][[ ;; , 2]][[j]], {j, nf}, {i, Nn}],
        Table[initialt[n[[i]]][[ ;; , 2]][[j]], {j, nf}, {i, Nn}], 2]]]],
   Chop[Thread[Flatten[q'[0]] → Flatten[ConstantArray[0, {nf, 4 Nn}]]]],
   Thread[qbody[0] → {xcom, ycom, 0}],
   Thread[qbody'[0] → 0]];
```

Robot Graphics Functions

```
(*Spring graphics function*)
(*n = number of coils, h = aspect ratio = width w.r.t unit length*)
unitSpringPoints[n_, h_] := Block[{dl, xlist, ylist}, dl = 1 / (2 n + 1);
  xlist = Flatten[{0, 1.5 dl, dl Table[k, {k, 2, 2 n - 1}], 1 - 1.5 dl, 1}];
  ylist = Flatten[{0, 0, h / 2 Table[(-1) ^ (k + 1), {k, 2, 2 n - 1}], 0, 0}];
  Transpose[{xlist, ylist}]]
```

```
(*Transformation matrix*)
coordinateTrasformMatrix[{x0_, y0_}, {x1_, y1_}] :=
 Block[{theta}, theta = ArcTan[x1 - x0, y1 - y0];
  {{Cos[theta], -Sin[theta]}, {Sin[theta], Cos[theta]}}]
(*Function to transform list of coordinates*)
coordinateTransform[coords_List, {{x0_, y0_}, {x1_, y1_}}] :=
 Block[{scale, mat}, scale = Sqrt[(x1 - x0)^2 + (y1 - y0)^2];
  mat = coordinateTrasformMatrix[{x0, y0}, {x1, y1}];
  ({x0, y0} + mat.({scale, 1} * #)) & /@ coords]
(*Spring function that take the end coordinates of
 spring as input along with no. of coils and aspect ratio*)
spring[{{x0_, y0_}, {x1_, y1_}}, n_ : 10, h_ : 0.05] :=
  Line[coordinateTransform[unitSpringPoints[n, h], {{x0, y0}, {x1, y1}}]];

(*Leg base graphics function*)
base[{x_, y_}, l_ : L / 2.5, ang_ : 0] :=
 Line[{{x - (l / 2) Sin[ang Degree], y + (l / 2) Cos[ang Degree]},
   {x + (l / 2) Sin[ang Degree], y - (l / 2) Cos[ang Degree]}}]

(*Coordinate basis vectors*)
basis[{x_, y_, thick_ : 0.015, len_ : 0.5}] := {Thickness[thick],
   Arrowheads[Small], Arrow[{{{x, y}, {x, y + len}}, {{x, y}, {x + len, y}}}],
   Text[Style["x", Medium], {x + len, y}, {-1, 0}, {1, 0}],
   Text[Style["y", Medium], {x, y + len}, {0, -1}, {1, 0}]};

(*Robot body graphics function*)
body[{x_, y_, ang_}] := Rotate[ {Rectangle[
     {x - xcom - Lβ - tns, y - ((nf + 1) / 2) Lα - tns}, {x - xcom - Lβ, y + ((nf + 1) / 2) Lα + tns}],
    Rectangle[{x - xcom - Lβ, y - ((nf + 1) / 2) Lα - tns}, {x - xcom + clc, y - ((nf + 1) / 2) Lα}],
    Rectangle[{x - xcom - Lβ, y + ((nf + 1) / 2) Lα},
     {x - xcom + clc, y + ((nf + 1) / 2) Lα + tns}]}, ang, {x, y}];
```

Equations of Motion of the Filaments

Internal Driving Force of the Filaments

```
(*The internal forces f of each filament*)f[x_] = Table[0., {k, nf}];
(*The function F(s)=-∫_s^' f(σ) dσ*)F[x_] := -Integrate[f[ξ], {ξ, x, L}];
```

Mass Matrix

```
(*The expression ∫_0^' (ϕ^T ϕ) ds_i*)
MX = Table[Integrate[ϕ[x][[i]] ϕ[x][[j]], {x, 0, L}], {i, 2 Nn}, {j, 2 Nn}];
(*Construction of the mass matrix*)
MM = Join[Join[MX, ConstantArray[0, {2 Nn, 2 Nn}], 2],
   Join[ConstantArray[0, {2 Nn, 2 Nn}], MX, 2]];
```

Nonlinear Matrix

```
(*Function for evaluating the function Q_i*)QQ[x_, t_] :=
  Table[kM - (a F[x][[i]] / Sqrt[(DDXa[x, t][[i]])^2 + (DDYa[x, t][[i]])^2]), {i, 1, nf}];
```

```
(*The expression ∫₀^ℓ (Qᵢϕ''^T ϕ'') dsᵢ*)
NX = Table[Integrate[QQ[x, t][[k]] × DDϕ[x][[i]] × DDϕ[x][[j]], {x, 0, L}],
   {k, nf}, {i, 2 Nn}, {j, 2 Nn}];
(*Construction of the nonlinear matrix*)
NM = Table[Join[Join[NX[[i]], ConstantArray[0, {2 Nn, 2 Nn}], 2],
    Join[ConstantArray[0, {2 Nn, 2 Nn}], NX[[i]], 2]], {i, nf}];
```

Penalty Vector

```
(*The expression ∫₀^ℓ (ϕ'^T ϕ') dsᵢ*)
VX = Table[Integrate[Dϕ[x][[i]] × Dϕ[x][[j]], {x, 0, L}], {i, 2 Nn}, {j, 2 Nn}];
(*Constraint Equation*)
VV = Table[q[t][[i]].Join[Join[VX, ConstantArray[0, {2 Nn, 2 Nn}], 2],
       Join[ConstantArray[0, {2 Nn, 2 Nn}], VX, 2]].q[t][[i]] - L, {i, nf}] // FullSimplify;
(*Construction of the penalty vector*)
LL = Table[λ (D[VV[[j]], {t, 2}] + 2 η μ D[VV[[j]], t] + (η)^2 VV[[j]])
    Table[D[VV[[j]], q[t][[j]][[i]]], {i, Length[q[t][[j]]]}], {j, nf}];
```

Input Vector

External Forces and Moments

```
(*Penalty function for the normal force*)
fLx[t_] = Table[Piecewise[{{ϵ (Xterr - q[t][[i, Nn]]), q[t][[i, Nn]] ≥ Xterr}}, 0], {i, nf}];
(*Penalty function for the frictional force*)
fLy[t_] = Table[Piecewise[{{ν (-q'[t][[i, 3 Nn]]), q[t][[i, Nn]] ≥ Xterr}}, 0], {i, nf}] ;
(*Moments applied at the base of the legs*)
M0[t_] = Table[Kp ( -ω - D[ArcTan[q[t][[i, Nn + 1]], q[t][[i, 3 Nn + 1]]], t]), {i, nf}];
(*Note that, Xᵢ(ℓ,t) = ϕ(ℓ).q_{xᵢ}(t) = a_in, Ẏᵢ(ℓ,t) = ϕ(ℓ).q̇_{yᵢ}(t) = ċ_in,
Xᵢ'(0,t) = ϕ'(0).q_x(t) = b_i1, Yᵢ'(0,t) = ϕ'(0).q_y(t) = d_i1*)

(*Computataion of the vector of external forces*)
fext = Table[Join[fLx[t][[i]] ϕ[L], fLy[t][[i]] ϕ[L]], {i, nf}];
(*Computation of the vector of external moments*)
mext = Table[(M0[t][[i]]) Join[-Table[Dϕ[0][[i]] × Dϕ[0][[j]], {i, 2 Nn}, {j, 2 Nn}].qy[t][[i]],
     Table[Dϕ[0][[i]] × Dϕ[0][[j]], {i, 2 Nn}, {j, 2 Nn}].qx[t][[i]]], {i, nf}];
```

Reactive Spring Forces

