The Minnesota Journal of Undergraduate Mathematics

# A Motion Planning Algorithm in a Lollipop Graph

Allaoua Boughrira and Hellen Colman

Wright College

The Minnesota Journal of Undergraduate Mathematics

Volume 6 (2020-2021 Academic Year)

*Sponsored by*
School of Mathematics
University of Minnesota
Minneapolis, MN 55455

The Minnesota Journal of Undergraduate Mathematics

Volume 6 (2020-2021 Academic Year)

# A Motion Planning Algorithm in a Lollipop Graph

Allaoua Boughrira* and Hellen Colman

Wright College

Abstract. This paper is concerned with problems relevant to motion planning in robotics. Configuration spaces are of practical relevance in designing safe control schemes for robots moving on a track. The topological complexity of a configuration space is an integer which can be thought of as the minimum number of continuous instructions required to describe how to move robots between any initial configuration to any final one without collisions. We calculate this number for various examples of robots moving in different tracks represented by graphs. We present and implement an explicit algorithm for two robots to move autonomously and without collisions on a lollipop-shaped track.

## 1. Introduction

This paper surveys results concerning the motion planning problem for robots moving on graphs and includes a new explicit algorithm for the case of two robots on a lollipop graph. This algorithm is optimal in the sense that the motion planning is performed with the minimal number of instabilities.

First we introduce the configuration space of distinct robots on a graph. Configuration spaces in mathematics were introduced in the sixties by Fadell and Neuwirth [2] and first used in robotics in the eighties [11, 12]. More recently, configuration spaces of robots moving on graphs have been studied by Farber [5][8] and Ghrist [9] amongst others.

The configuration space for two robots moving on a graph is the space of all feasible combined positions of the robots. The motion planning problem deals with assigning paths between initial and final configurations. We are interested in algorithms that assign outputs (paths) to inputs (initial and final configurations) in a continuous way. These algorithms are rare in real world situations since most algorithms will have discontinuities. Farber introduced the notion of *topological complexity* of configuration spaces which measures the discontinuities in algorithms for robot navigation [4]. The topological complexity of a space is invariant under homotopy equivalence.

Our approach consists of presenting an explicit construction of the configuration space for the case of two robots moving along a lollipop-shaped track and then building a deformation retract of it that we call the skeleton. We calculate the topological complexity of the skeleton of the configuration space and exhibit an algorithm with the minimal

* *Corresponding author*

number of instructions. Finally, we translate back our instructions to the physical space where the robots move. We provide a detailed algorithm with concrete instructions for two robots to move on a lollipop-shaped track between any initial and final positions. We implement this algorithm as well and show a simulation for several cases.

We organize the paper in the following manner. In Section 2 we introduce some basic notions and notations. Section 3 introduces the basic ideas on continuous motion planning. Section 4 presents the topological complexity and its basic properties. Section 5 provides a detailed study for the case of two robots moving on a circle track. We construct the configuration space, calculate its topological complexity and present an algorithm for this case that will set the basis for our main example. Section 6 concerns the characterization of our main example: two robots moving on a lollipop track. We construct the configuration space as before and give an explicit algorithm for this case. We also implement and show a simulation for this case.

## 2. Preliminaries

In this section we briefly review some basic definitions and establish notations that we will use in this paper. From now on, the interval, circle and disk with the standard topology will be denoted as $I, S^1$ and $D$ respectively:

$$\begin{aligned} I &= [0,1] \subset \mathbb{R}, \\ S^1 &= \{(x,y) \in \mathbb{R}^2 \mid x^2+y^2 = 1\}, \\ D &= \{(x,y) \in \mathbb{R}^2 \mid x^2+y^2 < 1\}. \end{aligned}$$

2.1. **Product Topological Space.** Given two topological spaces $X$ and $Y$, consider the cartesian product:

$$X \times Y = \{(x,y) \mid x \in X,\ y \in Y\}.$$

The product space is the set $X \times Y$ endowed with the product topology.

**Example 2.1.** Let $A$ and $B$ be the following two subspaces of $\mathbb{R}^2$ in polar coordinates:

$$\begin{aligned} A &= \{(r,\theta) \mid \theta \in [0,2\pi],\ r = \frac{\pi}{6} + |\sin(\theta+\frac{\pi}{6})^5|\}, \\ B &= \{(r,\theta) \mid \theta \in [-\frac{\pi}{2},\frac{\pi}{2}],\ r = 3\cos(\theta)\}. \end{aligned}$$

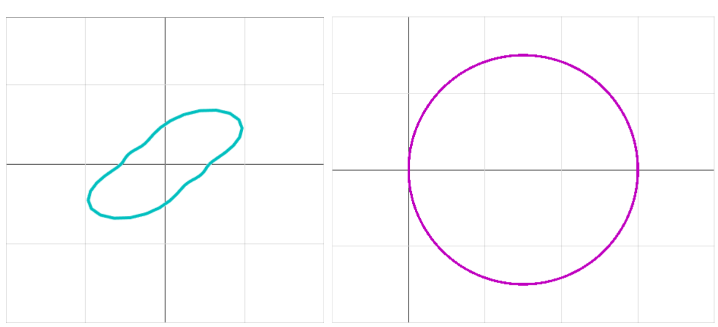

Figure 1. Subspaces $A$ and $B$

The product space shown in figure 2 is the set $A \times B = \{(x, y) \mid x \in A,\ y \in B\}$ endowed with the product topology.

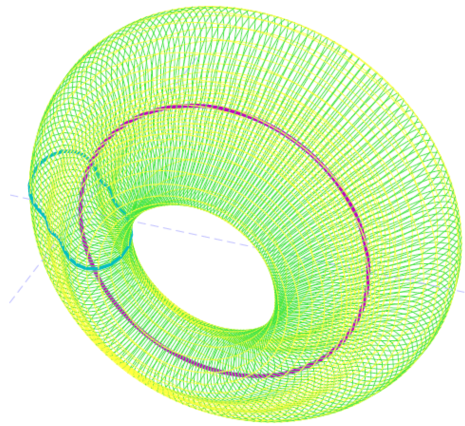

FIGURE 2. Product space $A \times B$

2.2. **Homeomorphism and Homotopy.** Two spaces $X$ and $Y$ are *homeomorphic* if there exists a homeomorphism from $X$ to $Y$. A continuous map $f : X \to Y$ is a *homeomorphism* if and only if $f$ is a bijection and $f^{-1}$ is also continuous.

Let $f$ and $g$ be two continuous maps between spaces $X$ and $Y$. Let $H : X \times I \to Y$ be a continuous map such that for each $x \in X$, $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$, then $f$ and $g$ are *homotopic* and $H$ is a *homotopy* between $f$ and $g$. We denote this by $f \simeq g$. Two spaces $X$ and $Y$ have the same *homotopy type* if there exist two functions $f : X \to Y$ and $g : Y \to X$ such that $f \circ g \simeq id_Y$ and $g \circ f \simeq id_X$. We write in this case $X \simeq Y$.

Informally, we can say that one space has the same homotopy type as another if we can *deform* one space into the other by compression, stretching and without *tearing* the space nor *gluing* any two parts of it. For instance, the space $A \times B$ in figure 2 has the same homotopy type as the *torus* $T = S^1 \times S^1$ in figure 3.

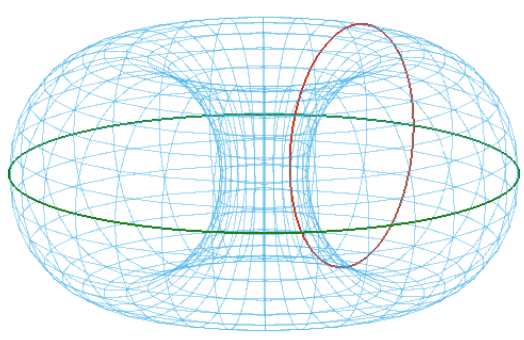

FIGURE 3. Torus $T = S^1 \times S^1$

If two spaces are homeomorphic then they have the same homotopy type. However, having the same homotopy type does not imply homeomorphism.

