INTRINSIC GEOMETRIC PATH FOR A NEW KINEMATIC TEMPLATE

by

Yan Liu

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Spring 2014

INTRINSIC GEOMETRIC PATH FOR A NEW KINEMATIC TEMPLATE

by

Yan Liu

Approved: _____

Herbert G. Tanner, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____

Suresh G. Advani, Ph.D.
Chair of the Department of Mechanical Engineering

Approved: _____

Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____

James G. Richards, Ph.D.
Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

First, I would like to express my sincere appreciation to my advisors, Dr. Herbert Tanner and Dr. Ioannis Poulakakis, thank them for helping and supporting me during this work. Besides my advisors, I would also like to thank Dr. Michael Keefe, for serving as committee member and his unwavering support and encouragement. Thanks to my lab mates in University of Delaware. It's really nice to have their companion during the master study. The method to quantify the curvature and make path solution for OctoRoACH used the kinematic model proposed by Konstantinos Karydis. Thanks for his help. Finally, I would like to give my thanks to my parents for giving birth to me, raising me up and providing support and encouragement throughout my life. The work in this thesis is supported in part by ARLMAST CTA wg11NF-0-2-0004.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ABSTRACT

OctoRoACH is a miniature eight legged robot which is envisioned to be a step toward the development of small crawling robots that can be used for reconnaissance, search and rescue, sensor coverage, etc. To enable the robot to complete task along these lines, it is necessary to solve the motion planning problem for these types of robotic platforms. In this thesis, we study a kinematic model that serves as an abstraction of the quasi-static motion behavior of the OctoRoACH robot. We see the eight-legged mechanism being abstracted into a switching four-bar linkage. Three motion primitives are defined for this model: straight line motion, left turn arc and right turn arc. The work described in this thesis contributes by providing a method to quantify the radius of curvature of the curved paths generated by this kinematic model. Once the link between model parameters and path curvatures is made, then the wealth of existing motion planning methods, specifically the ones that involve a well studied nonholonomic kinematic model known as the Dubin's car, become relevant to the kinematic analysis and the study of the motion planning problem in these miniature robots.

Chapter 1

INTRODUCTION

1.1 Miniature Legged Robots

Robots are designed to make people's life more convenient, increase production efficiency through automation and in general performing tasks that are not desirable for humans. Compared to stationary robots, which complete tasks within a fixed workspace, mobile robots have a workspace that is not constrained by their own geometry, but by that of their environment. There are three different types of ground mobile robots that move on land: wheeled, tracked and legged. It is quite common to require robots to conduct tasks in unstructured environments. However, the need of wheeled robots for prepared surfaces makes their deployment in such environments problematic, and has restricted them to a small range of the terrestrial terrains. On the other hand, legged robots do not need continuous contact with the ground for every leg and for all time, so they have the potential to operated better on uneven terrain. Miniature legged robots are particularly interesting because they can crawl and squeeze themselves in places that bigger ones cannot. Figure 1.1 shows the miniature robot STAR [1] which is designed in UC Berkeley.

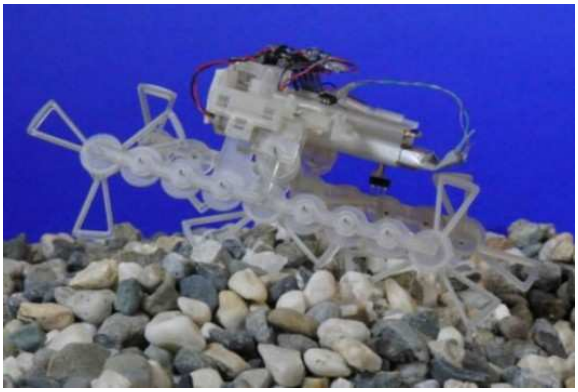


Figure 1.1: A miniature legged robot (STAR) [1]

Inspired by the locomotion behavior of insects, several quarupedal, hexapedal and octopedal robots have been designed in the past ten years. Since 2003, Case Western Reserve University has been designing a series of highly mobile small quadruped robots, which are called as Mini-Whegs (Figure 1.2).

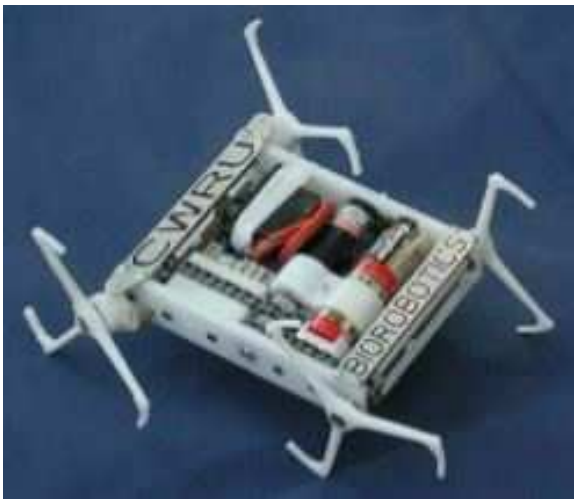


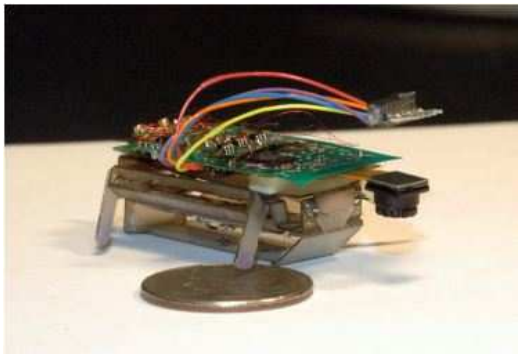
Figure 1.2: Mini-whegs robot [2, 3]

The 9 cm-long quarupedal robot, Mini-whegs can run forward and backward

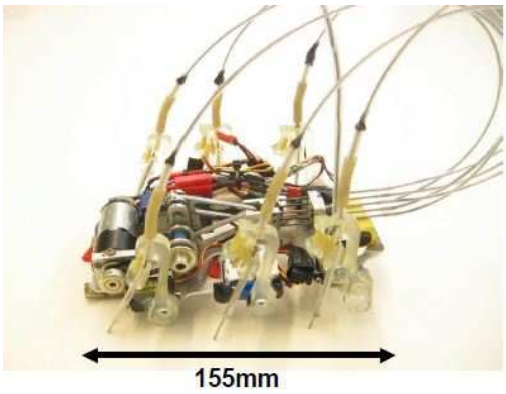
on both left and right sides. The three spoked appendages of each leg, called "whegs" enable it to step over obstacles.

The family of multi-legged crawling robot has been expanded by other researchers with DASH (Figure 1.3a) [4], Medic (Figure 1.3b) [5] and iSprawl (Figure 1.3c) [6].

(a) Dash runs on rough terrain [4]



(b) Medic robot on a US dollar [5]

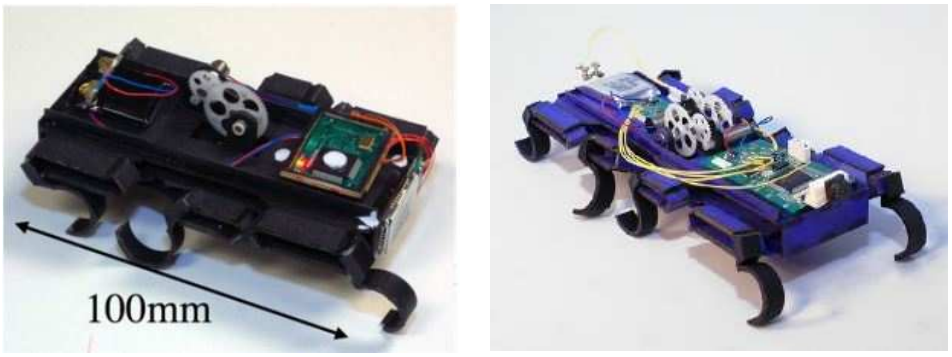


(c) iSprawl robot [6]

Figure 1.3: Crawling Robots

Both DASH and iSprawl are sprawled-posture hexapods, they use a statically-stable switching tripod gait to move in their environment.

In 2010, the 10cm-long hexapedal robot DynaRoACH (Figure 1.4a) [7], was fabricated through a novel manufacturing technique termed the Smart Composite Microstructuring (SCM) [24] process. This fabrication process made it possible to integrate a whole system at a centimeter scale. The eight legged robot OctoRoACH (Figure 1.4b [8]), which we will study in this thesis, resembles DynaRoACH. OctoRoACH is more stable than DynaRoACH. Through using eight-leg configuration, OctoRoACH avoid the intermittent bipedal gait of the opposite center legs [8].



(a) DynaRoACH robot [7] (b) OctoRoACH robot [8]

Figure 1.4: Two crawling robot cousins inspired by cockroach morphology and locomotion behavior

These robots are good for inspection, reconnaissance, search-and-rescue and sensor coverage in unstructured environments because of their small size and unique structure. In order to complete any one of these tasks, it is critical to be able to plan the motion of the robot from point A to point B. Our ultimate goal is to develop a motion planning algorithm which is compatible with the morphology and kinematics of robots such as the ones shown in Figure 1.4. Doing so is nontrivial because existing motion planning methods do not directly apply.

1.2 Motion Planning

Given initial configuration P_i and final configuration P_f, generating a continuous trajectory connecting the start point to the goal point and avoiding obstacles is an instance of a motion planning problem. Motion planning as a problem, appears in several real-world applications, such as designing a collision-free trajectory for mobile robots; enabling robots to conduct space exploration; automatically parking a vehicle in the crowded road; welding, painting and assembling tasks in manufacturing; even in designing a drug which can insert molecules into a protein cavity [9]. The completion of most reconnaissance or search task by a mobile robot typically involves motion planning; the OctoRoACH is no exception.

Usually, motion planning and vehicle navigation are problems that are studied at the discrete level. One can transform the continuous model into a discrete one and implement some type of a discrete combinatorial search [10]. Combinatorial methods can yield algorithmically complete solutions for many motion planning problems. When a solution do exist, the algorithm will find it in finite time, if it doesn't exist, the algorithm will return failure to find the solution, then the method is called algorithmically complete [10]. However, the implementation difficulties and their computational complexity render them inappropriate for many real-world applications. As an alternative to combinatorial methods, sampling-based algorithms are easier to implement and impose smaller computational overhead. The solution guarantees that are obtained with sampling based methods are usually of probabilistic nature, which means that with enough sample points, the probability to find the solution converge to one [10].

Among some popular sampling-based motion planning algorithms are Randomized Potential Field Methods [16], Probabilistic RoadMaps (PRMs) [11, 12] and

Rapidly-exploring Random Trees (RRTs) [10, 13, 14].

PRMs are multiple-query methods and perform well in high-dimensional state space. But computing a roadmap in advance may be infeasible. RRTs are single-query counterparts of PRMs.

It is important to note that these algorithms usually involve robot models with trivial dynamics. In particular, robots are assumed to be points and it is suggested that two spatially nearby configurations can be connected by a straight line [10]. For most ground vehicles, legged or wheeled, such motion is infeasible, since they cannot move at an arbitrary direction. Thus, motion planning for the OctoRoACH resembles more the type of planning problems for nonholonomic systems.

For the problem set in continuous space and time, we find significant prior work [25], [26], [27], [28], [29], [30]. One particularly interesting problem formulation is the Dubin's car model [20]. In the context of this model it was discovered that given the configuration of the initial and final points, the planar curve with minimum length that connects these two points while obeying the model's prescribed velocity orientation and radius of curvature constraints can be found. In order to study the slow, quasi-static planar motion of OctoRoACH, a newly kinematic abstraction is developed in [23], which has the form of a four-bar mechanism (Figure 1.5). A four-bar is a mechanism composed of four bodies (bars) which are connected in a kinematic chain by four joints.

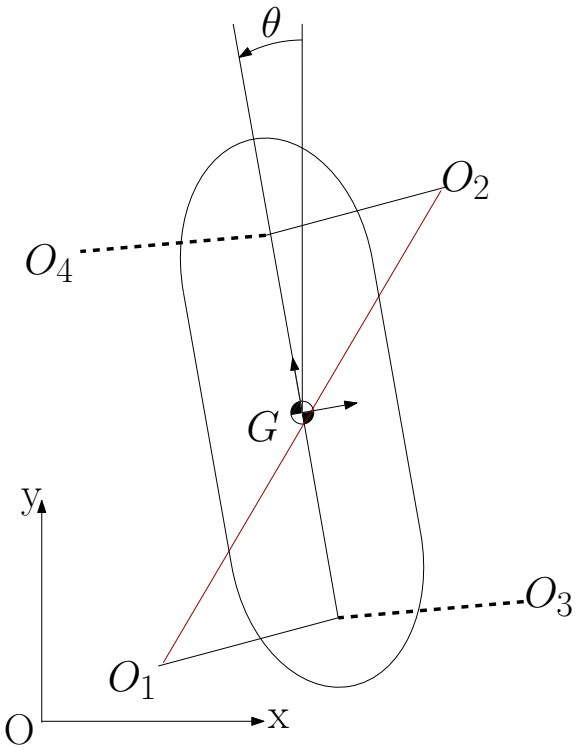


Figure 1.5: The switching four bar mechanism that will serve as an abstract kinematic model for the OctoRoACH

Existing motion planning methods either assume an omnidirectional point robot or use motion primitives (elementary, basic, motion behaviors like the ones in Dubin's model) to compose a motion plan. The Octoroach cannot be viewed as an omnidirectional point since it is subject of nonholonomic constraints. If we establish motion primitives for the OctoRoACH, then it is reasonable to expect that some of the powerful existing motion planning methodologies for nonholonomic systems, may apply. For planar motion, the most basic motion primitives that can be envisioned for the OctoRoACH is going straight and turning left or right.

Actually, motion primitives discretize the space of action trajectories [10].

Take a video game as an example. If we have a database of basic movements, such as kicking, punching, jumping, etc., we can use them to make different fighting maneuvers for our character. These basic movements can be thought of as motion primitives. Motion primitives do not need to be executed in a fixed time interval.