```
(*Euclidean distance between two ends of an Sαᵢ spring*)
γα =
  Join[{Sqrt[(Xa[0, t][[1]] - mfpt[t][[1, 1]])^2 + (Ya[0, t][[1]] - mfpt[t][[1, 2]])^2]}, Table[
    Sqrt[(Xa[0, t][[i]] - Xa[0, t][[i - 1]])^2 + (Ya[0, t][[i]] - Ya[0, t][[i - 1]])^2], {i, 2, nf}],
   {Sqrt[(Xa[0, t][[nf]] - mfpt[t][[nf + 2, 1]])^2 + (Ya[0, t][[nf]] - mfpt[t][[nf + 2, 2]])^2]}];
(*Change in length of an Sαᵢ spring*)
Δxα = Table[Lα - γα[[i]], {i, nf + 1}];

(*Euclidean distance between two ends of an Sβᵢ spring*)
γβ = Table[
   Sqrt[(Xa[0, t][[i]] - mfpt[t][[i + 1, 1]])^2 + (Ya[0, t][[i]] - mfpt[t][[i + 1, 2]])^2], {i, nf}];
(*Change in length of an Sβᵢ spring*)
Δxβ = Table[Lβ - γβ[[i]], {i, nf}];
```

```
(*x-component of the reactive spring force vector*)
Fsx = Join[{(α Δxα[[1]] / γα[[1]]) (Xa[0, t][[1]] - mfpt[t][[1, 1]]) + (β Δxβ[[1]] / γβ[[1]])
       (Xa[0, t][[1]] - mfpt[t][[2, 1]]) + (α Δxα[[2]] / γα[[2]]) (Xa[0, t][[1]] - Xa[0, t][[2]])},
   Table[(α Δxα[[i]] / γα[[i]]) (Xa[0, t][[i]] - Xa[0, t][[i - 1]]) +
     (β Δxβ[[i]] / γβ[[i]]) (Xa[0, t][[i]] - mfpt[t][[i + 1, 1]]) +
     (α Δxα[[i + 1]] / γα[[i + 1]]) (Xa[0, t][[i]] - Xa[0, t][[i + 1]]), {i, 2, nf - 1}],
   {(α Δxα[[nf]] / γα[[nf]]) (Xa[0, t][[nf]] - Xa[0, t][[nf - 1]]) +
     (β Δxβ[[nf]] / γβ[[nf]]) (Xa[0, t][[nf]] - mfpt[t][[nf + 1, 1]]) +
     (α Δxα[[nf + 1]] / γα[[nf + 1]]) (Xa[0, t][[nf]] - mfpt[t][[nf + 2, 1]])}];
(*y-component of the reactive spring force vector*)
Fsy = Join[{(α Δxα[[1]] / γα[[1]]) (Ya[0, t][[1]] - mfpt[t][[1, 2]]) + (β Δxβ[[1]] / γβ[[1]])
       (Ya[0, t][[1]] - mfpt[t][[2, 2]]) + (α Δxα[[2]] / γα[[2]]) (Ya[0, t][[1]] - Ya[0, t][[2]])},
   Table[(α Δxα[[i]] / γα[[i]]) (Ya[0, t][[i]] - Ya[0, t][[i - 1]]) +
     (β Δxβ[[i]] / γβ[[i]]) (Ya[0, t][[i]] - mfpt[t][[i + 1, 2]]) +
     (α Δxα[[i + 1]] / γα[[i + 1]]) (Ya[0, t][[i]] - Ya[0, t][[i + 1]]), {i, 2, nf - 1}],
   {(α Δxα[[nf]] / γα[[nf]]) (Ya[0, t][[nf]] - Ya[0, t][[nf - 1]]) +
     (β Δxβ[[nf]] / γβ[[nf]]) (Ya[0, t][[nf]] - mfpt[t][[nf + 1, 2]]) +
     (α Δxα[[nf + 1]] / γα[[nf + 1]]) (Ya[0, t][[nf]] - mfpt[t][[nf + 2, 2]])}];

(*Computation of the vector of reactive spring forces*)
fspring = Table[Join[Fsx[[i]] ϕ[0], Fsy[[i]] ϕ[0]], {i, nf}];
```

Total Input Vector

```
(*Total input vector consisting of reactive spring forces,
external forces and moments*)
UU = fext + fspring + mext +
   Table[ρ g Join[Integrate[ϕ[x], {x, 0, L}], ConstantArray[0, 2 Nn]], {i, nf}];
```

Finite Element Discrete Form

```
(*Equation of motion for all the legs in finite element discrete form*)
eqnleg = Flatten[Table[
    Thread[MM.(ρ q''[t][[j]] + ξ q'[t][[j]]) - LL[[j]] + NM[[j]].q[t][[j]] - UU[[j]] == 0], {j, nf}]];
```

Equations of Motion of the Robot's Body

Kinetic Energy

```
(*Moment of inertia of the robot's body about its center of mass*)
Ibody = (1 / 12) (m) ((len^2 + brd^2) - ((len - 2 tns)^2 + (brd - tns)^2));
(*Kinetic energy of the robot's body*)
Kbody = (1 / 2) (m) (Take[D[qbody[t], t], 2].Take[D[qbody[t], t], 2]) +
   (1 / 2) (Ibody) (Take[D[qbody[t], t], -1].Take[D[qbody[t], t], -1]);
```

Potential Energy

```
(*Potential energy due to gravity*) Ebody = m g (Xterr - qbody[t][[1]]);
```

Lagrangian

```
(*Computation of the Lagrangian for the robot body*) Lbody = Kbody - Ebody;
```

Non-Conservative Forces

Damping Forces

```
(*Rayleigh dissipation function*) Rbody = (ξ / 2) qbody'[t].qbody'[t];
(*Generalized damping force vector*)
QDbody = Table[-D[Rbody, qbody'[t][[i]]], {i, Length[qbody[t]]}];
```

Reactive Spring Forces

```
(*List of reactive spring forces acting on the body of the robot*)
FSbody = Join[{{(α Δxα[[1]] / γα[[1]]) (mfpt[t][[1, 1]] - Xa[0, t][[1]]),
      (α Δxα[[1]] / γα[[1]]) (mfpt[t][[1, 2]] - Ya[0, t][[1]])}},
   Table[{(β Δxβ[[i]] / γβ[[i]]) (mfpt[t][[i + 1, 1]] - Xa[0, t][[i]]),
     (β Δxβ[[i]] / γβ[[i]]) (mfpt[t][[i + 1, 2]] - Ya[0, t][[i]])}, {i, nf}],
   {{(α Δxα[[nf + 1]] / γα[[nf + 1]]) (mfpt[t][[nf + 2, 1]] - Xa[0, t][[nf]]),
     (α Δxα[[nf + 1]] / γα[[nf + 1]]) (mfpt[t][[nf + 2, 2]] - Ya[0, t][[nf]])}}];

(*Generalized reactive spring force vector*)
QSbody = Table[Sum[FSbody[[i]].D[mfpt[t][[i]], qbody[t][[j]]], {i, nf + 2}],
    {j, Length[qbody[t]]}]; // FullSimplify
```

Normal Force

```
(*Coordinates for points P1 and P2*)
P1 = rtran /@ {{fpt[t][[nf + 2, 1]] + clc, fpt[t][[nf + 2, 2]] + tns}} // Flatten;
P2 = rtran /@ {{fpt[t][[1, 1]] + clc, fpt[t][[1, 2]] - tns}} // Flatten;
(*Normal force vectors acting on points P1 and P2*)
FNP1 = {Piecewise[
    {{Piecewise[{{-ϵ * (P1[[1]] - Xterr), P1[[1]] ≥ Xterr}}, 0], qbody[t][[3]] ≤ 0}}, 0], 0};
FNP2 = {Piecewise[
    {{Piecewise[{{-ϵ * (P2[[1]] - Xterr), P2[[1]] ≥ Xterr}}, 0], qbody[t][[3]] > 0}}, 0], 0};