A topological space $X$ is *contractible* if $X$ has the same homotopy type as a point. In other words, there exists a *deformation* as described before from the space $X$ into a point. For instance, a disk $D$ is contractible whereas a circle $S^1$ is not.

## 3. Motion Planning Algorithms

In order to study the problem of planning the movement of robots in a certain space, we need to introduce certain notions and an associated space to aid with the statement as well as the solution of the problem.

3.1. **Physical and Configuration Space.** The *physical space* $\Gamma$ is the track where the robots move. The robots are assumed to be points. The combined positions of all robots at any given moment defines a *state* of the system. The space of all possible states is the *configuration space*. Each state in the configuration space is a point that represents a unique configuration of the robots' locations in the physical space. The physical space is a special case of a configuration space where there is only one robot.

The *configuration space* $X$ of $n$ robots moving on the physical space $\Gamma$ without collisions is:

$$X = C^n(\Gamma) = \underbrace{\Gamma \times \Gamma \times \ldots \times \Gamma}_{n \text{ times}} - \Delta_n = \Gamma^n - \Delta_n,$$

where $\Delta_n$ is the *fat diagonal*. The fat diagonal represents the set of $n$-tuples where some of the robots are co-located and is defined as:

$$\Delta_n = \{(x_1, x_2, \ldots, x_n) \in \Gamma^n \mid \exists j \neq i,\ x_i = x_j\}.$$

For all spaces $\Gamma$, we have that $C^1(\Gamma) = \Gamma$.

3.2. **Motion Planning.** A path $\alpha$ in $X$ is a continuous map $\alpha : I \to X$, where $I$ is the interval $[0,1]$. The *path space* $PX$ is the set of all paths in the space $X$,

$$PX = \{\alpha \mid \alpha \text{ is a path in } X\}$$

endowed with the compact-open topology [13].

The evaluation function $ev$ is a map that takes in a path $\alpha$ in $X$ and returns the *initial* and *final* points of the path.

$$\begin{aligned} ev : PX &\to X \times X \\ \alpha &\mapsto (\alpha(0), \alpha(1)) \end{aligned}$$

A *section* $s$ of the evaluation function is a function that takes in a pair of points in the space $X$ and gives out a path between them. That is,

$$\begin{aligned} s : X \times X &\to PX, \\ (a,b) &\mapsto \alpha_{a,b}. \end{aligned}$$

where $\alpha_{a,b}$ is an *instruction* to move the robots from the starting point $a$ to the ending point $b$ following that specific path. A *Motion Planning Algorithm* (MPA) is such a section.

3.3. **Continuity of Motion Planning.** Motion planning algorithms may or may not be *continuous*. In a continuous motion planning algorithm, the small changes of the path are continuously dependent on the small changes that occur at the initial or final positions. In other words, the instruction to go from $a$ to $b$ will depend *continuously* on the points $a$ and $b$.

The continuity of the MPA implies the stability of the robot behavior. That is, the *errors* due to imprecisions or uncertainties on the initial and final positions would still result in a *nearby* path.

We are interested in motion planning algorithms that assign instructions in a *continuous way*. Unfortunately, we'll see later that these are very rare.

Let us consider for instance, one robot moving on a circle, and we will try to construct a section that would define a MPA. This robot can move between its initial and final position in different ways: clockwise, counterclockwise, following the shortest path, following the longest geodesic, etc.

Let us call *preferred shortest path* from $a$ to $b$ the path along the shortest geodesic; in case that there are more than one shortest path (i.e. if $a$ is antipodal to $b$), choose the counterclockwise path.

Set the instruction to be "follow the preferred shortest path", and the section would be defined as:

$$s : S^1 \times S^1 \to PS^1,$$
$$(a, b) \mapsto \alpha_{shortest}.$$

We observe that this section is continuous in all pairs of positions except at the antipodal pairs, and so, this section is not continuous.

Figure 4 shows the initial (filled triangle icon) and final position (empty triangle icon) of a robot. We see that a small variation in the final position results in a huge variation in the output path.

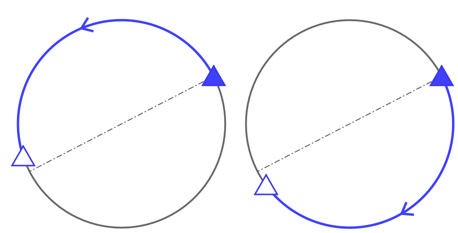

FIGURE 4. Discontinuity at the antipodal pairs

Now, let us try a different instruction, "go counterclockwise" for instance. Here also, there exists a state where the section is not continuous; that is the state $(a, a)$ for any $a$ in $S^1$. In other words where the initial and final positions coincide, the section presents a discontinuity (see figure 5).

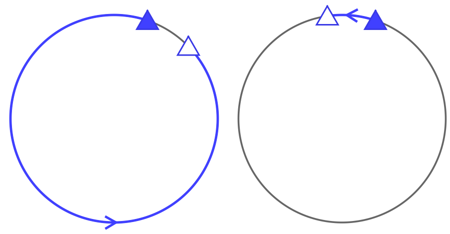

FIGURE 5. Discontinuity at configuration $(a,a)$

For both instructions, the section presented discontinuities. In fact, we will see in the next section that it is impossible to find a continuous section for this particular example.

## 4. Topological Complexity

Farber proved that a continuous MPA on a configuration space exists only if the space is contractible.

**Theorem 4.1.** [4] *A continuous motion planning algorithm in $X$ exists if and only if the configuration space $X$ is contractible.*

Since in the previous example the robot was moving on the circle $S^1$ which is also the configuration space in this case, and since a circle $S^1$ is not contractible, we have that there exists no continuous MPA for one robot moving on a circle.

We can however decompose the cartesian product of the configuration space into subdomains, with each having a continuous instruction.

**Definition 4.2.** [4] The *topological complexity* $TC(X)$ of a connected space $X$ is defined as the *minimum* integer $k$ such that the cartesian product $X \times X$ can be covered by $k$ open subsets $U_1, U_2, \dots, U_k$, such that for any $i = 1, 2, \dots, k$ there exists a *continuous* local section $s_i : U_i \to PX$ with $ev \circ s_i = \text{incl} : U_i \hookrightarrow PX$.

The definition of $TC(X)$ is given in terms of open subsets of $X \times X$ admitting continuous local sections of the evaluation map. To construct a motion planning algorithm in practice we partition the whole space $X \times X$ into pieces and define a continuous section over each of the pieces. Any such partition would contain sets which are not open and hence we need to be able to deal with subsets of $X \times X$ of more general nature.

**Definition 4.3.** A topological space $X$ is an Euclidean Neighborhood Retract (ENR) if it can be embedded into an Euclidean space $\mathbb{R}^n$ such that for some open neighborhood $X \subset U \subset \mathbb{R}^n$ there is a retraction $r : U \to X$, $r|_X = id_X : X \to X$.

**Definition 4.4.** Let $X$ be an ENR. A motion planning algorithm $s : X \times X \to PX$ is said to be tame if $X \times X$ can be split into finitely many sets (domains of continuity)

$$X \times X = F_1 \cup F_2 \cup \cdots \cup F_k$$

such that

(1) Each restriction $s|F_i : F_i \to PX$ is continuous, where $i = 1, \ldots, k$;

(2) $F_i \cap F_j = \emptyset$ for $i \neq j$;

(3) Each $F_i$ is an ENR.

It is known that for an ENR $X$, the minimal number of domains of continuity $F_1, \ldots, F_k$ in tame motion planning algorithms $s : X \times X \to PX$ equals $TC(X)$, see [7].

**Example 4.5.** One robot moving on an interval $I$. In this case, we have that the configuration space is $X = C^1(I) = I$ and $TC(I) = 1$ with domain of continuity $F = I \times I$. Since $I$ is contractible, there exists a continuous MPA where the instruction would be for example "go in straight line towards the final position".