We can define a more general kind of discretization to handle such models. The discrete-time model can be used to formulate a discrete-time state transition equation of the form

$$x_{k+1} = f_d(x_k, u_k) \tag{1.1}$$

in which $x_k = x((k-1)\Delta t), x_{k+1} = x(k\Delta t)$, and u_k is the action in $\mathbf{U_d}$ that is applied from time $(k-1)\Delta t$ to time $k\Delta t$. $f_d : \mathbf{X} \times \mathbf{U_d} \to \mathbf{X}$ that represents an approximation to f, the original state transition function. Every constant action $u \in \mathbf{U_d}$ applied over Δt can be considered as a motion primitive.

With this, motion plans can now be described as a concatenation of a variety of well-defined motion primitives.

Motion primitives have been widely studied to solve motion planning problems. A maneuver-based method to solve motion-planning problems for nonlinear system with symmetries, such as helicopters, mobile robots and autonomous vehicles has been developed in [33]. A novel computational and modeling framework and related algorithms using motion primitives to steer underactuated, nonholonomic mechanical systems is found in [34]. Planning for vehicles based on discretization of their workspace induced by motion primitives has been performed in [35], [36], [37], [38], in which complex planning computation is reduced while satisfying motion constraints.

A similar approach to path planning has been proposed by Brocket and is called motion description language (MDL) [39], [40], [41]. The basic idea in MDL is to use the collections of primitives to form a formal basis for robot programming

and incorporate kinematic and dynamic models of robots in the form of differential equations. References [42], [43] extend the early idea to a version of language, known as MDLe. The simplest element of MDLe is the atom, an vanishing vector field defined on space-time. An atom is a triple of the form $\sigma = (u, \xi, T)$ where u is the state input, ξ is a boolean interrupt function while T denotes the time. Applying the input to the kinetic state machine until the boolean interrupt function becomes zero or until time elapses is called running the atom. A string consists of atoms and carries its own interrupt function and timer are called behaviors. Behaviors can be used to form partial plans, and finally be nested into plans.

This methodology can be applied to complete robot navigation tasks, multi-robot motion control [44], motion planning for nonholonomic robots [42], [43].

1.3 Contribution

The work presented in this thesis has made contributions to motion planning for small multi legged locomotion.

While several planners have been designed for wheeled robots using the Dubin's Car model [31], none such application has been reported in the context of legged robots. The closest work along this line is reported in [47] currently under review. The difference here is that in this thesis, motion planning is performed using Dubin's curves as opposed to a direct incorporation of the OctoRoACH kinematic abstraction in an RRT planner. In order to utilize the Dubin's car model, we need first to discover how we can generate net motions in the model that correspond to straight and curved paths and quantify path curvatures in terms of the model's parameters as well.

Many real world applications require to plan the motion of a robot so that it goes from an initial position A in the workspace to a final desired one, B. This

results in sequences of curves that the robot needs to follow to realize the task. The major contribution of this thesis is that it provides a method that links the high-level specifications regarding the path that the robot needs to follow (curvature, direction of motion etc.) to the low-level controls (touchdown, liftoff angles) of a robot model to realize these paths.

More specifically, the contributions of this thesis are as follows:

1. Find the combinations of model parameters that produce the desired motion primitives, the concatenation of which drives the robot towards a desired location.

2. Quantify the geometry of the produced primitives in terms of curvature, and associate that curvature with the model parameters.

3. Test the hypothesis that the coarse kinematic behavior of this model, as expressed by the average of those motion primitives, can be related to the Dubin's car model.

If this hypothesis is proven correct, then under certain conditions and simplifying assumptions, a large collection of tools available in the motion planning literature for nonholonomic systems can be made availabe potentially for this abstract model, the switching four-bar linkage, which is intended to capture at a high level the quasi-static kinematics of a large range of miniature crawling robots.

This thesis advocates that there is significant evidence in support of this hypothesis. The match we have observed between the Dubin's model solutions and the OctoRoACH model simulations is encouraging, but more analysis is needed to investigate the source of the small path deviations that are observed.

Chapter 2

ROBOT DESCRIPTION AND MODELING

In this chapter, we will give a brief overview of the robot that is at the center of our study of path planning. We present an abstract kinematics model that is to be used later for motion analysis and planning.

2.1 Robot Design

The OctoRoACH robot has a rectangular body 130 mm long and 60 mm wide. Its height is 30 mm and its weight is 35 g. It has eight legs, four on each side. Each side of the legs is independently driven by a DC brushed motor.

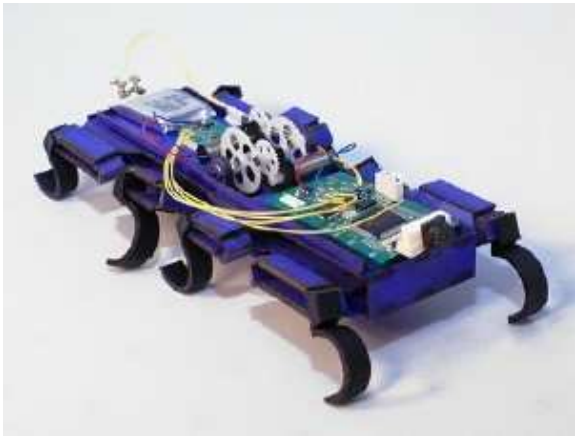
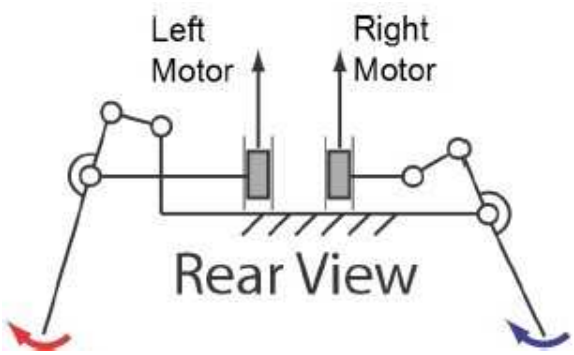


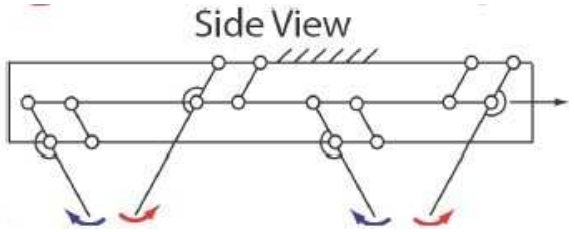
Figure 2.1: The OctoRoACH robot. (Courtesy of A. O. Pullin; reproduced from [8] with the author's permission)

The two sides are not mechanically coupled. The maximum speed the robot can reach is 0.5 m/s with a maximum stride frequency of 25 Hz. The OctoRoACH was designed at the University of California, Berkeley, and manufactured by Motile Robotics, Inc. for the ARL MAST Consortium.

The rear view (Figure 2.2a) of the ideal robot mechanism kinematics indicates that the ab- and adduction of different pairs of legs on each side occurs out of phase when the middle member of the linkage is translated vertically. The side view (Figure 2.2b) of the ideal robot mechanism kinematics shows that protraction and retraction of the legs is controlled by the motion of the middle member in the fore-aft direction [8].



(a) Robot rear view



(b) Robot side view

Figure 2.2: Rear and side view of OctoRoACH (Courtesy of A. O. Pullin; reproduced from [8] with the author's permission)

2.1.1 Fabrication

The transmission mechanism is fabricated using a scaled version of the smart composite microstructures (SCM) process [24]. The SCM method is inexpensive, it can provide a relatively strong, lightweight mechanism which has minimum friction and backlash while keeping the cost low. What is more, the energy absorbing structure and high surface-to-weight ratio it provides enable the robot to survive falls from large heights.

2.1.2 Leg Design

The leg is designed to have a C shape, similar to RHex [22]. This C shape has three primary advantages [7].

First, for primary loading, the C leg would have material flexure different from the material compression in straight vertical leg, resulting in lower vertical stiffness. Second, in case of obstacle climbing, the C section can fold backwards when colliding with an object which is taller than the swing clearance height enabling the leg to move forward and step on the obstacle. Finally, a typical straight leg has isolated point contact, whereas the C shape leg, has a rolling contact with the ground.

2.2 Robot Model

2.2.1 OctoRoACH motion behavior abstraction

The kinematic model we describe here is proposed by Konstantinos Karydis and the kinematic analysis is collaborated between Konstantinos Karydis and me in [23].

The configuration variables of this model consist of the Cartesian position (x, y) of the centroid G with respect to an inertial coordinate frame Oxy, and the intersection angle between the longitudinal body-frame axis and y-axis of the inertia

frame, which is called θ (heading angle). Figures 2.4a and 2.4b show the foot fall pattern of the robot. The robot implements an alternating tetrapod gait, assuming a relatively slow quasi-static motion that excludes flight phases which are typical of running. The solid circles in Figure 2.4 represent leg tips in contact with the ground. As shown in Figures 2.4a and 2.4b, the legs 1,2,3,4 move simultaneously while legs 5,6,7,8 form the other quadruped [23]. An ideal horizontal robot model is shown in Figure 2.3.

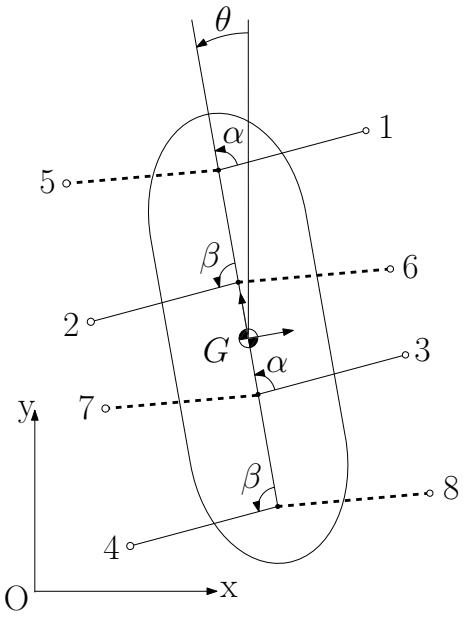


Figure 2.3: The OctoRoACH robot model [23].

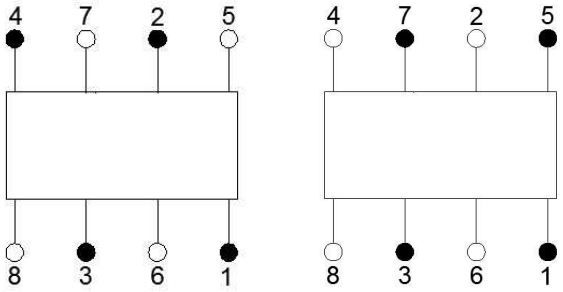


Figure 2.4: OctoRoACH foot fall pattern [23].

In Figure 2.3 we see that, leg 1 and leg 3 of the model have the same angle α with respect to the longitudinal body-frame axis. Similarly, leg 2 and leg 4 rotate with some angle $\beta \neq \alpha$ relative to the longitudinal body-frame axis. Due to the mechanical restriction, the angles of ipsilateral (pertaining to the same side of the body) legs of each quadruped corresponding to the longitudinal body-frame axis are always the same. In other words, the legs in this model have the same angular velocity and motion phase. The same happens for the other quadruped. Because these two legs on each side move in synchrony and have the same phase angle, they are further abstracted into a single virtual leg. The virtual legs connect to imaginary joints on the body-frame longitudinal axis, and generate the same displacement and angle motion as the pair of homolateral legs. This reduction gives rise to an abstraction of the robot kinematics in the form of a tetrapod. This kinematic abstraction obess the following three assumptions [23].

Assumption 1. *Only one pair of the virtual legs is in contact with the ground propelling the robot.*

Assumption 2. *Once a pair of virtual legs turns active, the ground contact points remain fixed until the other pair turns active.*

Assumption 3. *The legs in each pair begin and complete the step at the same time instant.*

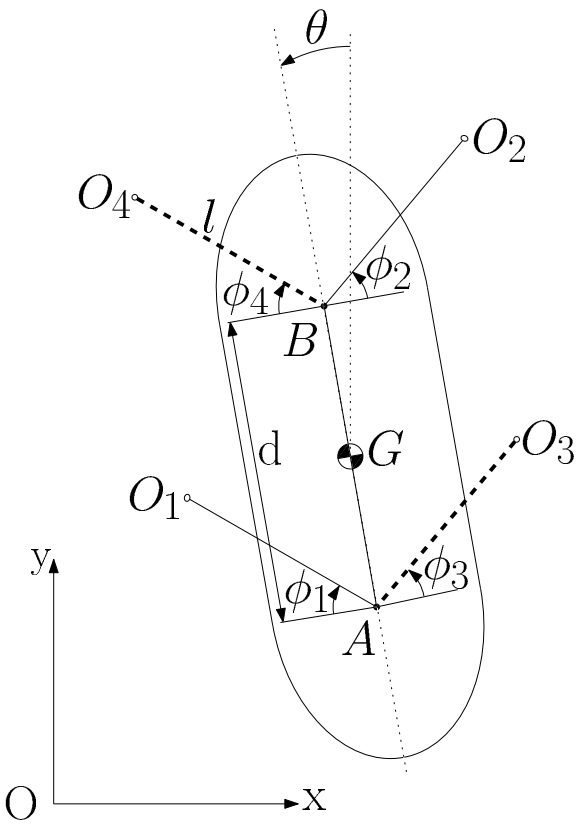


Figure 2.5: A kinematic tetrapod abstraction of the kinematics of the OctoRoACH in the horizontal plane [23].

Based on the above assumptions, the kinematic robot model during quasi-static motion can be viewed as a switching four-bar linkage. Figure 2.6 shows the situation when the right pair is active. When this pair of legs reaches the end of their motion range, the left pair becomes active, and the cycle is repeated.