(*Generalized normal force vector*)
QNbody = Table[FNP1.D[P1, qbody[t][[j]]], {j, Length[qbody[t]]}] +
   Table[FNP2.D[P2, qbody[t][[j]]], {j, Length[qbody[t]]}];
```

Vector of Generalized Forces

```
(*Generalized non-conservative force vector acting on the body of the robot*)
Qbody = QDbody + QSbody + QNbody;
```

Euler-Lagrange Equations

```
(*Equations of motion for the body of the robot*)
eqnbody = Table[D[D[Lbody, qbody'[t][[i]]], t] - D[Lbody, qbody[t][[i]]] == Qbody[[i]],
   {i, Length[qbody[t]]}];
```

System Solver and Solutions

System of Equations

```
(*Complete system of equations describing the motion of the robot*)
diffeqn = Chop[Join[eqnleg, eqnbody]];
```

Solver

```
(*Event condition is used to stop executing NDSolve*)
eventcond = {WhenEvent[stop, {tf = t;
     "StopIntegration"}, "LocationMethod" → "LinearInterpolation"]};

(*In case NDSolve takes more time than expected to solve the differential equations,
the stop button below can be clicked to obtain the
 solution evaluated until the time when stop button was clicked*)
Button["Stop", stop = True]
(*The following command shows the progress of NDSolve at the bottom left
 corner of the screen. It shows the amount of elapsed integration time*)
showStatus[status_] := LinkWrite[$ParentLink,
   SetNotebookStatusLine[FrontEnd`EvaluationNotebook[], ToString[status]]];
clearStatus[] := showStatus[""];
clearStatus[]

(*NDSolve function to find the solution for the differential equations*)
stop = False;
soleqn = First@NDSolve[Join[diffeqn, icvalues, eventcond],
     Join[(q[t] // Flatten), qbody[t]], {t, 0, tf}, AccuracyGoal → 3,
     PrecisionGoal → 3, EvaluationMonitor :→ showStatus["t = " <> ToString[CForm[t]]],
     Method → {"EquationSimplification" → Residual}];
```

Solutions

```
(*Intervals at which the solutions are collected*)
interval = 1 / 100;

(*Solutions for all the time-varying variables of the legs*)
qDet[t] = q[t] /. soleqn;
(*Function that gives the value of X(s,t) for all the legs*)
XDet[x_, t_] = Table[ϕ[x].Take[qDet[t][[j]], 2 Nn], {j, nf}];
(*Function that gives the value of Y(s,t) for all the legs*)
YDet[x_, t_] = Table[ϕ[x].Drop[qDet[t][[j]], 2 Nn], {j, nf}];
(*Function that gives the value of X'(s,t) for all the legs*)
DXDet[x_, t_] = Table[Dϕ[x].Take[qDet[t][[j]], 2 Nn], {j, nf}];
(*Function that gives the value of Y'(s,t) for all the legs*)
DYDet[x_, t_] = Table[Dϕ[x].Drop[qDet[t][[j]], 2 Nn], {j, nf}];
(*Function that gives the value of X''(s,t) for all the legs*)
DDXDet[x_, t_] = Table[DDϕ[x].Take[qDet[t][[j]], 2 Nn], {j, nf}];
(*Function that gives the value of Y''(s,t) for all the legs*)
DDYDet[x_, t_] = Table[DDϕ[x].Drop[qDet[t][[j]], 2 Nn], {j, nf}];
(*Function that gives the value of X'''(s,t) for all the legs*)
DDDXDet[x_, t_] = Table[DDDϕ[x].Take[qDet[t][[j]], 2 Nn], {j, nf}];
```

```
(*Function that gives the value of Y'''(s,t) for all the legs*)
DDDYDet[x_, t_] = Table[DDDϕ[x].Drop[qDet[t][[j]], 2 Nn], {j, nf}];

(*Function to evaluate position vector of the
 neutral axis at arclength x and time t of the jth leg*)
rDet[x_, t_, j_] := {XDet[x, t][[j]], YDet[x, t][[j]]};
(*Function to evaluate curvature of the
 neutral axis at arclength x and time t of the jth leg*)
κDet[x_, t_, j_] := Sqrt[(DDXDet[x, t][[j]])^2 + (DDYDet[x, t][[j]])^2];
(*Functions to evaluate local tangent angle of the
 neutral axis at arclength x and time t of the jth leg*)
ψDet[x_, t_, j_] := ArcTan[DXDet[x, t][[j]], DYDet[x, t][[j]]];
(*Solution for the generalized coordinates of the robot's body*)
qbodydet[t_] = qbody[t] /. soleqn;
(*Solution for the points at which the springs are fixed to the robot's body*)
fixptdata[t_] = mfpt[t] /. soleqn;

(*Data generation for animation*)
Module[{},
  (*Leg animation data*)
  anidata = Table[rDet[n[[i]], t, j], {t, 0, tf, interval}, {j, nf}, {i, Nn}];
  (*Sα springs animation data*)
  Sαdata = Table[{{XDet[0, t][[j]], YDet[0, t][[j]]}, {XDet[0, t][[j + 1]], YDet[0, t][[j + 1]]}},
    {j, nf - 1}, {t, 0, tf, interval}];
  (*Sβ springs animation data*)
  Sβdata =
   Join[{Table[{fixptdata[t][[1]], {XDet[0, t][[1]], YDet[0, t][[1]]}}, {t, 0, tf, interval}]},
    Table[{fixptdata[t][[j + 1]], {XDet[0, t][[j]], YDet[0, t][[j]]}},
     {j, nf}, {t, 0, tf, interval}], {Table[
      {fixptdata[t][[-1]], {XDet[0, t][[-1]], YDet[0, t][[-1]]}}, {t, 0, tf, interval}]}] ;
  (*Robot's body animation data*)
  bodydata = Table[body[qbodydet[t]], {t, 0, tf, interval}];
  (*Terrain animation data*)
  terrdata = Table[Xterr, {t, 0, tf, interval}];
 ];

(*Random color generating function to encode individual leg*)
colordata = RandomColor[nf + 1];

(*Plot range for the animation. Ensure that the X-range and Y-
 range of the plot range is equal. If not equal, the animation will be skewed*)
pltrng = {{-16.5, 16.5}, {-6.5, 26.5}};
```