FIGURE 6. One robot moving on I

**Example 4.6.** One robot moving on a circle $S^1$. Since there is just one robot, the configuration space is $X = C^1(S^1) = S^1$.

Since $S^1$ is not contractible, $TC(S^1) > 1$. We have seen previously two examples of MPAs for one robot on $S^1$ where both had discontinuities. Now we can consider the two subsets where each instruction is continuous so that the union would cover the cartesian product $S^1 \times S^1$.

Then, the domains of continuity will be:

$$F_1 = \{(a, b) \in S^1 \times S^1 \mid a \text{ is not antipodal to } b\} \text{ and}$$
$$F_2 = \{(a, b) \in S^1 \times S^1 \mid a \text{ is antipodal to } b\}.$$

Therefore, $TC(X) = 2$ since we exhibit an explicit covering by domains of continuity with exactly two sets.

From the MPA standpoint, this decomposition into $k$ subspaces can be interpreted as having a set of $k$ continuous instructions, where each operates in its corresponding domain of continuity. For this example, the resulting two instructions in the MPA would be as follows: If the initial and final positions are not antipodal then go following the shortest path. Otherwise, go counterclockwise.

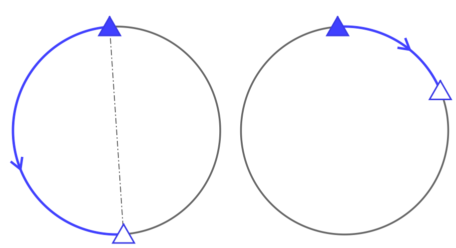

FIGURE 7. Two Instructions for one robot moving on $S^1$

Farber also proved that the topological complexity $TC(X)$ of a space $X$ is *invariant* under homotopy.

**Theorem 4.7.** [4] *If* $X \simeq Y$ *then* $TC(X) = TC(Y)$.

Computing the topological complexity of a configuration space will equip us with a lower bound for the number of instructions in any continuous motion planning algorithm in the physical space. The examples seen so far have relatively simple configuration spaces, but as we increase the number of robots and the physical space gets more complex, so does the configuration space. Finding the discontinuities directly while in the configuration space is not trivial, and the above theorem is crucial to reduce the overall complexity by using the homotopy not only to calculate the topological complexity of a simpler space, but also to construct the actual algorithm.

## 5. Two robots moving on a graph

We consider the case of two *distinct* robots $A$ and $B$ moving on a graph $\Gamma$. The problem to solve is finding a *continuous* and *collision-free* motion planning algorithm with the *least* number of instructions needed to move both robots A and B from their initial positions to their final positions.

### 5.1. **Two robots moving on an interval** $I$.

Let $A$ and $B$ be two robots moving on $\Gamma = I$.

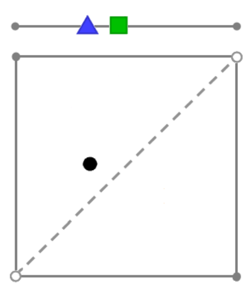

Figure 8. Physical and configuration spaces for two robots in $I$

The configuration space in this case is $X = C^2(I) = I \times I - \Delta$. The physical space and the configuration space are shown in figure 8. Each pair of positions of the two robots in the physical space defines a state in the configuration space. Figure 8 shows a concrete pair of positions in the physical space and its corresponding state in the configuration space.

We observe that the configuration space in this case is *not connected* because of the removal of the diagonal. Even though there are some states where there is a path between the initial and final positions of the two robots (see figure 9), the problem of finding a path between any initial and final configuration has no solution (figure 10). In other words, there is no MPA for this case.

This impossibility to connect two states in the configuration space by a path has its counterpart in the physical space: two robots cannot swap positions within the interval.

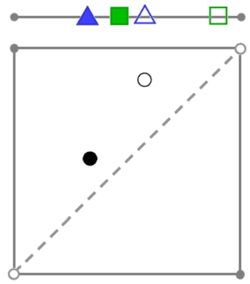

FIGURE 9. Initial and final states are in the same connected component.

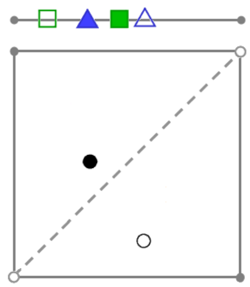

FIGURE 10. Initial and final states are in different connected components.

5.2. **Two robots moving on a circle $S^1$.** Now, let's consider two robots $A$ and $B$ moving on $\Gamma = S^1$. The configuration space in this case is given by $X = C^2(S^1) = S^1 \times S^1 - \Delta = T - \Delta$. This space is homeomorphic to the open cylinder $X = C^2(S^1) = S^1 \times (0,1)$.

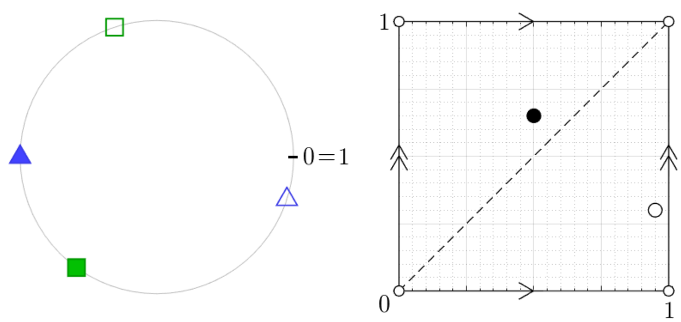


FIGURE 11. Physical and configuration spaces for two robots on $S^1$

A very useful way to represent this space would be using the *flat torus* representation as shown in figure 11. We also show in this figure the initial and final positions of two robots in the physical space and their corresponding states in the configuration space.

By working in the configuration space, the complexity of the problem is reduced from trying to find two paths to move two distinct robots from their initial positions to their respective final positions into finding one *combined* path to move the initial state of the two robots to the final state in the configuration space.

5.2.1. *Topological Complexity of the Configuration Space.* As we have seen before, the configuration space is $X = C^2(S^1) = S^1 \times (0,1)$. Since the configuration space is homotopy equivalent to a circle, we have that $TC(X) = TC(S^1 \times (0,1)) = TC(S^1) = 2$ by theorem 4.7.

This gives us the minimum number of instructions needed for a continuous MPA for two robots moving on a circle $S^1$. Knowing how to transition back and forth between the physical space and the configuration space, our objective is first to find two instructions that define the algorithm to move from the initial state to the final state in the the configuration space, and then translate them back to the physical space.

We will take advantage of the construction of an explicit homotopy between the configuration space and the circle to build our algorithm based on the known algorithm for the circle given in the example 4.6.

5.2.2. *MPA for two robots moving on $S^1$.* Recall that the configuration space $X$ is path-connected and is homotopy equivalent to a circle. For the purpose of the homotopy, we will choose a special circle to deform the configuration space into: the *antipodal circle*.

**Definition 5.1.** The *antipodal circle* $\mathcal{A} \subset C^2(S^1)$ is the set of pairs of points on the circle such that they are antipodal to each other. We write:

$$\mathcal{A} = \{(a,b) \in S^1 \times S^1 \mid a \text{ is antipodal to } b\}.$$

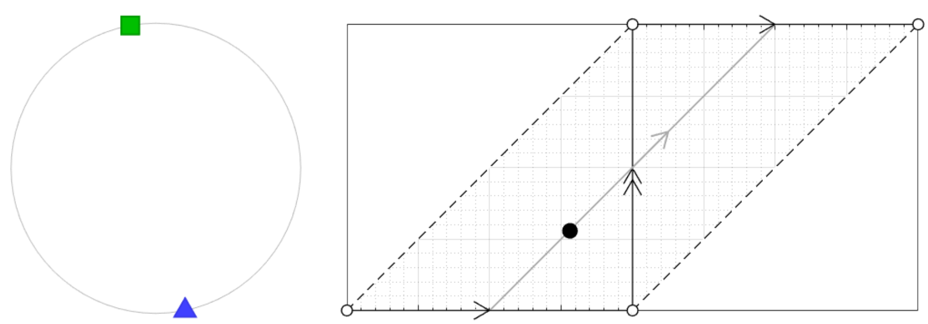

Figure 12. The antipodal circle $\mathcal{A}$ and its counterclockwise orientation

In the flat torus representation (figure 12), the antipodal circle $\mathcal{A}$ will be the set of points in $X$ lying on the line $b = a - \frac{1}{2}$.