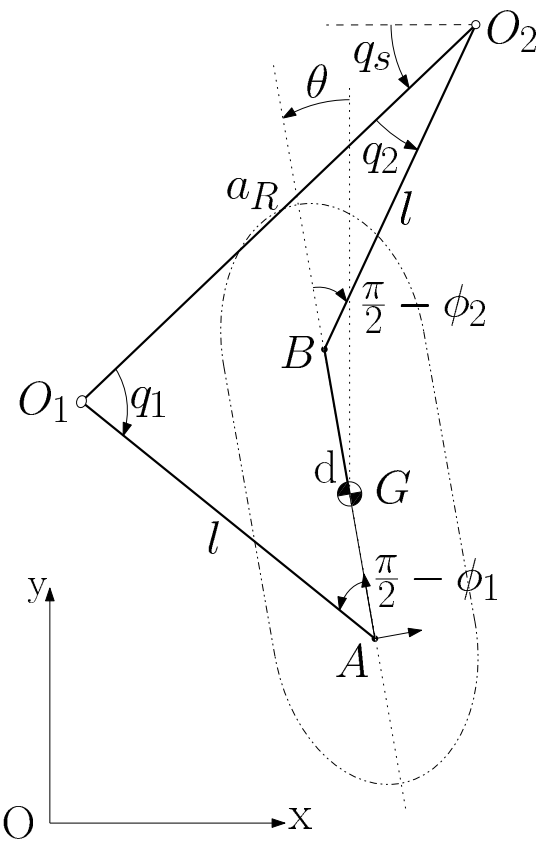


Figure 2.6: Abstracted model for OctoRoACH [23].

2.2.2 OctoRoACH kinematic analysis

When one pair is active-say the right one, for the sake of argument-the active four bar linkage represents a one degree of freedom mechanism whose motion is determined by ϕ_1. Similarly, for the left pair, ϕ_3 determines the motion of the other tetrapod. One tetrapod (four-bar linkage) becomes the mirror image of the other. For this reason we analyze here-one pair- (the right pair), which is assumed hinged at the fixed ground points, O_1, O_2. The position vector-loop equation is [23]

$$\mathbf{R}_{AO_1} + \mathbf{R}_{O_1O_2} - \mathbf{R}_{AB} - \mathbf{R}_{BO_2} = 0. \tag{2.1}$$

Using the exponential representation of the participating vectors in equation (2.1), we can obtain [23]

$$le^{j(\pi-\phi_1)} + a_R e^{j(\phi_1-\pi/2+\theta_1)} - de^{j\pi/2} - le^{j(\phi_2)} = 0. \tag{2.2}$$

Then, by setting $e^{\pm j\theta} = \cos\theta \pm j\sin\theta$, and separating real and imaginary parts [23]

$$\begin{aligned} l\sin\phi_1 + a_R\sin(q_1-\phi_1) - d - l\sin\phi_2 = 0 \\ l\cos\phi_1 + l\cos\phi_2 - a_R\cos(q_1-\phi_1) = 0. \end{aligned} \tag{2.3}$$

Note that d is the distance between hip joints A and B, l is the length of the virtual leg. Distance d is set to 13 cm, matching the length of the torso of the physical OctoRoACH platform. Parameter l is set to 3 cm, which is half of the robot's width. From Figure 2.6, the relation between the spatial coordinates of A, B and G is as follows:

$$\begin{aligned} x_A &= x_G - d/2\cos(\theta+\pi/2) \\ y_A &= y_G + d/2\sin(\theta+\pi/2) \\ x_B &= x_G + d/2\cos(\theta+\pi/2) \\ y_B &= y_G - d/2\sin(\theta+\pi/2). \end{aligned} \tag{2.4}$$

Note that segment O_1A has length l, and is oriented $\pi/2 - \phi_1 + \theta$ relative to the y axis, while segment O_2B has length l, and is oriented $\pi/2 - \phi_2 - \theta$ relative to the y axis. Hence, the spatial coordinates of O_1, O_2 can be set

$$\begin{aligned} x_{O_1} &= x_A - d/2\sin(\theta+\pi/2-\phi_1) \\ y_{O_1} &= y_A + d/2\cos(\theta+\pi/2-\phi_1) \\ x_{O_2} &= x_B + d/2\sin(\pi/2-\theta-\phi_2) \\ y_{O_2} &= y_B - d/2\cos(\pi/2-\theta-\phi_2). \end{aligned} \tag{2.5}$$

The single degree of freedom in the system, ϕ_1, appears explicitly in the above expressions. Let angular velocity $\dot{\phi_1}$ be constant during the whole step. Hence, ϕ_1 at any given time t can be expressed as

$$\phi_1 = \phi_1^0 + \dot{\phi_1} t. \tag{2.6}$$

Since this is a one degree of freedom system, angle ϕ_2 can be expressed as a function of ϕ_1, and generally written as

$$\phi_2 = f(\phi_1). \tag{2.7}$$

Along similar lines, after explicitly solving (2.3), θ can be expressed as a function of ϕ_1 and ϕ_2, in the form

$$\theta = g(\phi_1, \phi_2). \tag{2.8}$$

Before moving forward, we would like to define some variables first.

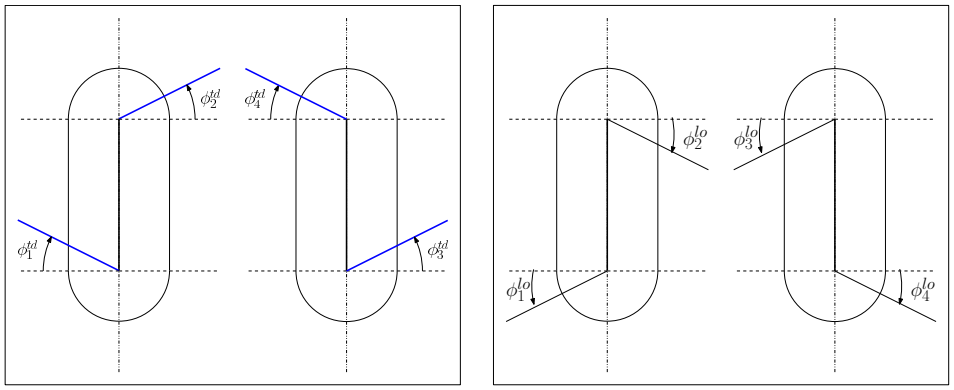
1. ϕ^{td} (*touchdown* angle), is the angle that both legs in a pair touch the ground, as Figure 2.7a shows.

2. ϕ^{lo}(*liftoff* angle), is the angle that both legs in a pair lift off the ground, as Figure 2.7b shows.

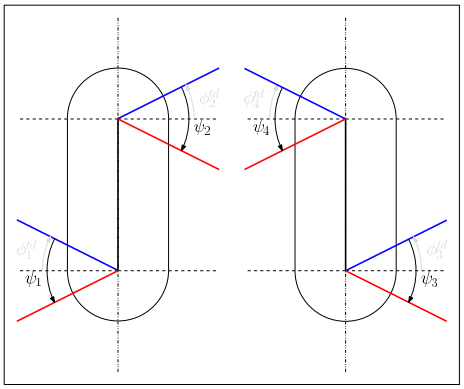
3. ψ (*sweep* angle), the range of values for each ϕ_i, as Figure 2.7c shows.

$$\psi_i = \left|\phi_i^{td}\right| + \left|\phi_i^{lo}\right|. \tag{2.9}$$

These angles are viewed as model parameters, and will be used to generate motion primitive in the next chapter.



(a) Touchdown angle (b) Liftoff angle



(c) Sweep angle

Figure 2.7: Angle input of OctoRoACH model [23].

Since we have obtained the relation between the position of A, B, G, O_1, O_2 and the values at angles of ϕ_i, θ, the steps of the algorithm that calculates the angles of the right pair of legs in this abstract model is as follows.

Algorithm 2.1 Solve positions based on motion primitive parameters for right pair

1 Specify motion primitive parameters ϕ_1^{td}, ϕ_2^{td}, ϕ_1^{lo}, ϕ_2^{lo}.

2 Give the initial θ, x_G, y_G.

3 Define points A, B according to the value of θ, x_G, y_G based on equation (2.4).

4 Define the fixed points O_1, O_2 based on equation (2.5).

5 Calculate a_R and q_s.

6 Let ϕ_1 change linearly, solve for ϕ_2 and θ as a function of ϕ_1.

7 Calculate the position of point A, B and G.

Utilizing the kinematic model, we are able to analyze the kinematics for right pair. Since the right and left pairs are just the mirror image of each other, we can plot the trajectory for one single step and make minor adjustments to compute the evolution during the other step. Temporally concatenating these single steps, the primitive elementary motions for the model can be generated.

Chapter 3

PRIMITIVE MOTIONS

Given the kinematics analysis performed in the previous chapter, we now use it to discover how to generate gross motions for the abstract model that resemble straight and curved paths. This part of work is collaborated between me, Konstantinos Karydis, Dr. Poulakis and Dr. Tanner in [23]. I put forward the method to introduce asymmetry between touchdown and liftoff angle to produce curved path.

3.1 One Stride Computation Procedure

At the end of Chapter 2, we obtained a procedure to analyze the kinematics for right leg-pair. Since left leg-pair just mirrors the right pair, we can specify the computation procedures for one stride as follows.

Based on this procedure, we can plot the horizontal path generated for the geometric center of the mechanism in one single stride. Figure 3.1 shows an example of this one stride path for the Octoroach kinematic abstraction.

Algorithm 3.1 Simulation Procedure for one step

1 Specify motion primitive parameters $\phi_1^{td}, \phi_2^{td}, \phi_1^{lo}, \phi_2^{lo}$.
2 Give the initial θ, x_G, y_G in right pair.
3 Calculate points A, B according to the value of θ, x_G, y_G based on (2.4).
4 Calculate the fixed points O_1, O_2 based on (2.5).
5 Calculate a_R and q_s.
6 Let ϕ_1 change linearly, solve for ϕ_2 and θ as a function of ϕ_1.
7 Calculate the position of point A, B and G.
8 Specify motion primitive parameters $\phi_3^{td}, \phi_4^{td}, \phi_3^{lo}, \phi_4^{lo}$.
9 Give the initial θ, x_G for left pair.
10 Calculate points A, B according to the value of θ, x_G,y_G based on (2.4).
11 Calculate the fixed points O_3, O_4.
12 Calculate a_R and q_s.
13 Let ϕ_3 change linearly, solve for ϕ_4 and θ as a function of ϕ_3
14 Calculate the position of point A, B and G.

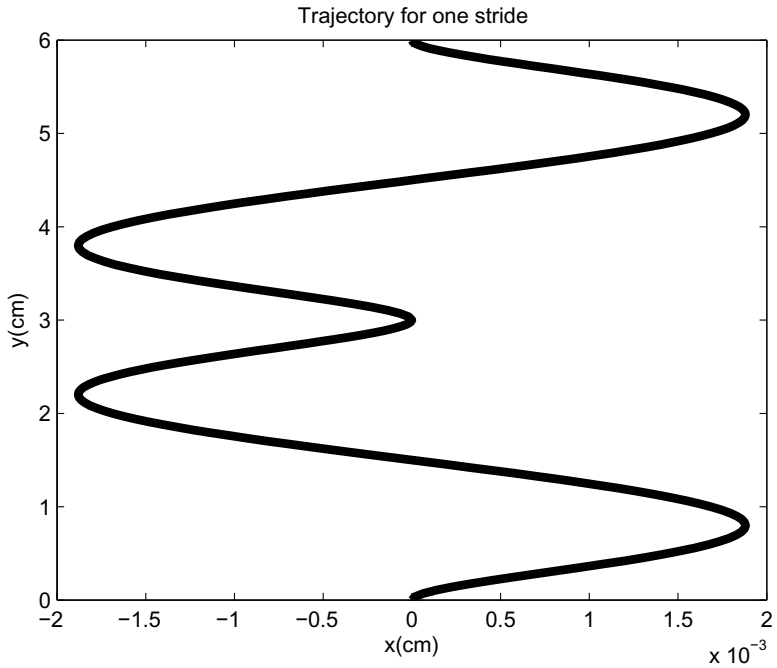


Figure 3.1: One stride trajectory for Octoroach, $\phi_i^{td} = \phi_i^{lo} = \pi/6, (i = 1, 2, 3, 4)$. In order to distinctly show the path, x and y axis are not in the same scale.

3.2 Primitive Parametrization

The analysis of kinematics of the abstract model in a single step forms the basis for the definition of primitive motions.

Different pair parameterizations in terms of touchdown and liftoff angles produce different postures for the model at the end of each step, as Figure 3.2 shows.

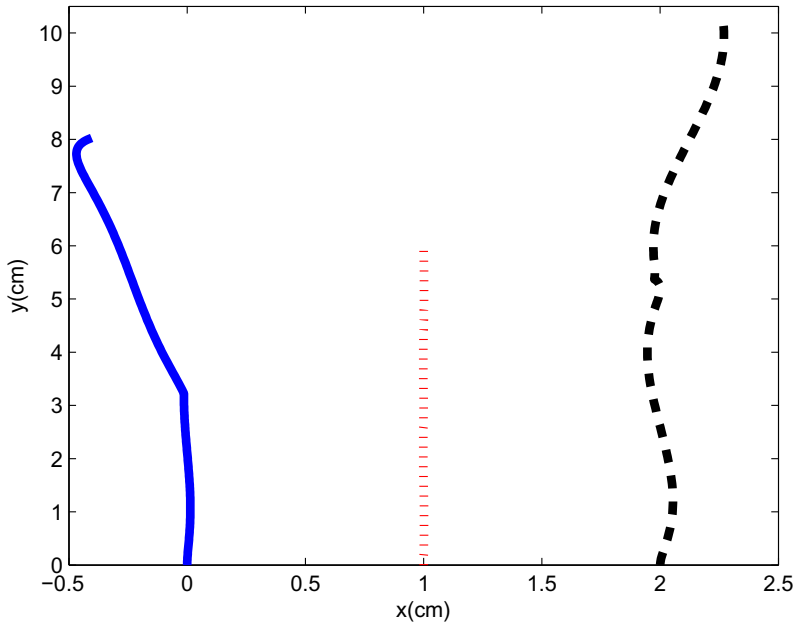


Figure 3.2: One stride path for different parameterizations. Blue solid line corresponds to $\phi_i^{td} = \pi/4 (i = 1, 2, 3, 4), \phi_{1,2}^{lo} = \pi/4 - 0.5, \phi_{3,4}^{lo} = \pi/4 + 0.5$, red dotted line corresponds to $\phi_i^{td} = \phi_i^{lo} = \pi/6, (i = 1, 2, 3, 4)$, black dashed line corresponds to $\phi_i^{td} = \pi/3, (i = 1, 2, 3, 4), \phi_{1,2}^{lo} = \pi/3 + 0.2, \phi_{3,4}^{lo} = \pi/3 - 0.2$. Note that angular velocity $\dot{\phi_1}$ and $\dot{\phi_3}$ remain constant for each case. In order to show the displacement in x axis clearly, x axis and y axis are not in the same scale.