```
(*Robot animation*)
plt1 =
  Manipulate[Legended[Rotate[Show[ListPlot[anidata[[j]], Joined → True, PlotRange → pltrng,
       Axes → False, PlotStyle → Thread[{Thickness[0.005], colordata}]],
      Graphics[{Table[{Pink, spring[Sαdata[[i]][[j]]]}, {i, Length[Sαdata]}],
        Table[{Pink, spring[Sβdata[[i]][[j]]]}, {i, Length[Sβdata]}],
        {bodydata[[j]]}, {{InfiniteLine[{Xterr, Xterr}, {0, Xterr}]}},
        {{Rotate[Text[Style["Robot's Position on the Terrain",
             FontWeight → Bold, FontSize → 15], {-2.5, 10}], 90 Degree]}},
        {{Rotate[Text[Style["Robot's Configuration", FontWeight → Bold,
             FontSize → 15], {-15.9, 10}], 90 Degree]}},
        {{Rotate[Text[Style["Envelope of Tips of Legs", FontWeight → Bold, FontSize → 15],
            {10.5, 10}], 90 Degree]}}}, PlotRange → pltrng], Epilog →
       {Inset[Show[Graphics[{Table[{Pink, spring[Sαdata[[i]][[j]]]}, {i, Length[Sαdata]}],
            Table[{Pink, spring[Sβdata[[i]][[j]]]}, {i, Length[Sβdata]}],
            {bodydata[[j]]}, {{InfiniteLine[{Xterr, Xterr}, {0, Xterr}]}}}],
          ListPlot[anidata[[j]], Joined → True, PlotRange → Full, Axes → False,
           PlotStyle → Thread[{Thickness[0.03], colordata}]]], {-12.5, Center}],
        Inset[Show[ListPlot[Table[anidata[[j, k, -1]], {k, Dimensions[anidata][[2]]}],
           Joined → True, Axes → False, PlotRange → pltrng, PlotStyle → colordata[[nf + 1]]],
         {12.5, Center}, Automatic, 32]}], -90 Degree],
    Join[Table[LineLegend[{colordata[[i]]}, {StringJoin["Leg ", ToString[i]]}], {i, nf}],
     {LineLegend[{colordata[[nf + 1]]}, {"Envelope"}]}]],
   {j, 1, Length[anidata], 1},
   ControlPlacement →
    Top];

(*Plot of X-coordinate of the tips of the legs*)
plt2 = Plot[(XDet[L, t] // Evaluate), {t, 0, tf}, PlotRange → All, PlotStyle → colordata,
   PlotLegends → Table[StringJoin["Leg ", ToString[i]], {i, nf}], Axes → False,
   Frame → True, FrameLabel → {"Time (in Seconds)", "X-Coordinate"}];

(*Plot of X-coordinate of the tips of the legs*)
plt3 = Plot[(YDet[L, t] // Evaluate), {t, 0, tf}, PlotRange → All, PlotStyle → colordata,
   PlotLegends → Table[StringJoin["Leg ", ToString[i]], {i, nf}], Axes → False,
   Frame → True, FrameLabel → {"Time (in Seconds)", "Y-Coordinate"}];

(*Plot of the basal angles of the legs*)
plt4 = Plot[Table[ψDet[0, t, j], {j, nf}] / Degree // Evaluate, {t, 0, tf}, PlotRange → All,
   PlotStyle → colordata, PlotLegends → Table[StringJoin["Leg ", ToString[i]], {i, nf}],
   Axes → False, Frame → True, FrameLabel → {"Time (in Seconds)", "Angle (in Degrees)"}];

(*Plot of the inextensibility condition of the legs*)
plt5 = Plot[(VV /. soleqn) // Evaluate, {t, 0, tf}, PlotRange → All,
   PlotStyle → colordata, PlotLegends → Table[StringJoin["Leg ", ToString[i]], {i, nf}],
   Axes → False, Frame → True, FrameLabel → {"Time (in Seconds)", "Error"}];