Consider the homotopy $H : X \times I \to X$ given by the projection into $\mathcal{A}$ following horizontal lines. Recall that each state $x = (A,B)$ represents the collective positions of robots $A$ and $B$ in the circle.

Let $x_i$ and $x_f$ be the initial and final states of the robots. We call the *initial antipodal* state $x_i' = H_1(x_i)$ the image of the initial state in the antipodal circle $\mathcal{A}$ by the homotopy. Likewise, we call the *final antipodal* state $x_f' = H_1(x_f)$ the projection per homotopy of the final state in $\mathcal{A}$ (see figure 13).

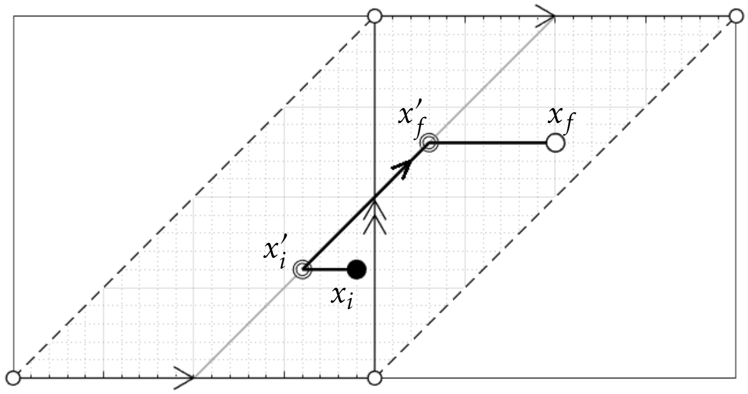


Figure 13. Initial and final antipodal states

The idea of our algorithm will be to move the initial and final states $x_i$ and $x_f$ to their antipodal states $x'_i$ and $x'_f$ respectively, and then apply in the circle $\mathcal{A}$ the known algorithm (example 4.6) to move from $x'_i$ to $x'_f$ (see figure 14).

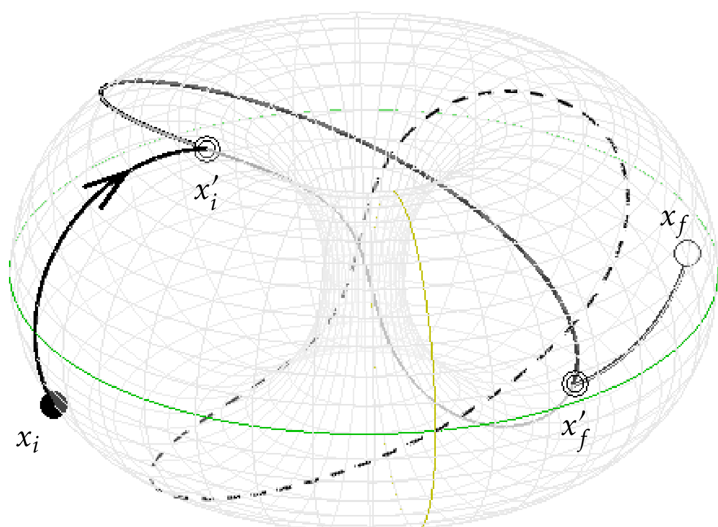


Figure 14. The steps of the MPA in $X$

We will also define certain distinguished positions in the configuration space that will correspond to the action of interchanging robots in the physical space.

**Definition 5.2.** Two states $x_i = (A_i, B_i)$ and $x_f = (A_f, B_f)$ are *swapped states* if $A_i = B_f$ and $A_f = B_i$.

Swapped states correspond to points in the configuration space that are symmetric with respect to the diagonal (see figure 16).

*Remark.* Observe that the swapped states that are in the antipodal circle $\mathcal{A}$ correspond exactly to the antipodal points in that circle.

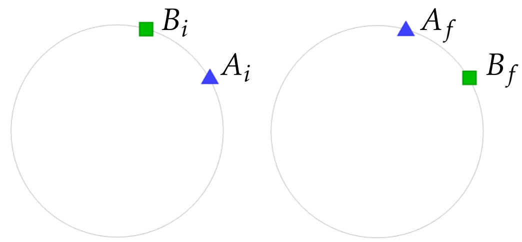


Figure 15. Swapped states in the physical space

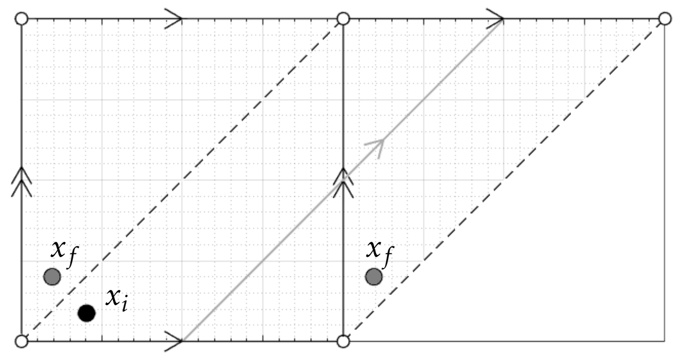


Figure 16. Swapped states in the configuration space

The following is the description of our algorithm in the configuration space $X$.

(1) Preliminary step: Move initial state $x_i$ to its antipodal state $x_i'$ on $\mathcal{A}$ following the path $H_t(x_i)$. Find the final antipodal state $x_f'$.

(2) Main step: While on $\mathcal{A}$, if the two antipodal states $x_i'$ and $x_f'$ are swapped states, then move the initial antipodal state $x_i'$ counterclockwise towards the final antipodal state $x_f'$. Otherwise, move the initial antipodal state $x_i'$ following the shortest path on $\mathcal{A}$ towards the final antipodal state $x_f'$.

(3) Final step: Move the final antipodal state $x_f'$ back to the final state $x_f$ following the reverse path to $H_t(x_f)$.

When transitioning back to the physical space, we obtain the following MPA to move two robots $A$ and $B$ in a circle:

(1) Preliminary step: Move robot $A$ away from robot $B$ until robot $A$ reaches the antipodal position of the robot $B$. Robot $B$ remains stationary.

(2) Main step: If the final position of the robot $B$ is antipodal to its initial position, then move both robots in counterclockwise direction until robot $B$ reaches its final destination. Otherwise, move both robots in the same direction following the shortest path for robot $B$ to its final position, until robot $B$ reaches its final position.

(3) Final step: Move robot $A$ following the shortest path to its final position, until it reaches its final position. Robot $B$ remains stationary.

Note that the preliminary and final steps are common steps in both instructions, the main step is where we deal with the discontinuity of the MPA.

The domains of continuity will be determined by the regions described before. Let $F_1 = \{((a,b),(a',b')) \in X \times X \mid b \text{ is antipodal to } b'\}$ and $F_2$ the set of all other pairs of points in the configuration space. Then $\{F_1, F_2\}$ is a covering of $X \times X$ by domains of continuity.