The idea is to concatenate left-right pair step motions, which have been identically parameterized, repeat this cycle for a specified number of iterations and thus form one motion primitive. In Figure 3.2, the blue solid line makes a left turn, the red dashed line goes straight while the black dotted line turns right. Our hypothesis now is that if we concatenate this left-right pair cycles into sufficiently long strings, we can produce three motion primitives (go straight, turn left, turn right).

3.2.1 Straight line Paths

In order to generate a straight line path, we need to set the leg angle model parameters as follows [23].

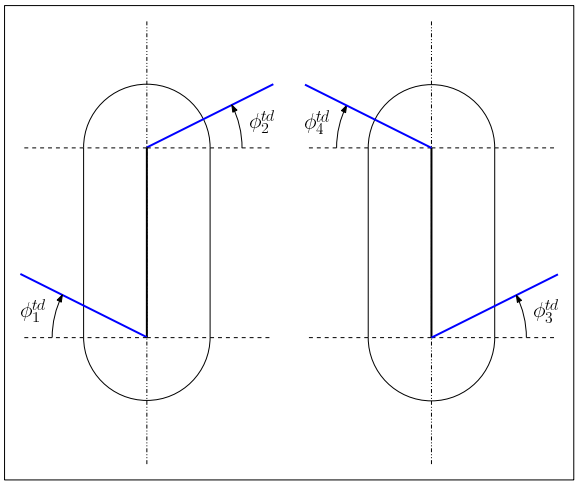
$$\begin{aligned} \phi_{SL}^{td} = \phi_1^{td} = \phi_2^{td} = \phi_3^{td} = \phi_4^{td} \\ \phi_{SL}^{lo} = \phi_1^{lo} = \phi_2^{lo} = \phi_3^{lo} = \phi_4^{lo} = \phi_{SL}^{td}. \end{aligned} \tag{3.1}$$

where the subscript SL is used to denote straight path configuration.

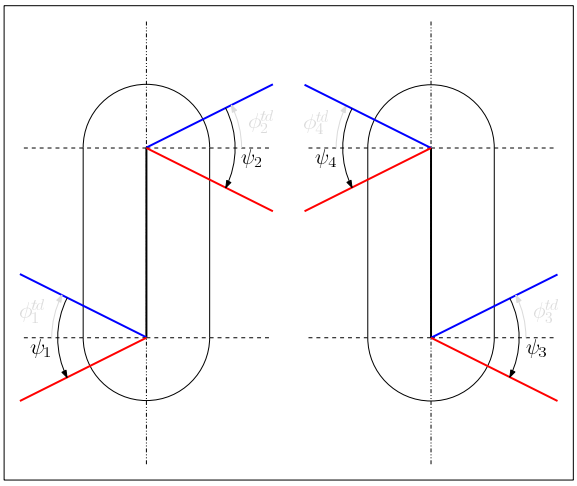
The configurations in (3.1) suggest that to produce a straight line, the touchdown and liftoff angle should be symmetric, which means that ϕ^{td} and ϕ^{lo} are equal in magnitude and opposite in sign. Substitute (3.1) into (2.9), we can find the equality of sweep angle of each leg.

$$\psi_{SL} = \psi_1 = \psi_2 = \psi_3 = \psi_4 = \left|\phi_{SL}^{td}\right| + \left|\phi_{SL}^{lo}\right| = 2\phi_{SL}^{td}. \tag{3.2}$$

Figure 3.3a and Figure 3.3b show the initial and straight line path configurations, respectively.

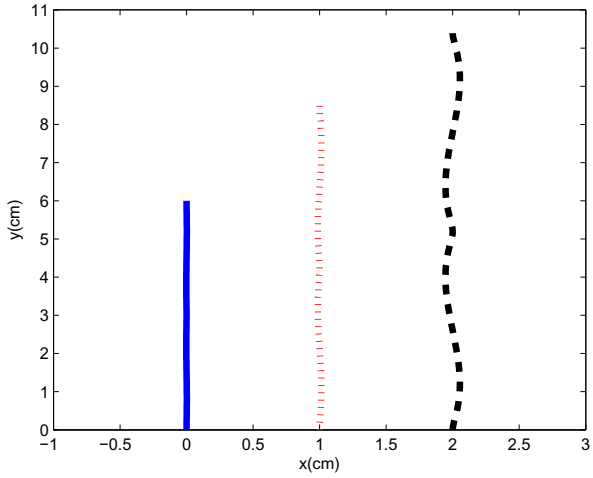


(a)

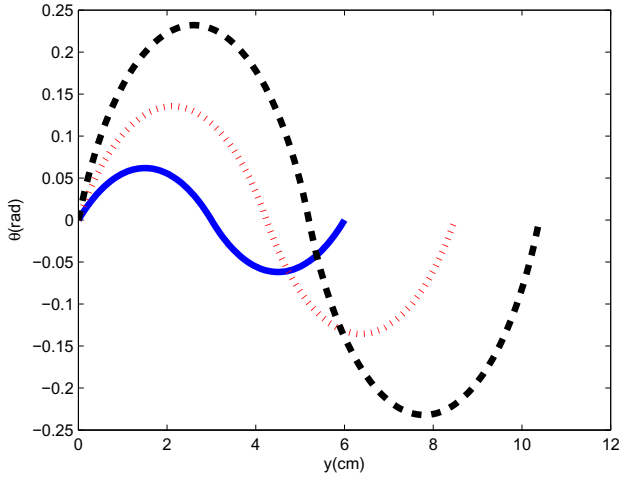


(b)

Figure 3.3: Straight line model configuration: lift off angles are equal in magnitude and opposite in sign compared to touchdown angles [23].



(a) One cycle path



(b) Head angle

Figure 3.4: Evolution of model's state for the straight line path. Blue solid line corresponds to $\phi_{sl}^{td} = \pi/6$, red dotted line corresponds to $\phi_{sl}^{td} = \pi/4$, black dashed line corresponds to $\phi_{sl}^{td} = \pi/3$. Angular velocities $\dot{\phi}_1$ and $\dot{\phi}_3$ remain constant for each case. (a) Increasing ϕ_{sl}^{td} enables the robot to propel itself for a longer distance in y axis. (b) Increasing ϕ_{sl}^{td} causes the robot's in heading angle to oscillate more.

Figure 3.4 demonstrates how the state of the model evolves over time for the straight line case. All plots correspond to one cycle (one left-right pair switching), with the same initial geometric center position and body orientation for all runs. As we increase the value of sweep angles (which is equivalent to increasing touch-down angle) takes longer steps covering more ground in each step while the oscillation of the heading angle becomes more pronounced.

3.2.2 Curved Paths

Next, we would like to investigate the combination of model parameters that generates curved paths. Figure 3.4b shows that the heading angle, θ, is cyclic in the straight line case. This is due to the symmetry of leg sweep angle. Introducing asymmetry into the sweep angle, we compute new paths and observe the evolution of the heading angle (Figure 3.5).

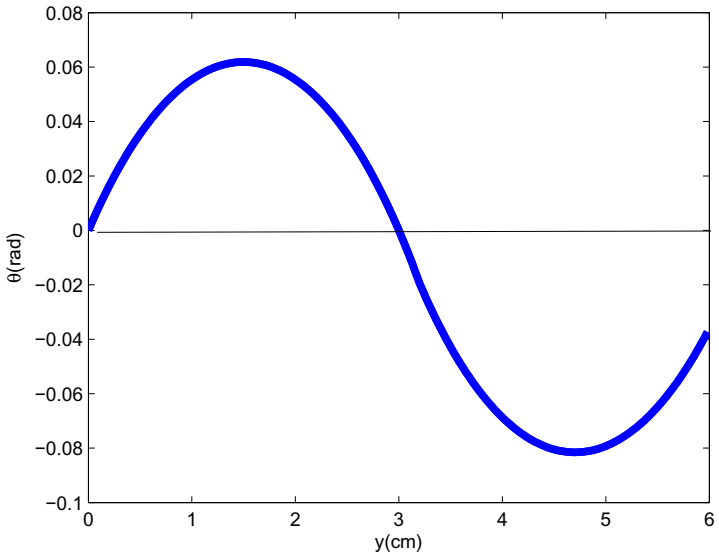


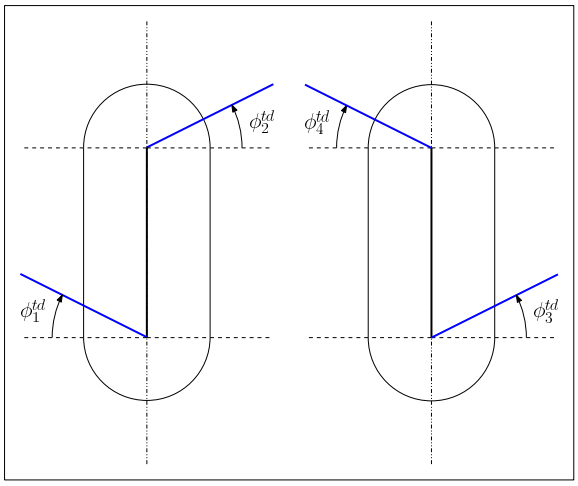
Figure 3.5: Head angle for asymmetric sweep angles

Figure 3.5 shows that the heading angle is not symmetric anymore and this offset in orientation adds up from stride to stride, to generate a curved path.

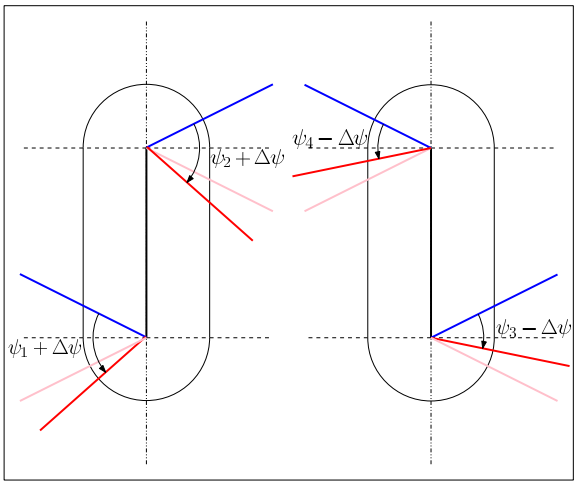
The asymmetry in the sweep angle of the leg is denoted

$$\Delta\psi = |min\{\psi_1, \psi_2\} - min\{\psi_3, \psi_4\}| \,. \tag{3.3}$$

Figure 3.6a and Figure 3.6b show how the asymmetry in the sweep angle manifests itself in the case where the model turns to the right, if we change the sign of $\Delta\psi$ in Figure 3.6b, the model will turn counter-clockwise (left).



(a)



(b)

Figure 3.6: Model parameter configuration for curved path generation [23]. Courtesy of Konstantinos Karydis; reproduced from (Karydis et al., 2014) with the author's permission.

The evolution of the geometric center of the model when generating clockwise curves is depicted in Figure 3.7. All paths shown contain twenty strides and start

at the same initial conditions for the geometric center and body orientation. We see that increasing $\Delta\psi$ generates paths with smaller radius of curvature.

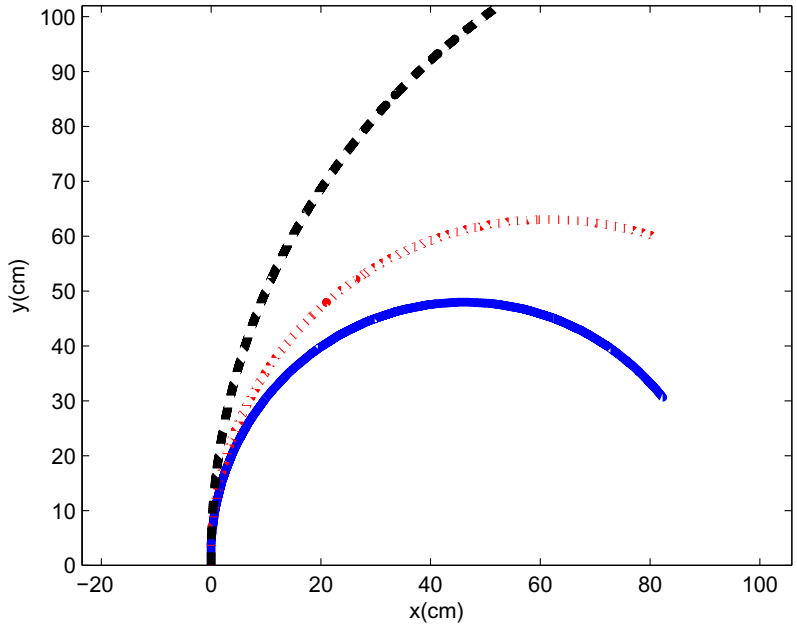


Figure 3.7: Clockwise curved paths. All paths contain twenty strides and have touchdown angles, $\phi_i^{td} = \pi/6, (i = 1, 2, 3, 4)$. The blue solid line corresponds to $\Delta\psi = 2\pi/9$, the red dotted line corresponds to $\Delta\psi = \pi/6$, and the black dashed line corresponds to $\Delta\psi = \pi/12$. The angular velocities $\dot{\phi}_1$ and $\dot{\phi}_3$ remain constant for each case. Increasing the sweep angle asymmetry, $\Delta\psi$, will increase the path curvature.

In this chapter, we utilized the kinematic model developed in Chapter 2, to define a small number of motion primitives. Specific model parameter combinations have been identified in order for the model to generate straight line and curved paths. The key insight is that when the legs sweep along different arcs between the left and

the right leg pair, i.e., there is asymmetry in the sweep angles in the two steps, then the model's heading angle ends the stride with an offset. The sweep angle asymmetry is denoted $\Delta\psi$, and it was observed that larger $\Delta\psi$ produces bigger path curvature. Our next goal is to quantify path curvatures in terms of the model's parameters, so that we can link the model's behavior to that of the Dubin car model.