(*Plot of the center of mass coordinates of the robot's body*)
plt6 = Plot[Take[qbodydet[t], 2] // Evaluate, {t, 0, tf}, PlotRange → All,
   PlotLegends → {"X-Coordinate of the Body", "Y-Coordinate of the Body"},
   Axes → False, Frame → True, FrameLabel → {"Time (in Seconds)", "Coordinate"}];
```

Bibliography

[1] Seyed Saman Khedmatgozar Dolati et al. "Non-Destructive Testing Applications for Steel Bridges". In: *Applied Sciences* 11.20 (20 Jan. 2021), p. 9757. DOI: 10.3390/app11209757. URL: http://www.mdpi.com/2076-3417/11/20/9757.

[2] Rami A. Mattar and Remy Kalai. "Development of a Wall-Sticking Drone for Non-Destructive Ultrasonic and Corrosion Testing". In: *Drones* 2.1 (1 Mar. 2018), p. 8. DOI: 10.3390/drones2010008. URL: http://www.mdpi.com/2504-446X/2/1/8.

[3] Jamie Luong et al. "Eversion and Retraction of a Soft Robot Towards the Exploration of Coral Reefs". In: *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. 2019 2nd IEEE International Conference on Soft Robotics (RoboSoft). Apr. 2019, pp. 801–807. DOI: 10.1109/ROBOSOFT.2019.8722730.

[4] Jun Qian et al. "The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System". In: *Sensors* 17.9 (9 Sept. 2017), p. 2073. DOI: 10.3390/s17092073. URL: http://www.mdpi.com/1424-8220/17/9/2073.

[5] Felisa Cordova and Victor Olivares. "Design of Drone Fleet Management Model in a Production System of Customized Products". In: *2016 6th International Conference on Computers Communications and Control (ICCCC)*. 2016 6th International Conference on Computers Communications and Control (ICCCC). May 2016, pp. 165–172. DOI: 10.1109/ICCCC.2016.7496756.

[6] Björn Lindqvist et al. "Multimodality Robotic Systems: Integrated Combined Legged-Aerial Mobility for Subterranean Search-and-Rescue". In: *Robotics and Autonomous Systems* 154 (Aug. 1, 2022), p. 104134. DOI: 10.1016/j.robot.2022.104134. URL: https://www.sciencedirect.com/science/article/pii/S0921889022000756.

[7] Widodo Budiharto et al. "Development of Tank-Based Military Robot and Object Tracker". In: *2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*. 2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS). July 2019, pp. 221–224. DOI: 10.1109/ACIRS.2019.8935962.

[8] Junjie Chen et al. "AiRobSim: Simulating a Multisensor Aerial Robot for Urban Search and Rescue Operation and Training". In: *Sensors* 20.18 (18 Jan. 2020), p. 5223. DOI: 10.3390/s20185223. URL: http://www.mdpi.com/1424-8220/20/18/5223.

[9] James Trevelyan, William R. Hamel, and Sung-Chul Kang. "Robotics in Hazardous Applications". In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Springer Handbooks. Cham: Springer International Publishing, 2016, pp. 1521–1548. DOI: 10.1007/978-3-319-32552-1_58. URL: https://doi.org/10.1007/978-3-319-32552-1_58.

[10] Emaad Mohamed H. Zahugi, Mohamed M. Shanta, and T. V. Prasad. "Oil Spill Cleaning Up Using Swarm of Robots". In: *Advances in Computing and Information Technology*. Ed. by Natarajan Meghanathan, Dhinaharan Nagamalai, and Nabendu Chaki. Advances in Intelligent Systems and Computing. Berlin, Heidelberg: Springer, 2013, pp. 215–224. DOI: 10.1007/978-3-642-31600-5_22.

[11] Nathan Michael et al. "Collaborative Mapping of an Earthquake Damaged Building via Ground and Aerial Robots". In: *Field and Service Robotics: Results of the 8th International Conference*. Ed. by Kazuya Yoshida and Satoshi Tadokoro. Springer Tracts in Advanced Robotics. Berlin, Heidelberg: Springer, 2014, pp. 33–47. DOI: 10.1007/978-3-642-40686-7_3. URL: https://doi.org/10.1007/978-3-642-40686-7_3.

[12] Farbod Fahimi. *Autonomous Robots*. Boston, MA: Springer US, 2009. DOI: 10.1007/978-0-387-09538-7. URL: http://link.springer.com/10.1007/978-0-387-09538-7.

[13] Chenghui Nie, Xavier Pacheco Corcho, and Matthew Spenko. "Robots on the Move: Versatility and Complexity in Mobile Robot Locomotion". In: *IEEE Robotics & Automation Magazine* 20.4 (Dec. 2013), pp. 72–82. DOI: 10.1109/MRA.2013.2248310.

[14] Xiaodong Zhou and Shusheng Bi. "A Survey of Bio-Inspired Compliant Legged Robot Designs". In: *Bioinspiration & Biomimetics* 7.4 (Dec. 1, 2012), p. 041001. DOI: 10.1088/1748-3182/7/4/041001. URL: https://iopscience.iop.org/article/10.1088/1748-3182/7/4/041001.

[15] Hans Machemer. "Ciliary Activity and the Origin of Metachrony in Paramecium: Effects of Increased Viscosity". In: *Journal of Experimental Biology* 57.1 (1972), pp. 239–259. DOI: 10.1242/jeb.57.1.239.

[16] Raymond E. Goldstein. "Green Algae as Model Organisms for Biological Fluid Dynamics". In: *Annual Review of Fluid Mechanics* 47.1 (2015), pp. 343–375. DOI: 10.1146/annurev-fluid-010313-141426.

[17] Thomas Niedermayer, Bruno Eckhardt, and Peter Lenz. "Synchronization, Phase Locking, and Metachronal Wave Formation in Ciliary Chains". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 18.3 (Sept. 2008), p. 037128. DOI: 10.1063/1.2956984. URL: https://aip.scitation.org/doi/full/10.1063/1.2956984.

[18] Sidney L. Tamm. "CILIARY MOTION IN PARAMECIUM : A Scanning Electron Microscope Study". In: *Journal of Cell Biology* 55.1 (Oct. 1, 1972), pp. 250–255. DOI: 10.1083/jcb.55.1.250. URL: https://doi.org/10.1083/jcb.55.1.250.

[19] Klaus Hausmann and Richard D. Allen. "Chapter 7 - Electron Microscopy of Paramecium (Ciliata)". In: *Methods in Cell Biology*. Ed. by Thomas Müller-Reichert. Vol. 96. Electron Microscopy of Model Systems. Academic Press, Jan. 1, 2010, pp. 143–173. DOI: 10.1016/S0091-679X(10)96007-X. URL: https://www.sciencedirect.com/science/article/pii/S0091679X1096007X.

[20] A. Hilfinger and F. Jülicher. "The Chirality of Ciliary Beats". In: *Physical Biology* 5.1 (Mar. 2008), p. 016003. DOI: 10.1088/1478-3975/5/1/016003. URL: https://dx.doi.org/10.1088/1478-3975/5/1/016003.

[21] Kirsty Y. Wan and Raymond E. Goldstein. "Coordinated Beating of Algal Flagella Is Mediated by Basal Coupling". In: *Proceedings of the National Academy of Sciences* 113.20 (May 17, 2016), E2784–E2793. DOI: 10.1073/pnas.1518527113. URL: https://www.pnas.org/doi/10.1073/pnas.1518527113.

[22] Arkady Pikovsky, Michael Rosenblum, and Jürgen Kurths. *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge Nonlinear Science Series. Cambridge: Cambridge University Press, 2001. DOI: 10.1017/CBO9780511755743.

[23] Donald M. Wilson. "Insect Walking". In: *Annual Review of Entomology* 11.1 (1966), pp. 103–122. DOI: 10.1146/annurev.en.11.010166.000535. pmid: 5321575. URL: https://doi.org/10.1146/annurev.en.11.010166.000535.

[24] G.C. Haynes and A.A. Rizzi. "Gaits and Gait Transitions for Legged Robots". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. May 2006, pp. 1117–1122. DOI: 10.1109/ROBOT.2006.1641859.

[25] Jie Xu et al. "Multi-Objective Graph Heuristic Search for Terrestrial Robot Design". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021 IEEE International Conference on Robotics and Automation (ICRA). May 2021, pp. 9863–9869. DOI: 10.1109/ICRA48506.2021.9561818.

[26] Robert Baines, Frank Fish, and Rebecca Kramer-Bottiglio. "Amphibious Robotic Propulsive Mechanisms: Current Technologies and Open Challenges". In: *Bioinspired Sensing, Actuation, and Control in Underwater Soft Robotic Systems*. Ed. by Derek A. Paley and Norman M. Wereley. Cham: Springer International Publishing, 2021, pp. 41–69. DOI: 10.1007/978-3-030-50476-2_3. URL: http://link.springer.com/10.1007/978-3-030-50476-2_3.