5.3. **Running the proposed MPAs.**

**Example 5.3.** Initial and final positions of the second robot are not antipodals.

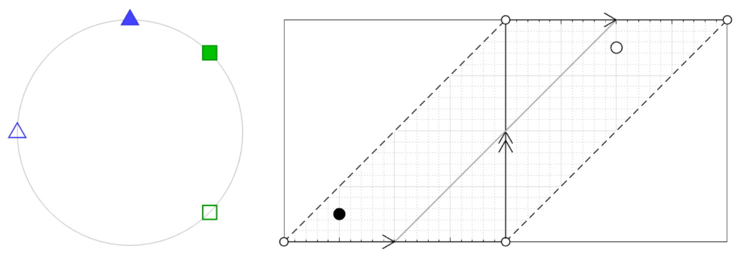

Figure 17. Initial setup

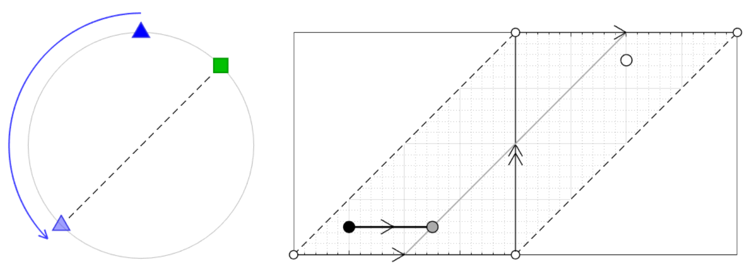

Figure 18. Preliminary step

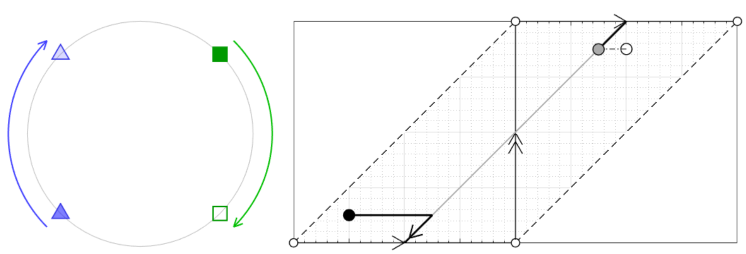

Figure 19. Main step: ”Move following the shortest path”

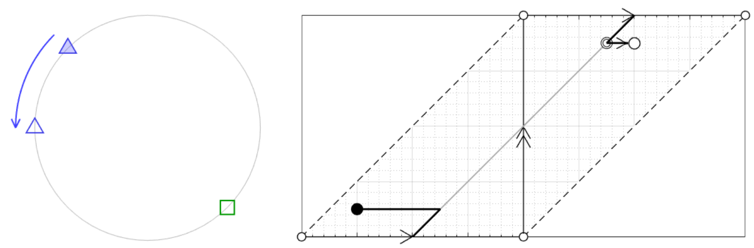

Figure 20. Final step

**Example 5.4.** Initial and final positions of the second robot are antipodals.

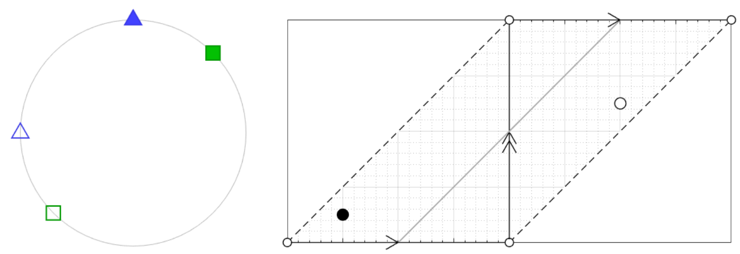

Figure 21. Initial setup

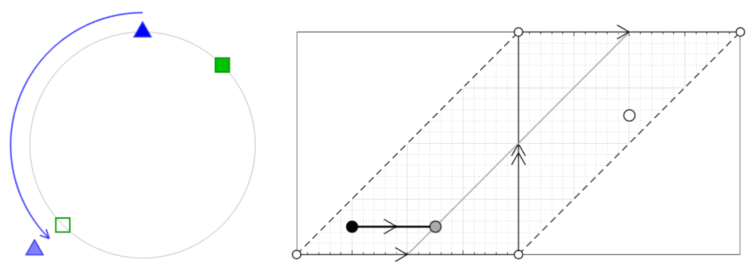

Figure 22. Preliminary step

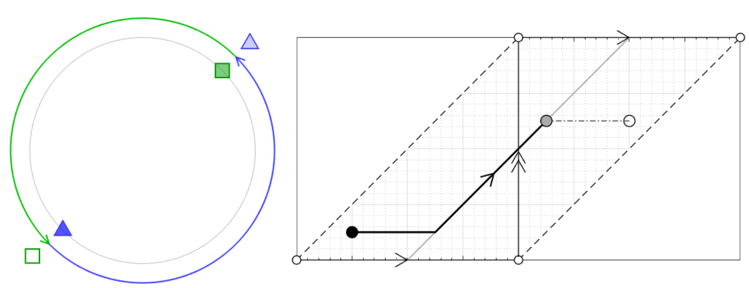

Figure 23. Main step: "Move both robots in counterclockwise direction"

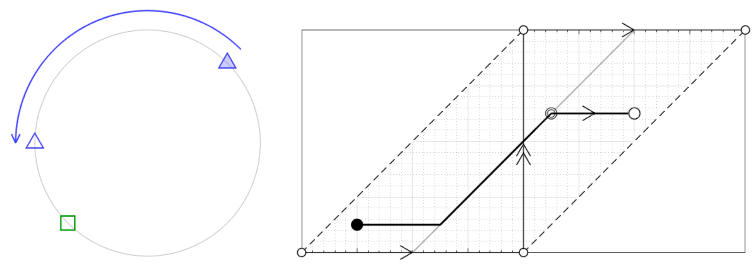

Figure 24. Final step

In the next section, we'll expand this construction of the algorithm to derive a MPA for two robots moving on a lollipop-shaped track.

## 6. Main example: Lollipop graph $\Gamma = L$

We focus our efforts now on the construction of a MPA for the case of two robots moving on a track consisting of a circle with an interval attached.

We consider the class $G_{n,g}$ of connected graphs on $n$ vertices with fixed girth $g$ (recall that the girth of a graph is the length of its shortest cycle) of which the *generalized lollipop graphs*, $C_{n,g}$, will be the ones obtained by appending a single $g$ cycle to a pendant vertex (i.e. a vertex of degree 1) of a path on $n-g$ vertices [3]. See figure 25.

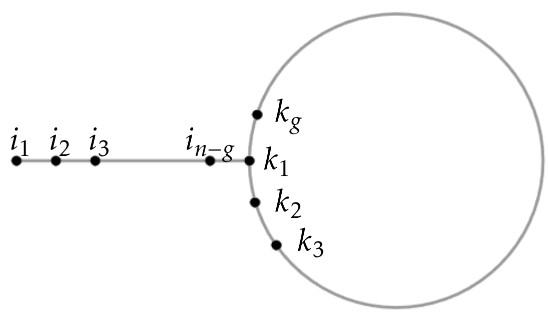


Figure 25. Generalized lollipop graph $C_{n,g}$

**Definition 6.1.** A *lollipop* graph $L$ will be a generalized lollipop graph with two vertices and girth 1, $L = C_{2,1}$. See figure 26.

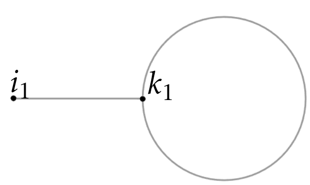


Figure 26. Lollipop graph $L = C_{2,1}$

We consider $L$ being a *metric graph* as defined by Bridson and Haefliger in [1, Ch. I.1 and I.5 Example 5.21]. With this metric, each edge has length one and the distance between two robots is given by the infimum of the lengths of paths joining them. We can say that the *interval edge* is $I$ and the *circle edge* is $S^1$. See figure 27 for examples of distances $d$ between robots.

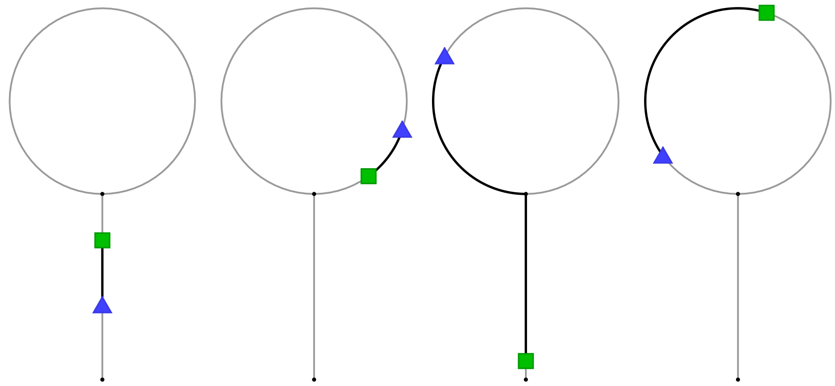

Figure 27. Distance between two robots

## 6.1. Configuration space.

The configuration space $X$ is given by:

$$X = C^2(L) = L \times L - \Delta.$$

where $L$ is a lollipop graph.