The definition of primitive motions is a product of joint work with Konstantinos Karydis. The individual contribution is in introducing asymmetry between touchdown and liftoff angle as a way to generate curved paths in the model.

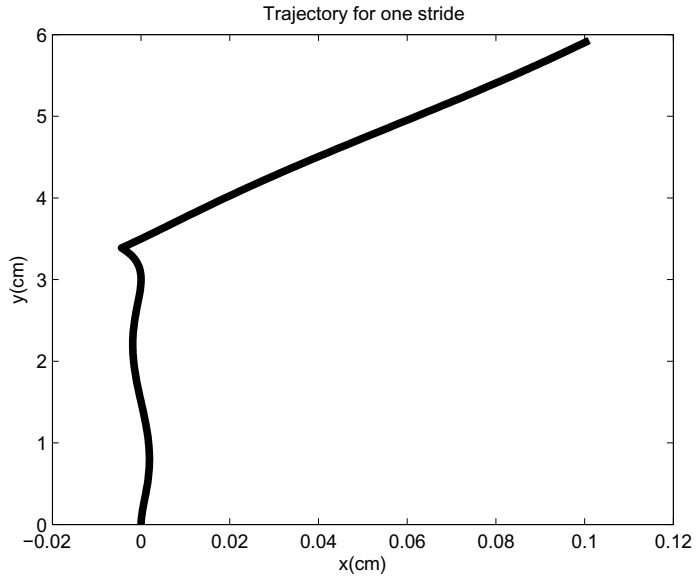
Chapter 4

CURVATURE CALCULATION

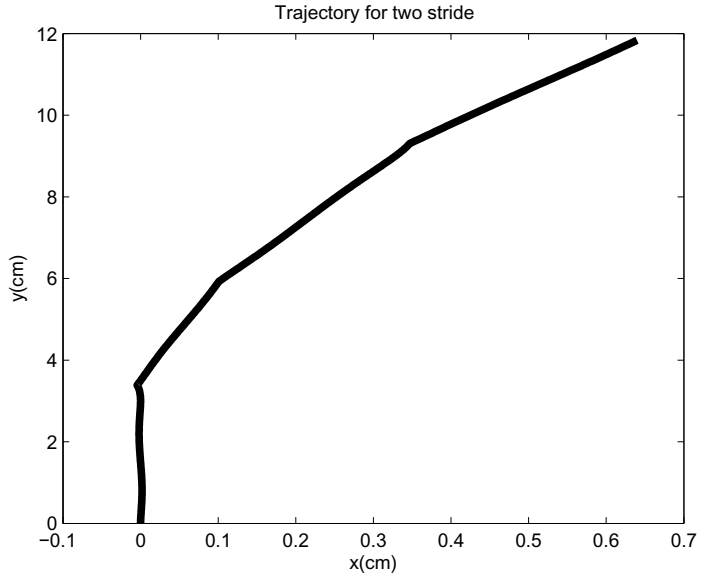
Now we move on to my own work. To use the Dubin's Car model for planning, we need first to relate the desired path curvature to the parameters of the switching four-bar model of the OctoRoACH, such as touchdown and liftoff angles, that realize these paths. To express the curvature of paths, we draw from differential geometry of curves and surfaces [32]. A challenge we face is that the model's paths are not smooth, but exhibit kinks and corners at points where left and right pairs switch from one to another, as we can see in Figure 4.1. These corners give rise to discontinuities in the trajectory of velocity and acceleration, as Figure 4.2 shows. Since velocity and acceleration are discontinuous at switching points between steps, the local expression of the curvature

$$k = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{\frac{3}{2}}} \tag{4.1}$$

is discontinuous at these switching points as well.



(a) Path during first step



(b) Path during the first two steps

Figure 4.1: Planar paths for two consecutive steps. x axis and y axis are not in the same scale to show the corner distinctly. For these curved paths, the touchdown angle was $\phi^{td} = \pi/6$ and the sweep asymmetry angle was $\Delta\psi = 0.2$.

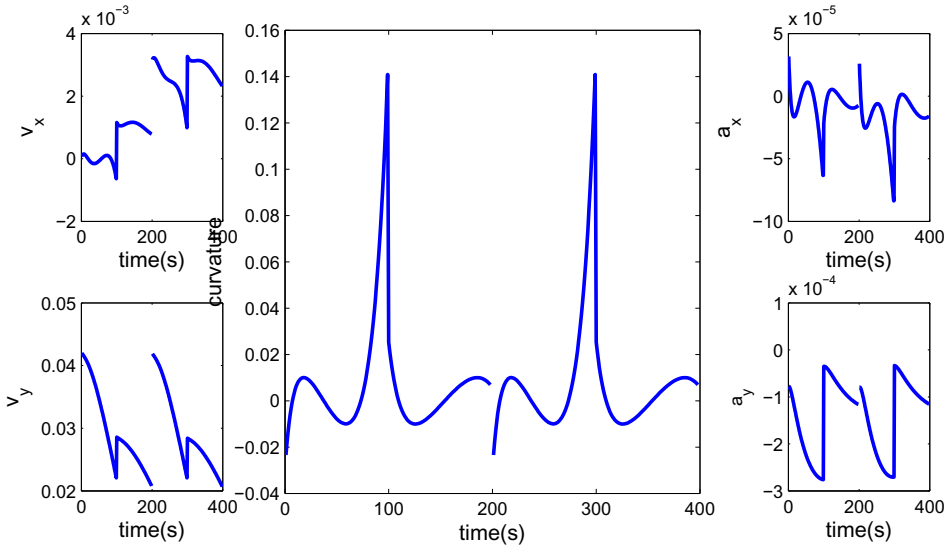


Figure 4.2: Velocity and acceleration along two strides (note the discontinuity between steps). For this simulation, the touchdown angle was $\phi^{td} = \pi/6$, and the sweep angle asymmetry was $\Delta\psi = 0.2$.

Due to the discontinuities, it is impossible to directly use (4.1) to obtain the radius of curvature for the path. The next section describes an approach to address this challenge.

4.1 Gauss-Bonnet Theorem

Before presenting the Gauss-Bonnet Theorem, we need to introduce some terminology.

Definition (Regular surface [32]). *A subset $S \subset R^3$ is a regular surface if, for each $p \in S$, there exists a neighborhood V in R^3, a neighborhood U in R^2 and a map $x : U \to V \cap S$ of $U \subset R^2$ onto $V \cap S \subset R^3$ such that (Figure 4.3)*

1. *x is differentiable. This means that if we write*
   $x(u,v) = (x(u,v), y(u,v), z(u,v)), (u,v) \in U$
   the functions $x(u,v), y(u,v), z(u,v)$ have continuous partial derivatives of all orders in U.

2. *x is a homeomorphism. Since x is continuous by condition 1, this means that x has an inverse $x^{-1} : V \cap S \to U$ which is continuous; that is, x^{-1} is the restriction of a continuous map $F : W \subset R^3 \to R^2$ defined on an open set W containing $V \cap S$*

3. *(The regularity condition.) For each $q \in U$, the differential $dx_q : R^2 \to R^3$ is one to one.*

The mapping x is called a parameterization or a system of (local) coordinates in (a neighborhood of)p. The neighborhood $V \cap S$ of p in S is called a coordinate neighborhood.

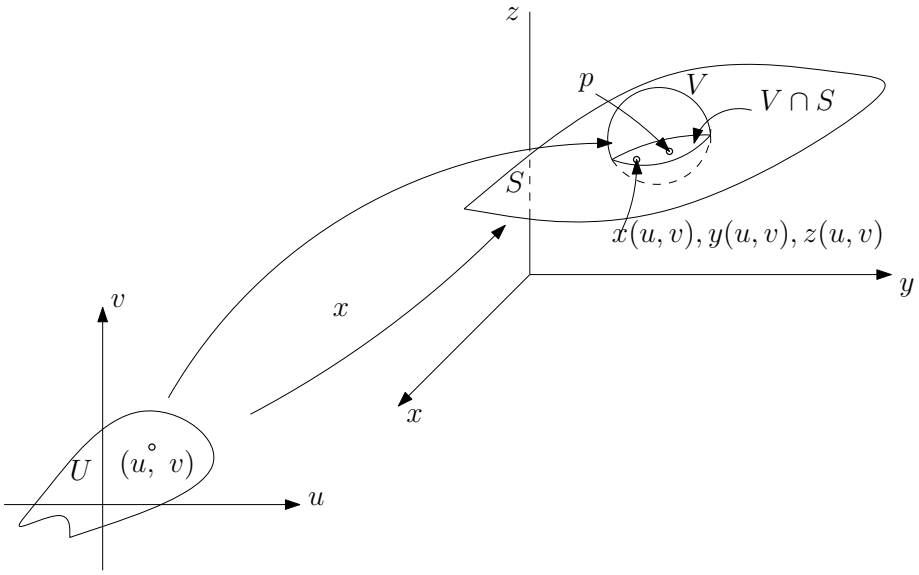


Figure 4.3: Regular surface [32]

Definition. *[32] Let $\alpha : [0, l] \to S$ be a continuous map from the closed interval $[0, l]$ into the regular surface S. If*

1. *$\alpha(0) = \alpha(l)$*
2. *$t_1 \neq t_2$, $t_1, t_2 \in [0, l]$, implies that $\alpha(t_1) \neq \alpha(t_2)$*
3. *There exists a subdivision $0 = t_0 \leq t_1 \leq \cdots \leq t_k \leq t_{k+1} = l$, of $[0, l]$ such that α is differentiable and regular in each $[t_i, t_{i+1}], i = 0, \ldots, k$,*

then α is a simple, closed, piecewise regular, parameterized curve.

Condition 3 guarantee curve α fails to have a well-defined tangent line only at a finite number of points.

The points $\alpha(t_i), i = 0, \ldots, k$ are called *vertices* of α, and the signed angle φ_i which determines the angle from $\alpha'(t_i - 0)$ to $\alpha'(t_i + 0)$ is called the external angle at the vertex $\alpha(t_i)$. The sign of φ_i is determined by the right hand rule, for $-\pi < \varphi_i < \pi$. Figure 4.4 shows the vertices and the external angles for them.

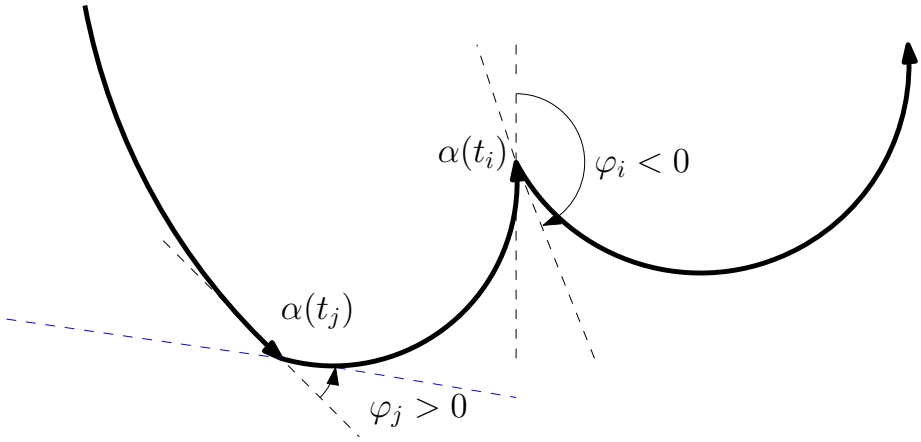


Figure 4.4: Vertices and external angles [32]

Definition (Jacobian)**.** *Given a function $f : R^n \to R^n$ with $y = f(x)$, written explicitly as*

$$\begin{cases} y_1 = f_1(x_1, x_2, \cdots, x_n) \\ y_2 = f_2(x_1, x_2, \cdots, x_n) \\ \vdots \\ y_n = f_n(x_1, x_2, \cdots, x_n) \end{cases}$$

the Jacobian matrix, sometimes simply called "the Jacobian" [45] is defined by

$$J(x_1, \cdots, x_n) = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} \cdots \frac{\partial y_1}{\partial x_n} \\ \vdots \ddots \vdots \\ \frac{\partial y_n}{\partial x_1} \cdots \frac{\partial y_n}{\partial x_n} \end{bmatrix}$$

.

Definition (Orientable surface [46])**.** *A regular surface S is called orientable if it is possible to cover it with a family of coordinate neighborhoods in such a way that if a point $p \in S$ belongs to two neighborhoods of this family, then the change of coordinates has positive Jacobian at p. The choice of such a family is called an orientation of S, in this case, is called oriented. If such a choice is not possible, the surface is called nonorientable.*

Definition (Patch/Local Surface [46])**.** *A* ***patch*** *or* ***local surface*** *is a differentiable mapping*

$\mathbf{x} : \mathcal{U} \to R^n$

where $\mathcal{U}$ is an open subset of R^2.

Definition (Injective Patch [46]). *An **injective patch** is a patch such that* $\mathbf{x}(u_1, v_1) = \mathbf{x}(u_2, v_2)$ *implies that* $u_1 = u_2$ *and* $v_1 = v_2$ *for all* (u_1, v_1) *and* (u_2, v_2).

Definition (Unit Normal Vector Field/Surface Normal [46]). *For an injective patch:* $\mathbf{x} : \mathcal{U} \to R^3$ *the **unit normal vector field** or **surface normal** **U** is defined by*

$$\mathbf{U}(u, v) = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\|\mathbf{x}_u \times \mathbf{x}_v\|}(u, v) \tag{4.2}$$

at those points $(u, v) \in \mathcal{U}$ *at which* $\mathbf{x}_u \times \mathbf{x}_v$ *does not vanish.*
$\mathbf{x}_u(u, v) = \frac{\partial x_1}{\partial u}(u, v) + \frac{\partial x_2}{\partial u}(u, v) + \frac{\partial x_3}{\partial u}(u, v)$
$\mathbf{x}_v(u, v) = \frac{\partial x_1}{\partial v}(u, v) + \frac{\partial x_2}{\partial v}(u, v) + \frac{\partial x_3}{\partial v}(u, v)$
are the partial derivatives of $\mathbf{x}$ *with respect to* u *and* v.