[27] Johan K Schonebaum, Farbod Alijani, and Giuseppe Radaelli. "Review on Mobile Robots That Exploit Resonance". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 235.24 (Dec. 1, 2021), pp. 7907–7924. DOI: 10.1177/09544062211036923. URL: https://doi.org/10.1177/09544062211036923.

[28] Bruno Siciliano et al. *Robotics*. Red. by Michael J. Grimble and Michael A. Johnson. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2009. DOI: 10.1007/978-1-84628-642-1. URL: http://link.springer.com/10.1007/978-1-84628-642-1.

[29] Francisco Rubio, Francisco Valero, and Carlos Llopis-Albert. "A Review of Mobile Robots: Concepts, Methods, Theoretical Framework, and Applications". In: *International Journal of Advanced Robotic Systems* 16.2 (Mar. 1, 2019), p. 1729881419839596. DOI: 10.1177/1729881419839596. URL: https://doi.org/10.1177/1729881419839596.

[30] Mary B. Alatise and Gerhard P. Hancke. "A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods". In: *IEEE Access* 8 (2020), pp. 39830–39846. DOI: 10.1109/ACCESS.2020.2975643.

[31] Ting Zou, Jorge Angeles, and Ferri Hassani. "Dynamic Modeling and Trajectory Tracking Control of Unmanned Tracked Vehicles". In: *Robotics and Autonomous Systems* 110 (Dec. 2018), pp. 102–111. DOI: 10.1016/j.robot.2018.09.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S0921889018300319.

[32] Shugen. "Analysis of Creeping Locomotion of a Snake-like Robot". In: *Advanced Robotics* 15.2 (Jan. 2001), pp. 205–224. DOI: 10.1163/15685530152116236. URL: https://www.tandfonline.com/doi/full/10.1163/15685530152116236.

[33] David Zarrouk, Inna Sharf, and Moshe Shoham. "Analysis of Wormlike Robotic Locomotion on Compliant Surfaces". In: *IEEE Transactions on Biomedical Engineering* 58.2 (Feb. 2011), pp. 301–309. DOI: 10.1109/TBME.2010.2066274.

[34] Z.Y. Bayraktaroglu. "Snake-like Locomotion: Experimentations with a Biologically Inspired Wheel-Less Snake Robot". In: *Mechanism and Machine Theory* 44.3 (Mar. 2009), pp. 591–602. DOI: 10.1016/j.mechmachtheory.2008.08.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S0094114X08001791.

[35] S. Hirose and M. Mori. "Biologically Inspired Snake-like Robots". In: *2004 IEEE International Conference on Robotics and Biomimetics*. 2004 IEEE International Conference on Robotics and Biomimetics. Aug. 2004, pp. 1–7. DOI: 10.1109/ROBIO.2004.1521742.

[36] Ran Liu and Yan-an Yao. "A Novel Serial–Parallel Hybrid Worm-like Robot with Multi-Mode Undulatory Locomotion". In: *Mechanism and Machine Theory* 137 (July 2019), pp. 404–431. DOI: 10.1016/j.mechmachtheory.2019.03.033. URL: https://linkinghub.elsevier.com/retrieve/pii/S0094114X18314356.

[37] Yasemin Ozkan-Aydin et al. "Lateral Bending and Buckling Aids Biological and Robotic Earthworm Anchoring and Locomotion". In: *Bioinspiration & Biomimetics* 17.1 (Jan. 1, 2022), p. 016001. DOI: 10.1088/1748-3190/ac24bf. URL: https://iopscience.iop.org/article/10.1088/1748-3190/ac24bf.

[38] V. Racek, J. Sitar, and D. Maga. "Design of Combined Snake Robot". In: *Recent Advances in Mechatronics*. Ed. by Ryszard Jabłoński, Mateusz Turkowski, and Roman Szewczyk. Berlin, Heidelberg: Springer, 2007, pp. 72–76. DOI: 10.1007/978-3-540-73956-2_15.

[39] Aksel Andreas Transeth and Kristin Ytterstad Pettersen. "Developments in Snake Robot Modeling and Locomotion". In: *Robotics and Vision 2006 9th International Conference on Control, Automation*. Robotics and Vision 2006 9th International Conference on Control, Automation. Dec. 2006, pp. 1–8. DOI: 10.1109/ICARCV.2006.345142.

[40] Lounis Douadi, Davide Spinello, and Wail Gueaieb. "Dynamics and Control of a Planar Multibody Mobile Robot for Confined Environment Inspection". In: *Journal of Computational and Nonlinear Dynamics* 10.1 (Sept. 12, 2014). DOI: 10.1115/1.4027303.

[41] Matteo Russo and Marco Ceccarelli. "A Survey on Mechanical Solutions for Hybrid Mobile Robots". In: *Robotics* 9.2 (2 June 2020), p. 32. DOI: 10.3390/robotics9020032. URL: http://www.mdpi.com/2218-6581/9/2/32.

[42] Magno Guedes et al. "ARES-III: A Versatile Multi-Purpose All-Terrain Robot". In: *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*. Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012). Sept. 2012, pp. 1–8. DOI: 10.1109/ETFA.2012.6489633.

[43] Wenhao Hao et al. "Enhancing the Obstacle-crossing Performance of All-Terrain Vehicle Based on Variable-wheelbase Chassis". In: *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). July 2019, pp. 1329–1334. DOI: 10.1109/AIM.2019.8868337.

[44] Aaron M. Johnson et al. "Autonomous Legged Hill and Stairwell Ascent". In: *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics. Nov. 2011, pp. 134–142. DOI: 10.1109/SSRR.2011.6106785.

[45] Takahide Yoshiike et al. "Development of Experimental Legged Robot for Inspection and Disaster Response in Plants". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Sept. 2017, pp. 4869–4876. DOI: 10.1109/IROS.2017.8206364.

[46] J. Hodgins. "Legged Robots on Rough Terrain: Experiments in Adjusting Step Length". In: *1988 IEEE International Conference on Robotics and Automation Proceedings*. 1988 IEEE International Conference on Robotics and Automation Proceedings. Apr. 1988, 824–826 vol.2. DOI: 10.1109/ROBOT.1988.12162.

[47] Daniel J. Lynch, Kevin M. Lynch, and Paul B. Umbanhowar. "The Soft-Landing Problem: Minimizing Energy Loss by a Legged Robot Impacting Yielding Terrain". In: *IEEE Robotics and Automation Letters* 5.2 (Apr. 2020), pp. 3658–3665. DOI: 10.1109/LRA.2020.2977260.

[48] X. Alice Wu et al. "Tactile Sensing and Terrain-Based Gait Control for Small Legged Robots". In: *IEEE Transactions on Robotics* 36.1 (Feb. 2020), pp. 15–27. DOI: 10.1109/TRO.2019.2935336.

[49] Davide Spinello and Javad S. Fattahi. "Peristaltic Wave Locomotion and Shape Morphing with a Millipede Inspired System". In: *Journal of Nonlinear Science* 27.4 (Aug. 2017), pp. 1093–1119. DOI: 10.1007/s00332-017-9372-7. URL: http://link.springer.com/10.1007/s00332-017-9372-7.

[50] Wenjie Chen, K. H. Low, and S. H. Yeo. "Adaptive Gait Planning for Multi-Legged Robots with an Adjustment of Center-of-Gravity". In: *Robotica* 17.4 (July 1999), pp. 391–403. DOI: 10.1017/S0263574799000958. URL: http://www.cambridge.org/core/journals/robotica/article/adaptive-gait-planning-for-multilegged-robots-with-an-adjustment-of-centerofgravity/CCB47BB034BE691D4DB909FFCD0FF7F0.

[51] Xi Chen et al. "Prototype Development and Gait Planning of Biologically Inspired Multi-Legged Crablike Robot". In: *Mechatronics* 23.4 (June 1, 2013), pp. 429–444. DOI: 10.1016/j.mechatronics.2013.03.006. URL: https://www.sciencedirect.com/science/article/pii/S0957415813000482.

[52] Jun Zhang et al. "Crab-Inspired Compliant Leg Design Method for Adaptive Locomotion of a Multi-Legged Robot". In: *Bioinspiration & Biomimetics* 17.2

(Jan. 2022), p. 025001. DOI: 10.1088/1748-3190/ac45e6. URL: https://dx.doi.org/10.1088/1748-3190/ac45e6.

[53] Shin-Min Song and K. J. Waldron. "Geometric Design of a Walking Machine for Optimal Mobility". In: *Journal of Mechanisms, Transmissions, and Automation in Design* 109.1 (Mar. 1, 1987), pp. 21–28. DOI: 10.1115/1.3258780. URL: https://doi.org/10.1115/1.3258780.