Figure 28 depicts $X$ and its flat representation. Recall that in the configuration space, each state represents the combined positions of the two robots in $L$, where the $x$-axis represents the positions of the first robot $A$, and the $y$-axis represents the positions of the second robot $B$ as in figure 29.

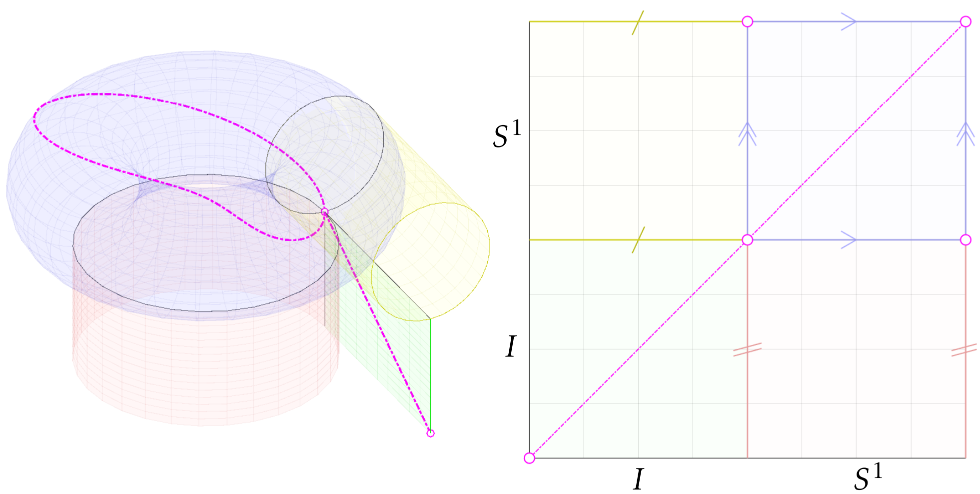


Figure 28. Configuration space $X$ and its flat representation

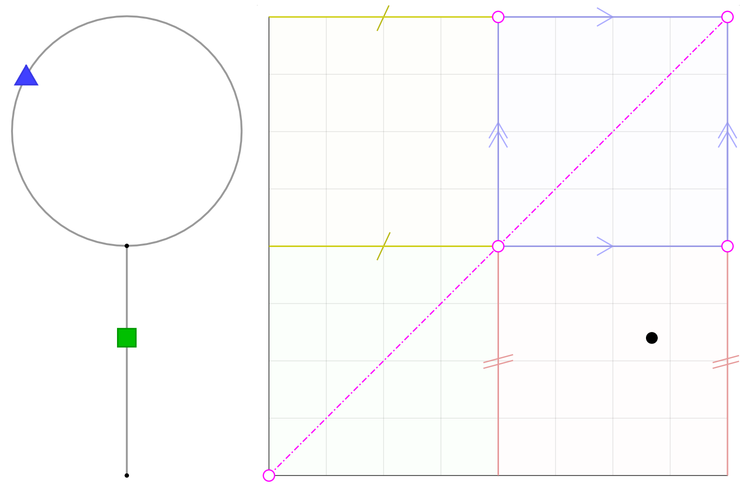

Figure 29. Physical positions and their state in configuration space

6.1.1. *Homotopy deformation.* Ghrist proved in [10] that the configuration space of any graph with $k$ vertices of degree greater than two, deformation retracts to a subcomplex of dimension at most $k$ . We have then that our configuration space is homotopy equivalent to a 1-dimensional space. Our motion planning algorithm will be derived while working in the configuration space, as we explained in the previous examples. The execution of the algorithm, however, will run in the physical space. We will choose a 1-dimensional space in the configuration space that has a clear meaning when projected back into the physical space.

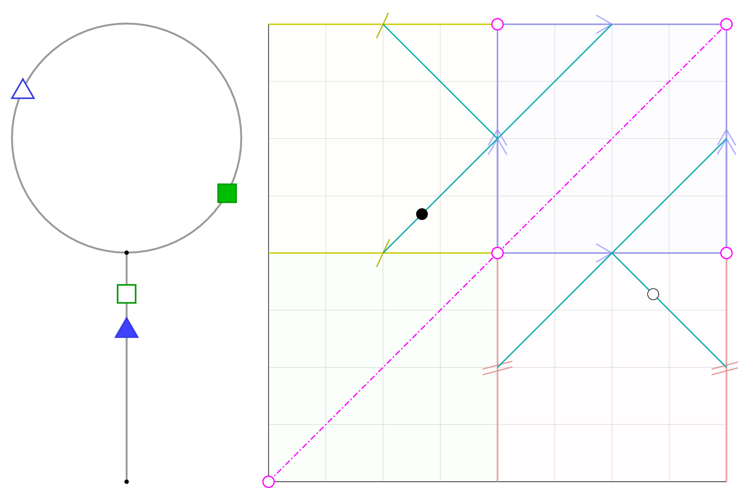

Figure 30. Two positions in the skeleton $S$

Following the same approach, this 1-dimensional space, that we will call the skeleton $S$, will be the antipodal circle as before with the addition of two more circles as in figure 30.

The homotopy $H$ deforming the configuration space $X$ into the skeleton $S$ is shown in figure 31. Note that based on the position of the two robots, the configuration space can be subdivided into three main regions as highlighted in figure 31: The blue regions are where the distance between the robots is larger or equal to half unit and the robots are in different edges. The yellow region is where the to robots are at half unit apart or less but not both at the interval edge. In the green region both robots are at the interval edge. This homotopy will be useful to transit in and out of the skeleton in the configuration space $X$.