Definition. *[46] Let* $\mathbf{W}$ *be a differential vector field on an open subset* $\mathcal{U} \subset R^n$, *and let* $\mathbf{v_p}$ *be a tangent vector to* R^n *at* $\mathbf{p} \in \mathcal{U}$. *Then the **derivative** of* $\mathbf{W}$ *with respect to* $\mathbf{v_p}$ *is the tangent vector* $\mathbf{D_v W} \in R^n_p$ *given by*

$$D_v W = \widetilde{W}(p + tv)'(0)_p = \lim_{x \to 0} \frac{\widetilde{W}(p + tv) - \widetilde{W}(p)}{t}|_p. \tag{4.3}$$

To measure how a regular surface M bends in R^3, one way is to estimate how the surface normal $\mathbf{U}$ varies from point to point. This can be done by defining the shape operator.

Definition (Shape Operator [46]). *Let* $M \subset R^3$ *be a regular surface, and let* $\mathbf{U}$ *be a surface normal to* M *defined in a neighborhood of a point* $p \in M$. *For a tangent vector* v_p *to* M *at* p *we put* $S(v_p) = -D_v \mathbf{U}$. *Then* S *is called the shape operator.*

Definition (Gaussian curvature [46]). *Let* M *be a regular surface in* R^3. *The Gaussian curvature* K *of* M *are the functions* K *defined by* $K(p) = det(S(p))$

Definition (Polygonal region [46]). *A polygonal region is a regular region with a finite number of vertices, each of which has a nonzero external angle.*

Definition (Polygonal decomposition [46]). *A polygonal decomposition of a regular region $\mathcal{Q}$ of an abstract surface is a finite collection $\Re$ of polygonal regions called faces such that*

1. *each $\mathcal{R} \in \Re$ is homeomorphic to a disk;*
2. *the union of all regions in $\Re$ is $\mathcal{Q}$;*
3. *if $\mathcal{R}_1, \mathcal{R}_2 \in \Re$, then the intersection $\mathcal{R}_1 \cap \mathcal{R}_2$ is either a common edge or a common vertex of $\mathcal{R}_1$ and $\mathcal{R}_2$*

Definition (Euler characteristic [46]). *Let R be a polygonal decomposition of a regular region O of an abstract surface. We put*
V = the number of vertices of R
E = the number of edges of R
F = the number of faces of R
*Then $\chi(O) = F - E + V$ is called the **Euler characteristic** of O.*

Theorem (Gauss-Bonnet Theorem [32]). *Let $R \subset S$ be a regular region of an oriented surface and let $C_1, \ldots, C_n$ be the closed, simple, piecewise, regular curves which form the boundary ∂R of R. Suppose that each C_i is positively oriented and let $\varphi_1, \ldots, \varphi_p$ be the set of all external angles of the curves $C_1, \ldots, C_n$. Then*

$$\sum_{i=1}^{n} \int_{C_i} k_g(s)\, \mathrm{d}s + \int\int_R K\, \mathrm{d}\sigma + \sum_{l=1}^{p} \varphi_i = 2\pi\chi(R) \tag{4.4}$$

where s denotes the arc length of C_i, $k_g(s)$ is the geodesic curvature of the regular arcs of C_i, K denotes the Gaussian Curvature, $\chi(R)$ is the Euler characteristic.

For the planar case, the Gaussian curvature is zero and the Euler characteristic equals to 1. Equation (4.4) reduces to

$$\sum_{i=1}^{n} \int_{C_i} k_g(s)\,\mathrm{d}s + \sum_{l=1}^{p} \varphi_i = 2\pi. \tag{4.5}$$

4.2 Application Of The Gauss-Bonnet Theorem

In the two-dimensional planar motion of the abstract model, the Gauss-Bonnet Theorem implies

$$\sum_{j=0}^{L} \int_{s_j}^{s_{j+1}} k(s)\,\mathrm{d}s + \sum_{j=0}^{L} \varphi_j = 2\pi. \tag{4.6}$$

In (4.6), L is the total number of steps that the model takes, s_j is the arc length of the j-th step, and $k(s)$ is the curvature of the curve associated with each step, given by

$$k = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{\frac{3}{2}}}. \tag{4.7}$$

The instantaneous change in orientation is denoted φ_j. Select an appropriate number of steps that allows the model to complete a closed circular curve, that is,

$$\int_C \mathrm{d}s = 2\pi R. \tag{4.8}$$

Substituting (4.8) into (4.6), we find

$$R = \frac{\int_C \mathrm{d}s}{\sum_{j=0}^{L} \int_{s_j}^{s_{j+1}} k(s)\,\mathrm{d}s + \sum_{j=0}^{L} \varphi_j}. \tag{4.9}$$

Since the model's path is constructed by concatenating the displacements generated by the left and right pairs, we focus on the radius of curvature in a single step. Let w_l denotes the left switch angle, w_r denotes the right switch angle, l_l denotes the left stride length, l_r denotes the right stride length, c_l denotes the left

stride curvature integral, c_r denotes the right stride curvature integral and p denotes the number of steps to complete the circle. Utilizing (4.9), since the number of left cycles and right cycles in the path are equal, the radius of curvature for the robot can be expressed as

$$R = \frac{p(l_l + l_r)}{p(w_l + w_r) + p(c_l + c_r)} = \frac{l_l + l_r}{w_l + w_r + c_l + c_r}. \tag{4.10}$$

For touchdown angles ${\phi_i}^{td} = \pi/6, (i = 1, 2, 3, 4)$, Table 4.1 lists the resulting path characteristics of the closed circular curve produced by different sweep angle asymmetries.

Table 4.1: Whole asymmetry data (l = 3, d = 13)

$\Delta\psi$ (rad)	0.2	0.18	0.16	0.14	0.12	0.1
left-right switch angle (rad)	0.0716	0.0620	0.0531	0.0448	0.0370	0.0297
right-left switch angle (rad)	0.0378	0.0331	0.0286	0.0243	0.0201	0.0162
left stride length (m)	0.0339	0.0335	0.0331	0.0328	0.0324	0.0320
right stride length (m)	0.0254	0.0259	0.0264	0.0269	0.0273	0.0278
left stride curvature integral	-0.0306	-0.0255	-0.0209	-0.0168	-0.0132	-0.0101
right stride curvature integral	-0.0031	-0.0034	-0.0036	-0.0037	-0.0036	-0.0035
strides	85	95	106	121	140	168
radius (m)	0.80	0.89	1.00	1.14	1.33	1.6

Substituting the values of $w_l, w_r, l_l, l_r, c_l, c_r$ into (4.10), we can verify the validity of the equation. There is always some small offset, the endpoints of the curve do not coincide exactly, which can be attributed to numerical errors and the fact that the model's path is circular only on average.

With this analysis, we can map the desired aggregate path characteristics with the model parameters that can generate paths with these attributes. In other words, we know how to run the model so that we produce desired paths on average. This knowledge can now be used to associate the motion of the abstract model to the curves of Dubin's car, for which extensive analysis exists and applications to motion and path planning abound.

Chapter 5

PATH PLANNING

In order to complete inspections, reconnaissance, search-and-rescue and sensor coverage in unstructured environments using a crawling robot such as the Octoroach, its critical to be able to plan the motion of this robot from point A to point B. The kinematic model and the corresponding motion primitives we have obtained in Chapter 3, in conjunction with the path characterization offered in the previous chapter will be used in this chapter to develop a path planning methodology for the OctoRoACH.

5.1 The Dubin's Car Model

Here, we have a simplified model: a car which can only move forward and always moving at a unit velocity so that we don't need to worry about acceleration and brake. This model, is called a Dubin's car model.

In 1957, Lester Eli Dubins demonstrated that any curve generated by this model is necessarily a continuously differentiable curve which is either

1. an arc of a circle of radius R connected by a line segment, connected by an arc of a circle of radius R.

2. a sequence of three arcs of circle of radius R.

3. a subpath of a path of type (1) or (2).

Let S denote a straight line segment, L a left turn arc, R a right turn arc, There can be six combinations of arcs and segments that conform to the description provided by Dubins regarding the optimal curves connecting one planar position and orientation to another: $D = [LSL, RSR, RSL, LSR, RLR, LRL]$. A more detailed characterization and computational prescription of these paths is given in literature [10]. In this thesis, we directly use those known minimum-length solutions for the Dubins' model to solve the path planning problem for the OctoRoACH.

5.2 Minimum Radius

Due to the mechanism geometry, the angle between O_1A and the robot's longitudinal axis should be no larger than π. In other words, the asymmetry in the sweep angle $\Delta\psi$ should be no larger than $\pi/2 - \phi^{td}$. Since the radius R decreases monotonically as $\Delta\psi$ increases, we can obtain the minimum radius R_{min} at the configuration of the largest $\Delta\psi$. For example, if $\phi^{td} = \pi/6$, then the smallest radius can be obtained at $\Delta\psi = \pi/3$.

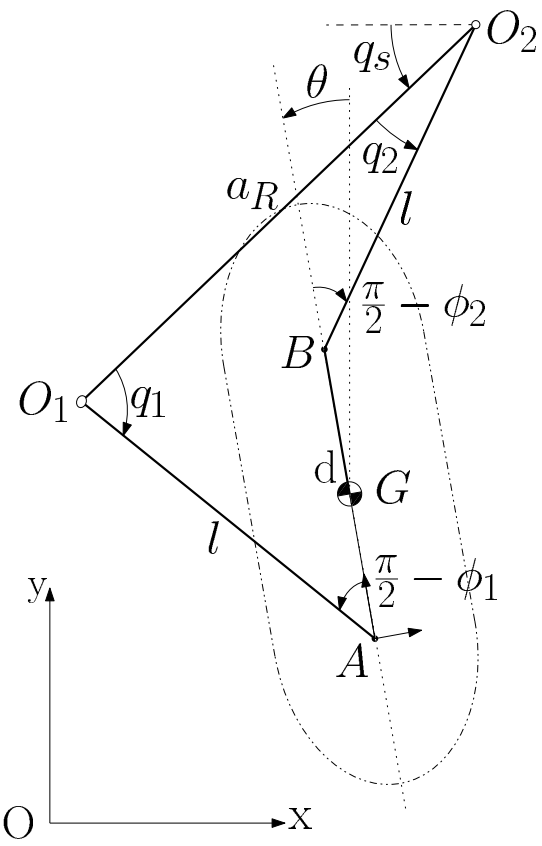


Figure 5.1: Abstracted model for OctoRoACH [23].

5.3 Planning Examples

Suppose the robot's geometric center initially coincides with the origin of the global, inertial coordinate frame with the longitudinal axis of the body frame aligned with the positive y semi-axis of the global frame. The goal for the system is for the geometric center to reach the (100 cm, 0) point with the system's orientation being that of the negative y semi-axis (Figure 5.2).

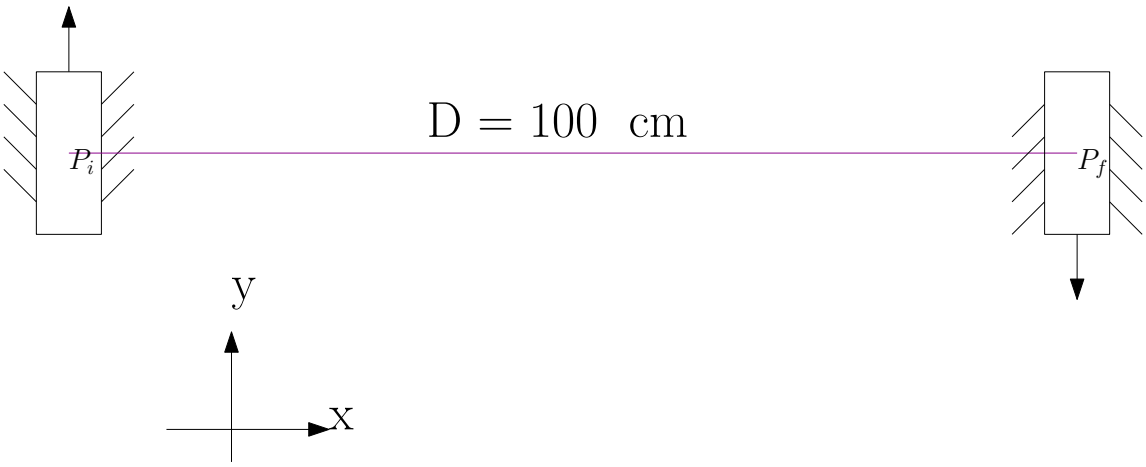


Figure 5.2: Initial and final configurations in the first motion planning problem instantiation

If the system were the Dubin's Car, then the minimum length path between these two configurations could have been obtained by solving a geometric construction problem, knowing the combination of line segments and arcs that make this optimal curve (Figure 5.3). Note that the smallest radius for $\phi^{td} = \pi/6$ is obtained at $\Delta\psi = \pi/3$, where R = 18.75 cm.

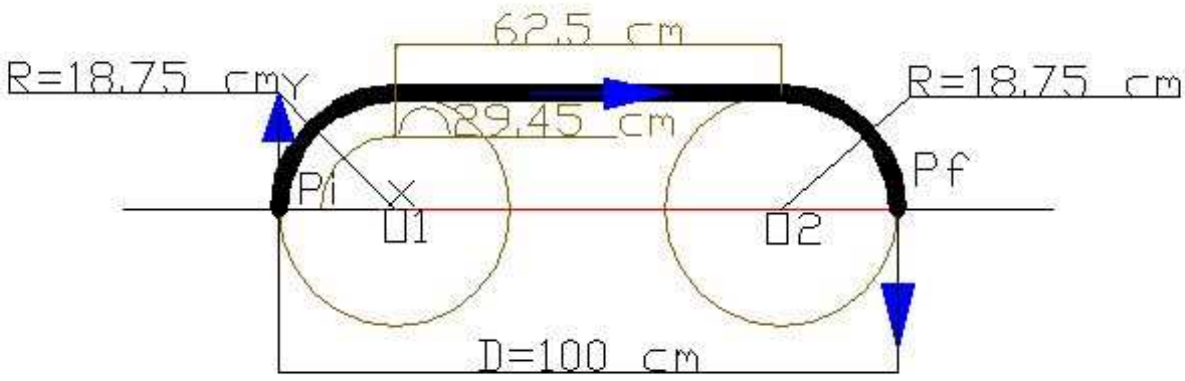


Figure 5.3: The Dubin's curve that connects initial and final configuration in the first problem instantiation

The optimal curve in this case is a RSR path: the robot first traverses a circular arc path of 90° with R = 18.75 cm, then moves along a straight segment for 62.5 cm, and then traverses another circular arc of same radius and angle, but in the opposite direction. Implementing this path plan in our model, we first set $\Delta\psi = \pi/3$ and $\psi_1 > \psi_3$ to achieve a tight right turn. For the straight line motion segment, we set $\Delta\psi = 0$. Since the model needs almost 27 steps to transcribe a circular path with $\Delta\psi = \pi/3$, we let the model take $\lfloor \frac{27}{4} \rfloor$ steps. The resulting path the model produces shown in Figure 5.4.