[54] Chang-De Zhang and Shin-Min Song. "Stability Analysis of Wave-Crab Gaits of a Quadruped". In: *Journal of Robotic Systems* 7.2 (1990), pp. 243–276. DOI: 10.1002/rob.4620070208. URL: http://onlinelibrary.wiley.com/doi/abs/10.1002/rob.4620070208.

[55] Shin-Min Song and Byoung Soo Choi. "On the Stability Properties of 2N-Legged Wave Gaits". In: *Advanced Robotics: 1989*. Ed. by Kenneth J. Waldron. Berlin, Heidelberg: Springer, 1989, pp. 334–348. DOI: 10.1007/978-3-642-83957-3_23.

[56] Xi-Ding Qiu and Shin-Min Song. "A Strategy of Wave Gait for a Walking Machine Traversing a Rough Planar Terrain". In: *Journal of Mechanisms, Transmissions, and Automation in Design* 111.4 (Dec. 1, 1989), pp. 471–478. DOI: 10.1115/1.3259023. URL: https://doi.org/10.1115/1.3259023.

[57] B.S. Choi and S.-M. Song. "Fully Automated Obstacle-Crossing Gaits for Walking Machines". In: *IEEE Transactions on Systems, Man, and Cybernetics* 18.6 (Nov. 1988), pp. 952–964. DOI: 10.1109/21.23093.

[58] Margaret L Byron et al. "Metachronal Motion across Scales: Current Challenges and Future Directions". In: *Integrative and Comparative Biology* 61.5 (Nov. 17, 2021), pp. 1674–1688. DOI: 10.1093/icb/icab105. URL: https://academic.oup.com/icb/article/61/5/1674/6287617.

[59] Kenneth S. Espenschied et al. "Leg Coordination Mechanisms in the Stick Insect Applied to Hexapod Robot Locomotion". In: *Adaptive Behavior* 1.4 (Mar. 1, 1993), pp. 455–468. DOI: 10.1177/105971239300100404. URL: https://doi.org/10.1177/105971239300100404.

[60] Hongri Gu et al. "Magnetic Cilia Carpets with Programmable Metachronal Waves". In: *Nature Communications* 11.1 (1 May 26, 2020), p. 2637. DOI: 10.1038/s41467-020-16458-4. URL: https://www.nature.com/articles/s41467-020-16458-4.

[61] Franco Tedeschi and Giuseppe Carbone. "Design Issues for Hexapod Walking Robots". In: *Robotics* 3.2 (2 June 2014), pp. 181–206. DOI: 10.3390/robotics3020181. URL: https://www.mdpi.com/2218-6581/3/2/181.

[62] Yong Ni et al. "A Novel Wheel-Legged Hexapod Robot". In: *Biomimetics* 7.4 (4 Dec. 2022), p. 146. DOI: 10.3390/biomimetics7040146. URL: https://www.mdpi.com/2313-7673/7/4/146.

[63] Roy E. Ritzmann, Roger D. Quinn, and Martin S. Fischer. "Convergent Evolution and Locomotion through Complex Terrain by Insects, Vertebrates and Robots". In: *Arthropod Structure & Development*. Arthropod Locomotion Systems: From Biological Materials and Systems to Robotics 33.3 (July 1, 2004), pp. 361–379. DOI: 10.1016/j.asd.2004.05.001. URL: https://www.sciencedirect.com/science/article/pii/S1467803904000271.

[64] Bin Zhong et al. "On a CPG-Based Hexapod Robot: AmphiHex-II With Variable Stiffness Legs". In: *IEEE/ASME Transactions on Mechatronics* 23.2 (Apr. 2018), pp. 542–551. DOI: 10.1109/TMECH.2018.2800776.

[65] Seyed Javad Fattahi. "Locomotion and Morphing of a Coupled Bio-Inspired Flexible System: Modeling and Simulation". Thesis. Université d'Ottawa / University of Ottawa, 2015. DOI: 10.20381/ruor-4153. URL: http://ruor.uottawa.ca/handle/10393/32827.

[66] Anthony Garcia, Shashank Priya, and Paul Marek. "Understanding the Locomotion and Dynamic Controls for Millipedes: Part 1 — Kinematic Analysis of Millipede Movements". In: *Volume 2: Integrated System Design and Implementation; Structural Health Monitoring; Bioinspired Smart Materials and Systems; Energy Harvesting*. ASME 2015 Conference on Smart Materials, Adaptive Structures and Intelligent Systems. Colorado Springs, Colorado, USA: American Society of Mechanical Engineers, Sept. 21, 2015, V002T06A005. DOI: 10.1115/SMASIS2015-8894. URL: https://asmedigitalcollection.asme.org/SMASIS/proceedings/SMASIS2015/57304/Colorado%20Springs,%20Colorado,%20USA/284849.

[67] Masahiro Masuda and Kazuyuki Ito. "Semi-Autonomous Centipede-like Robot with Flexible Legs". In: *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*. Oct. 2014, pp. 1–6. DOI: 10.1109/SSRR.2014.7017667.

[68] Jiefeng Sun et al. "Embedded Shape Morphing for Morphologically Adaptive Robots". In: *Nature Communications* 14.1 (Sept. 2023), p. 6023. DOI: 10.1038/s41467-023-41708-6.

[69] Sébastien Camalet and Frank Jülicher. "Generic Aspects of Axonemal Beating". In: *New Journal of Physics* 2.1 (Oct. 2000), p. 324. DOI: 10.1088/1367-2630/2/1/324. URL: https://dx.doi.org/10.1088/1367-2630/2/1/324.

[70] Ingmar H. Riedel-Kruse et al. "How Molecular Motors Shape the Flagellar Beat". In: *HFSP Journal* 1.3 (Sept. 1, 2007), pp. 192–208. DOI: 10.2976/1.2773861. URL: https://doi.org/10.2976/1.2773861.

[71] Alexander Konyukhov and Ridvan Izi. *Introduction to Computational Contact Mechanics: A Geometrical Approach*. John Wiley & Sons, June 15, 2015. 303 pp. Google Books: 0PSyBwAAQBAJ.

[72] Eduardo Corral et al. "Nonlinear Phenomena of Contact in Multibody Systems Dynamics: A Review". In: *Nonlinear Dynamics* 104.2 (Apr. 2021), pp. 1269–1295. DOI: 10.1007/s11071-021-06344-z.

[73] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, May 10, 2014. 412 pp. Google Books: j6LiBQAAQBAJ.

[74] Jacobus D. Brandsen et al. "A Comparative Analysis of Lagrange Multiplier and Penalty Approaches for Modelling Fluid-Structure Interaction". In: *Engineering Computations* 38.4 (Jan. 1, 2020), pp. 1677–1705. DOI: 10.1108/EC-04-2020-0183. URL: https://doi.org/10.1108/EC-04-2020-0183.

[75] S Vulovic et al. "A comparative study of contact problems solution based on the penalty and Lagrange multiplier approaches". In: (2007).

[76] Paulo Flores and Hamid M. Lankarani. *Contact Force Models for Multibody Dynamics*. Vol. 226. Solid Mechanics and Its Applications. Cham: Springer International Publishing, 2016. DOI: 10.1007/978-3-319-30897-5.

[77] Michael A. Sleigh. *The Biology of Cilia and Flagella.* International Series of Monographs on Pure and Applied Biology. Division: Zoology ; v.012. New York: Macmillan, 1962.

[78] Jacky G. Goetz et al. "Endothelial Cilia Mediate Low Flow Sensing during Zebrafish Vascular Development". In: *Cell Reports* 6.5 (Mar. 13, 2014), pp. 799–808. DOI: 10.1016/j.celrep.2014.01.032. URL: https://www.sciencedirect.com/science/article/pii/S2211124714000667.

[79] John Blake. "On the Movement of Mucus in the Lung". In: *Journal of Biomechanics* 8.3 (July 1, 1975), pp. 179–190. DOI: 10.1016/0021-9290(75)90023-8. URL: https://www.sciencedirect.com/science/article/pii/0021929075900238.

[80] Sami Chaaban and Gary J. Brouhard. "A Microtubule Bestiary: Structural Diversity in Tubulin Polymers". In: *Molecular Biology of the Cell* 28.22 (Nov. 2017), pp. 2924–2931. DOI: 10.1091/mbc.e16-05-0271. URL: https://www.molbiolcell.org/doi/full/10.1091/mbc.e16-05-0271.

[81] William Gilpin, Matthew Storm Bull, and Manu Prakash. "The Multiscale Physics of Cilia and Flagella". In: *Nature Reviews Physics* 2.2 (2 Feb. 2020), pp. 74–88. DOI: 10.1038/s42254-019-0129-0. URL: https://www.nature.com/articles/s42254-019-0129-0.

[82] Naoki Numata et al. "Molecular Mechanism of Force Generation by Dynein, a Molecular Motor Belonging to the AAA+ Family". In: *Biochemical Society Transactions* 36.1 (Jan. 22, 2008), pp. 131–135. DOI: 10.1042/BST0360131. URL: https://doi.org/10.1042/BST0360131.