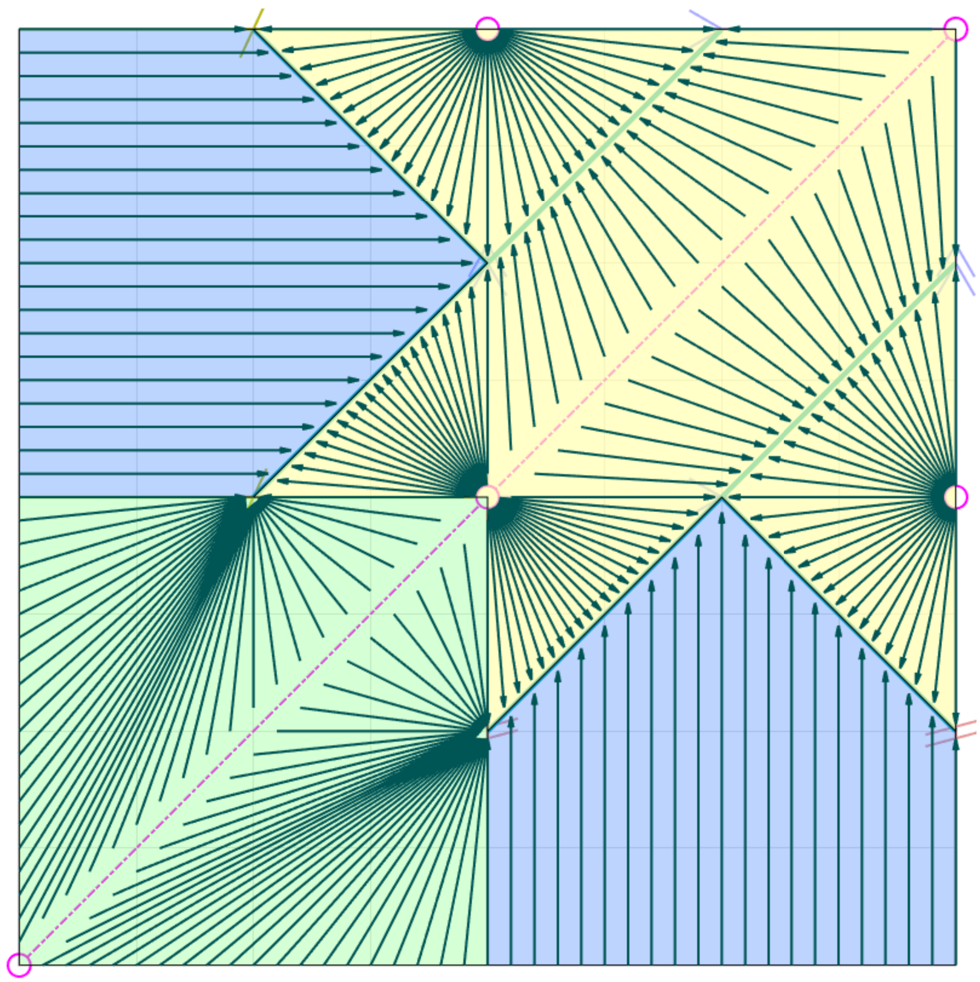

Figure 31. Traces of the homotopy deformation $H$

The skeleton $S$ is homeomorphic to a chain of three circles as shown in figure 32. By contracting the lower part of each circle, we can see that the skeleton is homotopy equivalent to a wedge of three circles $S^1 \vee S^1 \vee S^1$ as shown in figure 33.

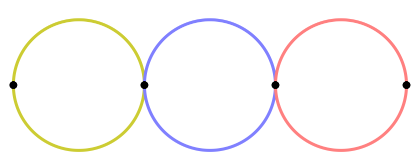

Figure 32. The skeleton $S$

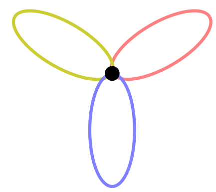

Figure 33. Wedge of three circles

6.2. **Topological Complexity.** In order to exhibit a set of instructions for the motion planning algorithm, we will consider the number of these continuous instructions given by the topological complexity of the configuration space, $TC(X)$. Farber calculated the topological complexity of all graphs based on their first Betti number.

In topological graph theory, the first Betti number of a graph $G$ with $n$ vertices, $m$ edges and $k$ connected components is given by $b_1(G) = m - n + k$.

**Proposition 6.2.** [6] *Let G be a graph, then*

$$TC(G) = \begin{cases} 1 & \text{if } b_1(G) = 0, \\ 2 & \text{if } b_1(G) = 1, \\ 3 & \text{if } b_1(G) > 1. \end{cases}$$

**Theorem 6.3.** *The topological complexity of the configuration space of two robots moving on a lollipop graph is three:* $TC(C^2(L)) = 3$.

*Proof.* We have shown that $C^2(L) \simeq S^1 \vee S^1 \vee S^1$. By proposition 4.7, we know that $TC(C^2(L)) = TC(S^1 \vee S^1 \vee S^1)$. Since the first Betti number of a wedge of three circles is $b_1 = 3 - 1 + 1 = 3$, we have that $TC(S^1 \vee S^1 \vee S^1) = 3$ by proposition 6.2. Therefore, $TC(C^2(L)) = 3$. □

We conclude that for two robots moving on a lollipop graph, any MPA would require at least three continuous instructions.

6.3. **Motion Planning Algorithm.** We proved previously that $TC(X) = 3$ if $X = C^2(L)$. Then we know that the minimum number of continuous local sections that we can define on $X \times X$ is 3. Recall that although the set of instructions of the algorithm will be derived in the configuration space, its execution will be run in the physical space.

If the state corresponding to the two robots' positions is on the skeleton $S$, we know that in the physical space the robots are one half unit apart from each other and no robot is in the first half of the interval edge $[0, \frac{1}{2})$. Recall that the distance between two robots in $L$ is the minimal length of the paths in $L$ joining them.

**Definition 6.4.** Two robots are in *generalized antipodal positions* in $L$ if the distance between them is equal to half unit and no robot is in the first half of the interval $[0, \frac{1}{2})$. See figure 34.

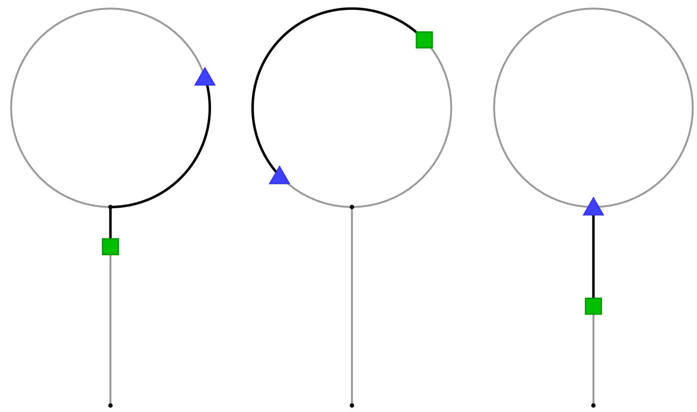

Figure 34. Generalized antipodal positions

Observe that the skeleton $S \subset X$ corresponds exactly to pairs of robots in $L$ that are in generalized antipodal position.

Consider the homotopy $H : X \times I \to X$ given by the projection into the skeleton $S$ following the traces of the homotopy as in figure 31.

Let $x_i$ and $x_f$ be the initial and final states of the robots. We call the *initial generalized antipodal* state $x_i' = H_1(x_i)$ the image of the initial state in the skeleton $S$ by the homotopy. Likewise, we call the *final generalized antipodal* state $x_f' = H_1(x_f)$ the projection per homotopy of the final state in $S$.

As before, the idea of our algorithm will be to move the initial and final states $x_i$ and $x_f$ to their generalized antipodal states $x_i'$ and $x_f'$ respectively, and then apply in the skeleton $S$ a new algorithm to move from $x_i'$ to $x_f'$ that we will explain next.

6.3.1. *Moving the states within the skeleton.* Recall that the skeleton $S$ is the graph shown in figure 35. Let $V = \{v_1, v_2, v_3, v_4\}$ be the set of vertices of the graph $S$. A vertex in the configuration space represents a pair of robots in the physical space in which one of the robots is at the center of the lollipop.

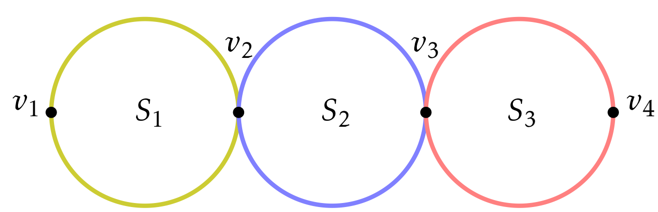


Figure 35. The skeleton $S$

The positions in the physical space corresponding to the vertices $v_1$,$v_2$,$v_3$ and $v_4$ in $S$ are characterized by having one robot at the center of the lollipop $L$ (figure 36).

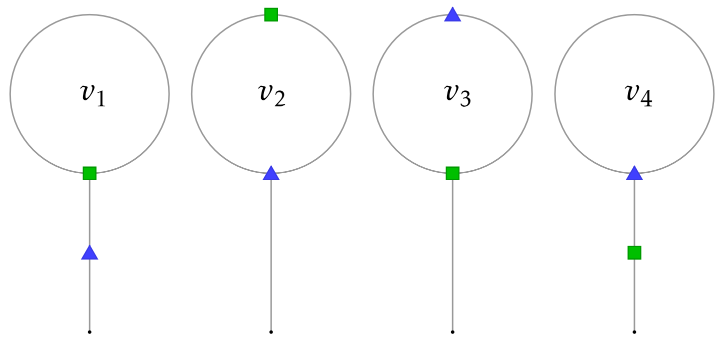


Figure 36. Vertex positions in $L$

**Definition 6.5.** If there is more than one path of minimal length from one state to another in $S$, we will say that *the preferred shortest path* is the one that traverses complete semicircles counterclockwise. Otherwise, the preferred shortest path is just the path of minimal length. See figure 37.