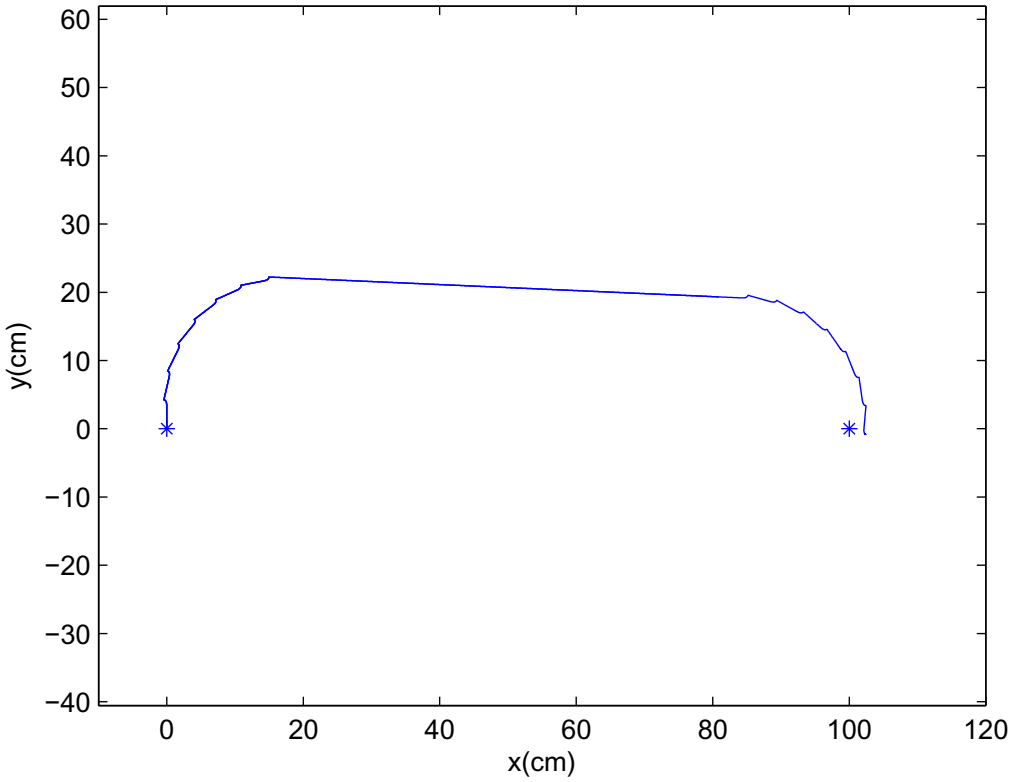


Figure 5.4: Realization of the path plan on the robot model for the first problem instantiation

A comparison of Figure 5.3 and 5.4 reveals that there are differences between the plan and its realization. One reason is that the Dubin's path segments are always an idealization of the model's motion. Another factor that contributes to the mismatch is the cumulative effect of numerical errors in matlab simulations. Yet another factor is that the motion of the model is quantized: it cannot stop the step mid-way, and therefore it is impossible for the model to realize $\frac{27}{4}$ steps in a circular path.

A different target configuration is shown in Figure 5.5.

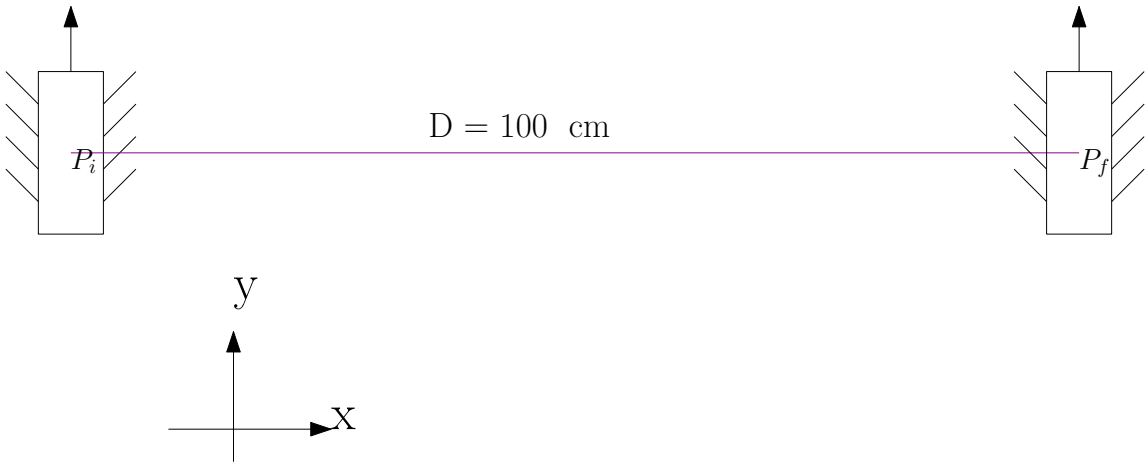


Figure 5.5: Initial and final configurations in the second motion planning problem instantiation

The Dubin's curve solution to this example is shown in Figure 5.6. After making the inner tangent line for the two circle, it shows that the robot first needs to turn 126° clockwise , then go straight for 50 cm, and finally make another 126° counter-clockwise turn.

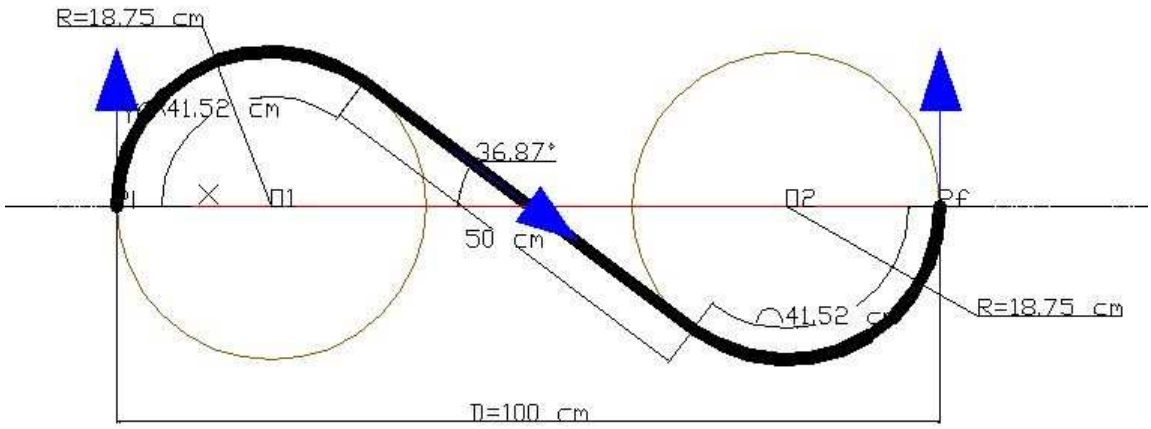


Figure 5.6: Dubin's curve solution for the second problem instantiation

To implement this plan, the model needs to take $\lfloor \frac{126*27}{360} \rfloor$ steps along a circle with radius R = 18.75 cm, continue straight for another 50 cm, and then walk $\lfloor \frac{126*27}{360} \rfloor$ steps of the curved resemble path counter-clockwise.

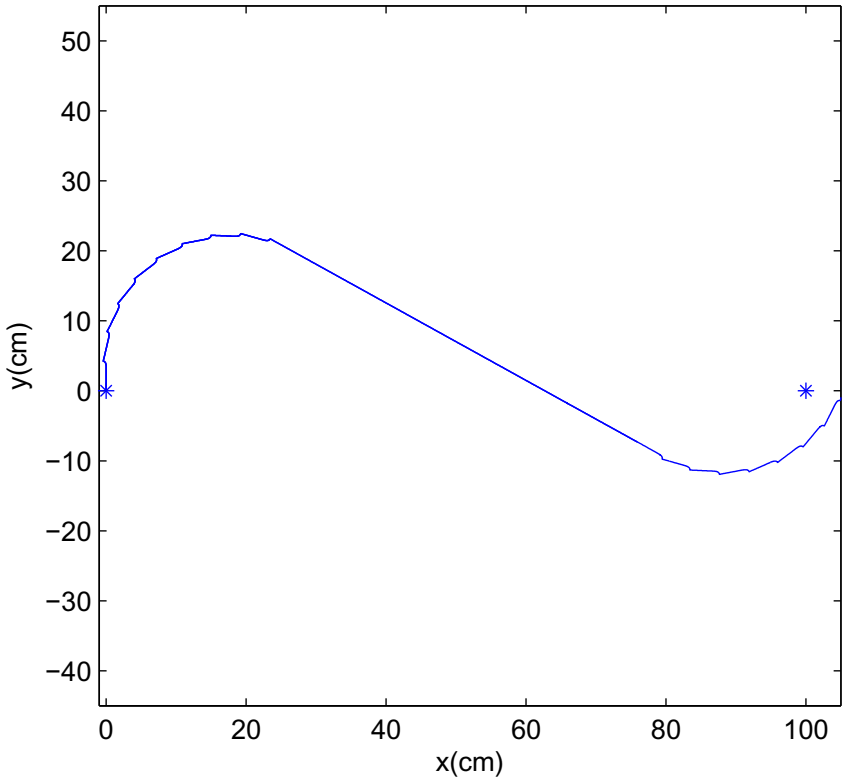


Figure 5.7: Realization of the path plan on the model for the second problem instantiation

In this chapter we cast the path planning problem for OctoRoACH model as a path planning problem for Dubin's car, by associating the optimal Dubin's curves to model parameters that make the path generated by the model match those of Dubin's. A key to this association has been the application of the Gauss-Bonnet theorem that allowed the quantification, on average, of the model's paths as a function of model's parameters.

Chapter 6

CONCLUSIONS AND FUTURE WORK

This chapter gives a short summary of the work presented in this thesis and suggest possible future directions this research can take on the topic of motion planning for miniature multi legged robots.

6.1 Problem Addressed

The Dubin's car model has been widely used in solutions of motion planning problems for wheeled robots. There is a body of literature on its use that can be leveraged for the solution of motion planning problems involving miniature legged robots.

In this thesis, we utilized an existing kinematic model that was developed as an abstraction of the quasi-static motion behavior of the OctoRoACH robot. The eight-legged mechanism has been abstracted into a switching four-bar linkage. The kinematics analysis of the model enables us to compute the heading angle and the position of the geometric center of the robot after each step, assuming continuous contact of active legs with the ground and no slipping.

Chapter 3 presents a kinematics analysis of the motion generated when the hind left and front right legs are pushing the model mechanism forward. It is indicated that the analysis of the case where the hind right and front left are pushing, mirrors the former one. This analysis forms the basis for the definition of primitive motions for the robot. Through concatenating these left-right leg pair cycles, we

can form motion primitives that inform us on how the robot can move straight and turn. We thus define three motion primitives: go straight, turn left, turn right. We identify the combinations of model parameters that generate these primitive motion behaviors. Specifically, we discovered that straight line motion is achieved by setting touchdown and liftoff angles of the active legs to be equal in magnitude and opposite in sign. To generate a curved path, one can introduce some asymmetry between touchdown and liftoff angles. In particular, clockwise (right) turn requires the liftoff angle of the right pair to be larger in size than the touchdown, and the liftoff angle of the left pair to be smaller in size than the touchdown angle. This implies that in terms of leg sweeping angles, $\psi_1 > \psi_3$; for counterclockwise (left) turn, these angle relations are reversed, and the sweep angle of left pair ends up being larger than that of the right pair, $\psi_3 > \psi_1$. We observed that in curved path motion, the radius of curvature is monotonically decreasing with $\Delta\psi$. The model kinematic analysis and motion primitives is a product of collaboration within our research group, with contributions from Konstantinos Karydis, Dr. Tanner and Dr. Poulakakis.

In Chapter 4, we offer a method to analytically compute the curvature of a path, using Gauss-Bonnet Theorem. In the planar motion case of our switching four bar mechanism, the Gauss-Bonnet Theorem simplifies to a directly computable equation. Application of this theorem allows us to account for the discontinuities in velocity, acceleration and curvature that occur when the mechanism switches between steps. This gives us an equation which yields an average estimate of the radius of curvature for the curved paths produced in our two of the three motion primitives.