[83] Keith E. Summers and I. R. Gibbons. "Adenosine Triphosphate-Induced Sliding of Tubules in Trypsin-Treated Flagella of Sea-Urchin Sperm". In: *Proceedings of the National Academy of Sciences* 68.12 (Dec. 1971), pp. 3092–3096. DOI: 10.1073/pnas.68.12.3092. URL: https://www.pnas.org/doi/abs/10.1073/pnas.68.12.3092.

[84] T. J. Mitchison and H. M. Mitchison. "How Cilia Beat". In: *Nature* 463.7279 (7279 Jan. 2010), pp. 308–309. DOI: 10.1038/463308a. URL: https://www.nature.com/articles/463308a.

[85] Charles B. Lindemann. "A "Geometric Clutch" Hypothesis to Explain Oscillations of the Axoneme of Cilia and Flagella". In: *Journal of Theoretical Biology* 168.2 (May 21, 1994), pp. 175–189. DOI: 10.1006/jtbi.1994.1097. URL: https://www.sciencedirect.com/science/article/pii/S0022519384710976.

[86] Pablo Sartori et al. "Dynamic Curvature Regulation Accounts for the Symmetric and Asymmetric Beats of Chlamydomonas Flagella". In: *eLife* 5 (May 11, 2016), e13258. DOI: 10.7554/eLife.13258.

[87] Hermes Gadêlha, Eamonn A. Gaffney, and Alain Goriely. "The Counterbend Phenomenon in Flagellar Axonemes and Cross-Linked Filament Bundles". In: *Proceedings of the National Academy of Sciences* 110.30 (July 2013), pp. 12180–12185. DOI: 10.1073/pnas.1302113110.

[88] J. A. Tolomeo and M. C. Holley. "Mechanics of Microtubule Bundles in Pillar Cells from the Inner Ear". In: *Biophysical Journal* 73.4 (Oct. 1997), pp. 2241–2247. DOI: 10.1016/S0006-3495(97)78255-9.

[89] "A Three-Dimensional Simulation of the Dynamics of Primary Cilia in an Oscillating Flow". In: *Applied Mathematical Modelling* 108 (Aug. 1, 2022), pp. 825–839. DOI: 10.1016/j.apm.2022.04.024. URL: https://www.sciencedirect.com/science/article/pii/S0307904X22001974.

[90] Robert H. Dillon and Lisa J. Fauci. "An Integrative Model of Internal Axoneme Mechanics and External Fluid Dynamics in Ciliary Beating". In: *Journal of Theoretical Biology* 207.3 (Dec. 2000), pp. 415–430. DOI: 10.1006/jtbi.2000.2182.

[91] Mark Bathe et al. "Cytoskeletal Bundle Mechanics". In: *Biophysical Journal* 94.8 (Apr. 2008), pp. 2955–2964. DOI: 10.1529/biophysj.107.119743.

[92] Andreas Hilfinger, Amit K. Chattopadhyay, and Frank Jülicher. "Nonlinear Dynamics of Cilia and Flagella". In: *Physical Review E* 79.5 (May 2009), p. 051918. DOI: 10.1103/PhysRevE.79.051918.

[93] Larisa Gheber, Alon Korngreen, and Zvi Priel. "Effect of Viscosity on Metachrony in Mucus Propelling Cilia". In: *Cell Motility* 39.1 (1998), pp. 9–20. DOI: 10.1002/(SICI)1097-0169(1998)39:1<9::AID-CM2>3.0.CO;2-3. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0169%281998%2939%3A1%3C9%3A%3AAID-CM2%3E3.0.CO%3B2-3.

[94] H. Gadêlha et al. "Nonlinear Instability in Flagellar Dynamics: A Novel Modulation Mechanism in Sperm Migration?" In: *Journal of The Royal Society Interface* 7.53 (May 12, 2010), pp. 1689–1697. DOI: 10.1098/rsif.2010.0136. URL: https://royalsocietypublishing.org/doi/full/10.1098/rsif.2010.0136.

[95] Jihun Han and Charles S. Peskin. "Spontaneous Oscillation and Fluid–Structure Interaction of Cilia". In: *Proceedings of the National Academy of Sciences* 115.17 (Apr. 24, 2018), pp. 4417–4422. DOI: 10.1073/pnas.1712042115. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1712042115.

[96] James Gray. *Ciliary Movement | Biophysics and Physiology*. Cambridge University Press. URL: https://www.cambridge.org/ca/academic/subjects/life-sciences/biophysics-and-physiology/ciliary-movement,%20https://www-cambridge-org.proxy.bib.uottawa.ca/ca/academic/subjects/life-sciences/biophysics-and-physiology.

[97] Fanlong Meng et al. "Conditions for Metachronal Coordination in Arrays of Model Cilia". In: *Proceedings of the National Academy of Sciences* 118.32 (Aug. 10, 2021), e2102828118. DOI: 10.1073/pnas.2102828118. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2102828118.

[98] Daniel Tam and A. E. Hosoi. "Optimal Feeding and Swimming Gaits of Biflagellated Organisms". In: *Proceedings of the National Academy of Sciences* 108.3 (Jan. 18, 2011), pp. 1001–1006. DOI: 10.1073/pnas.1011185108. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1011185108.

[99] Andrej Vilfan and Frank Jülicher. "Hydrodynamic Flow Patterns and Synchronization of Beating Cilia". In: *Physical Review Letters* 96.5 (Feb. 6, 2006), p. 058102. DOI: 10.1103/PhysRevLett.96.058102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.96.058102.

[100] Hanliang Guo et al. "Bistability in the Synchronization of Actuated Microfilaments". In: *Journal of Fluid Mechanics* 836 (Feb. 2018), pp. 304–323. DOI: 10.1017/jfm.2017.816.

[101] Yong Woon Kim and Roland R. Netz. "Pumping Fluids with Periodically Beating Grafted Elastic Filaments". In: *Physical Review Letters* 96.15 (Apr. 21, 2006), p. 158101. DOI: 10.1103/PhysRevLett.96.158101. URL: https://link.aps.org/doi/10.1103/PhysRevLett.96.158101.

[102] Greta Quaranta, Marie-Eve Aubin-Tam, and Daniel Tam. "Hydrodynamics Versus Intracellular Coupling in the Synchronization of Eukaryotic Flagella". In: *Physical Review Letters* 115.23 (Nov. 30, 2015), p. 238101. DOI: 10.1103/PhysRevLett.115.238101. URL: https://link.aps.org/doi/10.1103/PhysRevLett.115.238101.

[103] Eric R. Brooks and John B. Wallingford. "Multiciliated Cells". In: *Current Biology* 24.19 (Oct. 6, 2014), R973–R982. DOI: 10.1016/j.cub.2014.08.047. URL: https://www.sciencedirect.com/science/article/pii/S0960982214010604.

[104] Christiaan Huygens. *Horologium oscillatorium sive de motu pendulorum ad horologia aptato demonstrationes geometricae*. Apud F. Muguet, 1673. 186 pp. Google Books: SvkuAAAAQAAJ.

[105] Davide Spinello. "Coordination in Coupled Arrays of Stiff Filaments—Modelling and Simulation". In: *Mathematics* 8.8 (8 Aug. 2020), p. 1282. DOI: 10.3390/math8081282. URL: https://www.mdpi.com/2227-7390/8/8/1282.

[106] Victor A. Toponogov and Vladimir Y. Rovenski. *Differential Geometry of Curves and Surfaces: A Concise Guide*. Boston, UNITED STATES: Birkhäuser Boston, 2006. URL: http://ebookcentral.proquest.com/lib/ottawa/detail.action?docID=3062259.

[107] Raymond E. Goldstein and Stephen A. Langer. "Nonlinear Dynamics of Stiff Polymers". In: *Physical Review Letters* 75.6 (Aug. 7, 1995), pp. 1094–1097. DOI: 10.1103/PhysRevLett.75.1094. URL: https://link.aps.org/doi/10.1103/PhysRevLett.75.1094.

[108] Eduardo Bayo, Javier Garcia De Jalon, and Miguel Angel Serna. "A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems". In: *Computer Methods in Applied Mechanics and Engineering* 71.2 (Nov. 1, 1988), pp. 183–195. DOI: 10.1016/0045-7825(88)90085-0. URL: https://www.sciencedirect.com/science/article/pii/0045782588900850.

[109] Leonard Meirovitch. *Methods of Analytical Dynamics*. Courier Corporation, May 20, 2010. 546 pp. Google Books: kJkoAwAAQBAJ.

[110] Thomas J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Courier Corporation, May 23, 2012. 706 pp. Google Books: cHH2n_qBK0IC.

[111] Frank L. Stasa. *Applied Finite Element Analysis for Engineers*. Holt, Rinehart, and Winston, 1985.

[112] Katie L. Hoffman and Robert J. Wood. "Passive Undulatory Gaits Enhance Walking in a Myriapod Millirobot". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sept. 2011, pp. 1479–1486. DOI: 10.1109/IROS.2011.6094700.