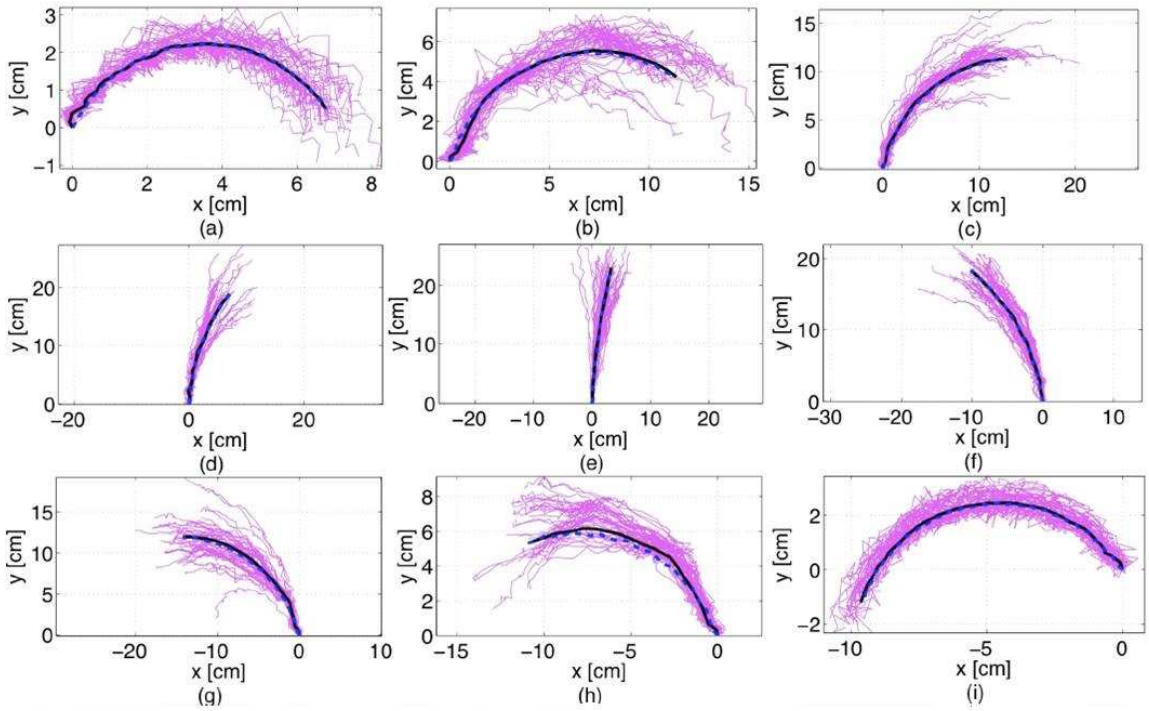


Figure 6.1: Model matches experiment data

This is where the main contribution of this thesis lies. Specifically, it is in associating model parameters to motion primitive geometries in an analytic, quantitative way, that gives the motion planner a look-up table like Table 4.1. Perhaps more importantly, this thesis outlines a methodology for generating tables like Table 4.1 for different desired motion primitives, and different possible model instantiations. As soon as the desired motion primitives are associated to average motion behaviors of the actual robotic hardware (Figure 6.1), the model populated with the parameters drawn from a table like Table 4.1 can be directly used for motion planning. Figure 6.1 shows how the path we plot for the model match with the real robot. The pink lines are the exact paths of OctoRoACH while the black solid lines are the average of these experiments. The blue dashed lines are the paths we plotted

for the model in Matlab. We can find that the black solid lines match with the blue dashed lines quite well. Plots (a), (b), (c), (d) show the right turn cases while plots (f), (g), (h), (i) show the left turn cases and plot (e) shows the straight line case.

Having established the motion primitive parameters and given the characterization of the resulting curvature of the paths generated, we associate the primitive motions of this model to the curves of the Dubin's car model. We borrow the solution approach to the path planning problem based on Dubin's model to show in two examples that the solution transfers over the legged robot model. We notice that there exist some small path deviations, which is reasonable since the Dubin's car model is ideal and the model's motion is quantized. Besides, matlab simulation involve some numerical errors.

6.2 Perspectives On Future Work

In this thesis, we have proposed a method to plan the motion of the miniature legged robot OctoRoACH by approximating path derived using Dubin's car model. Since the switching four bar mechanism used in this thesis can be thought of representative template model of low-speed crawling gaits, the algorithm proposed in this thesis can be used to plan the motion of other crawling robots. However, it should be mentioned that, Dubin's model does not take obstacles into consideration. To cope with this situation, future work will focus on extending the basic motion primitive path planing method we have described here to computing some obstacle-free paths for multilegged crawling robots like OcotRoACH.

With the ability to find paths that connect the initial position of the robot with the final, desired one and avoid obstacles that may exist in between, we can explore the potential of these robots in real-life tasks such as search-and-rescue missions or exploration of an unknown environment. To achieve such tasks, the robot platform

must be augmented with a number of suitable sensors that will enable the robot to sense its environment and locate it self. Future work will examine how sensor data can be used to plan suitable paths in real-time. Furthermore, applications of these robots as part of multi-agent systems will be explored. We anticipate that the simple kinematic model of the robot, together with the analytic characterization of the geometric properties in terms of practically relevant quantities of the paths it can produce will serve as a building block towards engaging robots like the OctoRoACH in real-world tasks.

BIBLIOGRAPHY

[1] Zarrouk D, Pullin A, Kohut N, Fearing R."STAR, A Sprawl Tuned Autonomous Robot." In: Proceedings of the *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany 2013, pp 20-25

[2] Morrey JM, Lambrecht B, Horchler AD, Ritzmann RE, Quinn RD. "Highly Mobile and Robust Small Quadruped Robots." In: Proceedings of the *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV 2003 vol 1, pp 82-87

[3] Lambrecht B, Horchler AD, Quinn R. "A small, insect-inspired robot that runs and jumps." In: Proceedings of the *IEEE/RSJ International Conference on Robotics and Automation*, Barcelona, Spain 2005, pp 1240-1245

[4] Birkmeyer P, Peterson K, Fearing RS. "DASH: A dynamic 16g hexapedal robot." In: Proceedings of the *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Saint Louis, MO 2009, pp 2683-2689

[5] Kohut N, Hoover A, Ma K, Baek S, Fearing R. "MEDIC: A legged millirobot utilizing novel obstacle traversal." In: Proceedings of the *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp 802-808

[6] Kim S, Clark JE, Cutkosky MR. "iSprawl: Design and Tuning for High-speed Autonomous Open-loop Running." The *International Journal of Robotics Research*, 2006, 25(9):903-912

[7] Hoover AM, Burden S, Fu XY, Sastry S, Fearing RS. "Bio-inspired design and dynamic maneuverability of a minimally actuated six-legged robot." In: Proceedings of the *IEEE International Conference on Biomedical Robotics and Biomechatronics*, Tokyo, Japan, 2010, pp 869-876

[8] Pullin A, Kohut N, Zarrouk D, Fearing R. "Dynamic turning of 13 cm robot comparing tail and differential drive." In: Proceedings of the *IEEE International Conference on Robotics and Automation*, Saint Paul, MN, 2012, pp 5086-5093

[9] Choset H, Burgard W, Hutchinson S, Kantor G, Kavraki LE, Lynch K, Thrun S. *Principles of Robot Motion: Theory, Algorithms, and Implementation*, MIT Press, 2005

[10] S. M. LaValle. *Planning Algorithms*, Cambridge University Press, 2006

[11] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." In: Proceedings of the *IEEE International Conference on Robotics and Automation*, 12(4) 1996, pp 566-580

[12] L. E. Kavraki, M. N. Kolountzakis, and J. C. Latombe. "Analysis of probabilistic roadmaps for path planning." In: Proceedings of the *IEEE Transactions on Robotics and Automation*, 14(1):166-171, 1998

[13] J. J. Kuffner and S. M. LaValle. "RRT-connect: An efficient approach to single-quert path planning." In: Proceedings of the *IEEE International Conference on Robotics and Automation*, 2000, pp 995-1001 vol.2

[14] S. M. LaValle and J. J. Kuffner. "Randomized kinodynamic planning." *International Journal of Robotics Research*, 20(5):378-400, May 2001

[15] Karydis K, Poulakakis I, Tanner HG "A switching kinematic model for an octapedal robot." In: Proceedings of the *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, 2012 Portugal, pp 507-512

[16] J. Barraquand and J.C. Latombe. "Robot motion planning: A distributed representation approach." The *International Journal of Robot and Research*, 10(6):628-649, December 1991

[17] E. Frazzoli, M. A. Dahleh, and E. Feron. "Real-time motion planning for agile autonomous vehicles." The *Journal of Guidance, Control, and Dynamics*, 25(1):116-129, 2002

[18] M. S. Branicky, M. M. Curtis, J. A. Levine, and S. B. Morgan. "RRTs for nonlinear, discrete, and hybrid planning and control." In *IEEE Conference on Decision and Control*, 2003, pp 657-663, Vol.1

[19] Jean-Claude Latombe. *Robot Motion Planning*, Boston: Kluwer Academic Publishers, 1991

[20] L.E. Dubins. "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents." *American Journal of Mathematics*, 79 (1957) 497-516

[21] Peng Cheng, Zuojun Sheng and Steven M. Lavalle. "RRT-Based Trajectory Design for Autonomous Automobiles and Spacecraft." *Archives of Control Sciences*, Volume 11, 2001, No. 3-4, pp 51-78

[22] Uluc Saranli , Martin Buehler , Daniel E. Koditschek. "RHex: A simple and highly mobile hexapod robot" *International Journal of Robotics Research*, 2001 vol.20, no.7, pp 616-631

[23] Konstantinos Karydis, Yan Liu, Ioannis Poulakakis, Herbert G. Tanner. "A Template Model for Miniature Legged Robots", submitted, 2014

[24] Hoover AM, Steltz E, Fearing RS. "RoACH: An autonomous 2.4g crawling hexapod robot" In Proceedings of the *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp 26-33, 2008

[25] D. J. Balkcom and M. T. Mason. "Time optimal trajectories for bounded velocity differential drive vehicles" *International Journal of Robotics Research*, 21(3):199-217, 2002

[26] J.-D. Boissonnat, A. Cérézo, and J. Leblond. "Shortest paths of bounded curvature in the plane" *Journal of Intelligent and Robotic Systems*, vol 11, pp 5-20, 1994.

[27] H. Chitsaz, S. M. LaValle, D. J. Balkcom, and M. T. Mason. "Minimum wheel-rotation paths for differential-drive mobile robots." In Proceedings *IEEE International Conference on Robotics and Automation*, 2006. pp 1616-1623

[28] J. A. Reeds and L. A. Shepp. "Optimal paths for a car that goes both forwards and backwards." In Proceedings *Pacific Journal of Mathematics*, 145(2), pp 367-393, 1990.

[29] P. Souéres and J.-P. Laumond. "Shortest paths synthesis for a car-like robot." In *IEEE Transactions on Automatic Control*, pp 672-688, 1996.

[30] H. Sussmann and G. Tang. "Shortest paths for the Reeds-Shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control." *Technical Report SYNCON* -91-10, Dept. of Mathematics, Rutgers University, Piscataway, NJ, 1991.

[31] Laumond, J.-P, Jacobs, P.E., Taix, M., Murray, R.M. "A motion planner for nonholonomic mobile robots." *IEEE transaction on Robotics and Automation*, Volume 10, Issue 5, pp 577-493, 1994.

[32] Do Carmo MP. *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ, 1976

[33] E. Frazzoli, M. A. Dahleh, and E. Feron. "Maneuver-based motion planning for nonlinear systems with symmetries." *IEEE Transactions on Robotics*, 21(6), pp 1077-1091, December 2005.

[34] E. Frazzoli. *Robust Hybrid Control of Autonomous Vehicle Motion Planning*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2001.

[35] E. Frazzoli, J. Go, T. Vu, and J. J. Kuffner. "Autonomous behaviors for interactive vehicle animations." In Proceedings *SIGGRAPH Symposium on Computer Animation*, Volume 68, Issue 2, March 2006, Pages 90-112.

[36] A. Piazzi, M. Romano, and C. G. Lo Bianco. "G3 splines for the path planning of wheeled mobile robots." In Proceedings *European Control Conference*, 2003.

[37] M. Pivtoraiko and A. Kelly. "Generating near minimal spanning control sets for constrained motion planning in discrete state spaces." In Proceedings *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3231 - 3237.

[38] A. Scheuer and T. Fraichard. "Collision-free and continuous-curvature path planning for car-like robots." In Proceedings *IEEE International Conference on Robotics and Automation*, pages 867-873, 1997.

[39] R.W. Brockett. "Formal languages for motion description and map making." In *Robotics*, pages 181-193. American Mathematical Society, 1990.

[40] R.W. Brockett. "Hybrid models for motion control." In H. Trentelman and J. C. Willems, editors, *Perspectives in Control*, pages 29-51. Birkhauser Verlag,1993

[41] R.W. Brockett. "On the computer control of movement." in Proceedings of *IEEE Conference on Robotics and Automation*,April 1988, pp. 534-540.

[42] V. Manikonda, P. S. Krishnaprasad, and J. Hendler, "A motion description language and a hybrid architecture for motion plannning with nonholonomic robots," In Proceedings *IEEE International Conference on Robotics and Automation*, vol. 2, May 1995, pp. 2021-8.

[43] V. Manikonda, J. Hendler, and P. S. Krishnaprasad. "Formalizing behavior-based planning for nonholonomic robots." In Proceedings *International Joint Conf. on Artificial Intelligence*, volume 1, pages 142-9, August 1995.

[44] D. Hristu-Varsakelis, S. Andersson, F. Zhang , P. Sodre, P. S. Krishnaprasad. "A Motion Description Language for Hybrid System Programming", 2003

[45] Kaplan, W. *Advanced Calculus*, 3rd ed. Reading, MA: Addison-Wesley, pp. 98-99, 123, and 238-245, 1984.

[46] Gray, A. *Modern Differential Geometry of Curves and Surfaces with Mathematica*, 2nd ed. Boca Raton, FL: CRC Press, pp. 281-286, 1997.

[47] Konstantinos Karydis, Ioannis Poulakakis and Herbert Tanner. "Motion planning for Miniature Legged Robot Using A Horizontal Plane Template Model." *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014 (submitted).

Appendix

PERMISSION FROM ANDREW PULLIN FOR USING OCTOROACH FIGURE

Hi Yan,

I'm happy to hear that you used the robot as a tool for a successful project. Using content from the OctoRoACh paper, or any picture or likeness of the robot is absolutely fine.

Will any of your work be shown at the upcoming MAST meeting?

Let me know if you want any higher resolution versions of any of the images from the paper.

- Andrew

On 2/24/2014 12:28 PM, Yan Liu wrote: Andrew,

Greetings.

First let me introduce myself. I am a master candidate in Cooperative Robotics Labtorary in Mechanical Engineering in University of Delaware. My master thesis focuses on finding the motion planning strategy for Octoroach, During my research, I have read your paper "Dynamic turning of 13 cm robot comparing tail and differential drive". The work you have done is awesome. The robot offered me an excellent platform to study the motion planning trajectory.

Now, I am writing my master thesis. Since my motion planning strategy is for your Octoroach, I would like to ask for your permission to involve some pictures of your paper (the rear and side view of the robot) into my thesis.

Looking forward to your reply. Thanks.

Yan