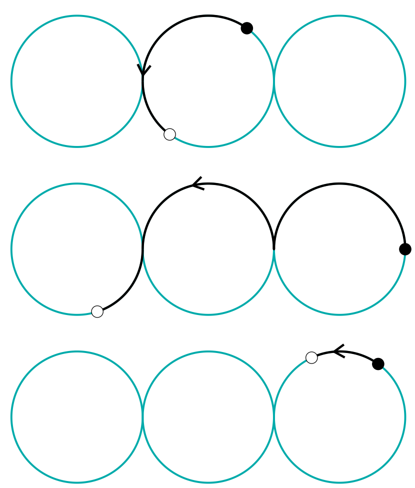

Figure 37. The preferred shortest path from initial to final states in $S$

Now our instruction will be simply: *go always following the preferred shortest path*. This instruction is obviously not continuous in $S \times S$. We will decompose the Cartesian product $S \times S$ in three regions over each of which our instruction will be continuous.

The first region $F_1$ consists of the pairs of points in $S$, initial and final, such that exactly one of them is a vertex (see figure 38):

$$F_1 = \{(x_i, x_f) | (x_i \in V \text{ and } x_f \notin V) \text{ or } (x_i \notin V \text{ and } x_f \in V)\}.$$

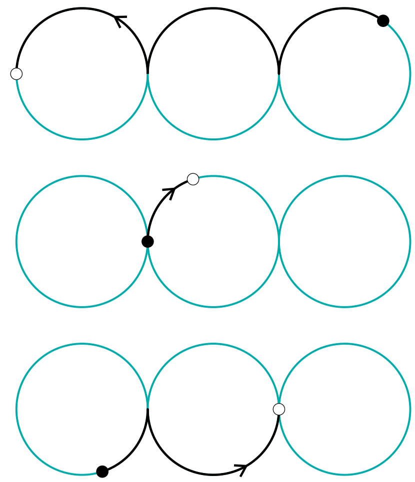

Figure 38. Motions in $F_1$

Let $A = \bigcup_{j=1}^{3} S_j \times S_j$. The second region consists of all pairs of points that are antipodal to each other in the same circle, or both are vertices (see figure 39):

$$F_2 = A \cup (V \times V).$$

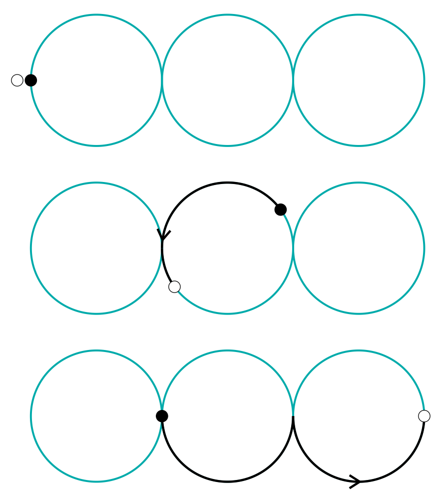

Figure 39. Motions in $F_2$

The third region is the rest of the Cartesian product, $F_3 = (S \times S) \setminus (F_1 \cup F_2)$. Explicitly $F_3$ is the set of pairs of states where initial and final states are neither vertices nor antipodal pairs (see figure 40):

$$F_3 = \{(x_i, x_f) \in S \times S \mid x_i \notin V \text{ and } x_f \notin V \text{ and } (x_i, x_f) \text{ are not antipodal pairs}\}.$$

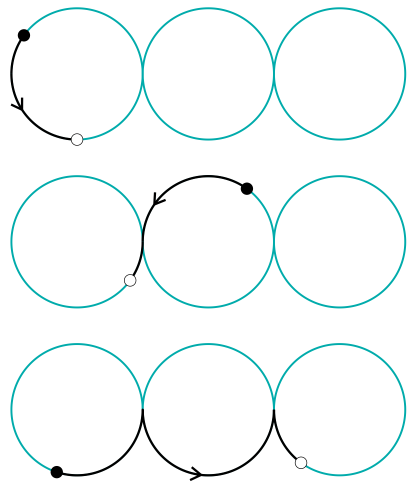

FIGURE 40. Motions in $F_3$

6.3.2. *Algorithm in the configuration space.* We need to extend this algorithm to the whole configuration space. We will do this by considering the traces of the homotopy shown in figure 31. The counterclockwise direction in each circle $S_j \subset S$ is shown in figure 41.

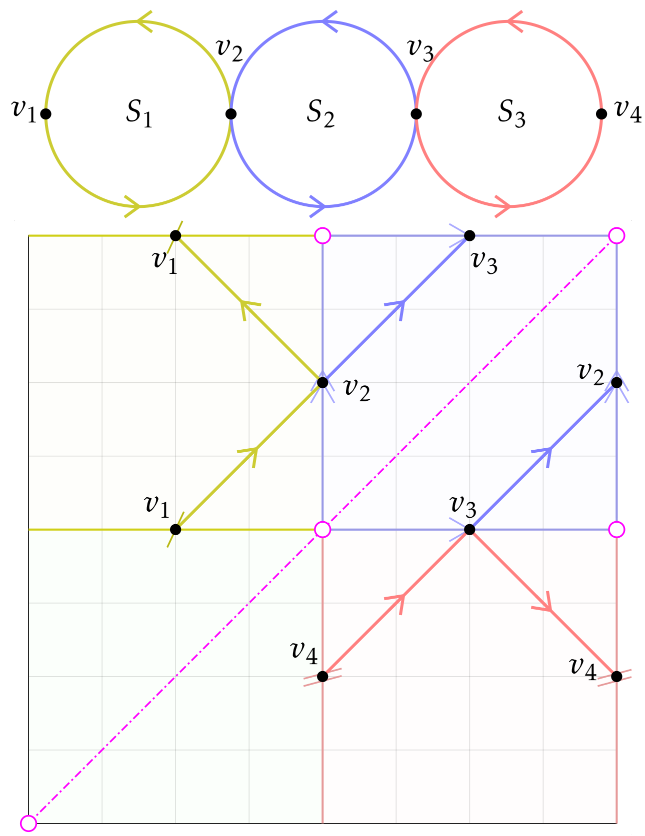


FIGURE 41. Counterclockwise directions in each circle $Sj \subset S$

The steps of the algorithm are as follows:

(1) Preliminary step: Move initial state $x_i$ to its generalized antipodal state $x_i'$ in $S$ following the path $H_t(x_i)$. Find the generalized antipodal state $x_f'$ in $S$.

(2) Main step: While on $S$, go from $x_i'$ to $x_f'$ following the preferred shortest path.

(3) Final step: Move the final antipodal state $x_f'$ to the final state $x_f$ following the reverse path of $H_t(x_f)$.

The three domains of continuity $F_1', F_2'$ and $F_3'$ are determined by the regions of the configuration space $X$ that end up in $F_1, F_2$ and $F_3$ respectively after applying the homotopy deformation, i.e. $F_j' = H_1^{-1}(F_j)$, for $j = 1, 2, 3$.

6.3.3. *Algorithm in the physical space L.* By translating the algorithm to the physical space we mean to describe our instruction in terms of the positions of the robots in the lollipop. We will say that the robots are *far from each other* if their distance is greater than half unit and they are *close to each other* if their distance is shorter than half unit. See figures 42 and 43.

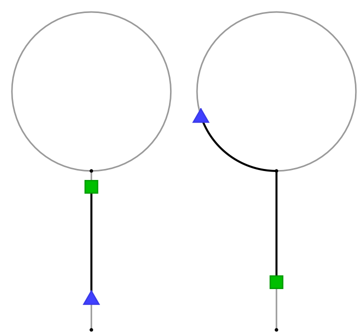

FIGURE 42. Distance greater than half unit: *far from each other*

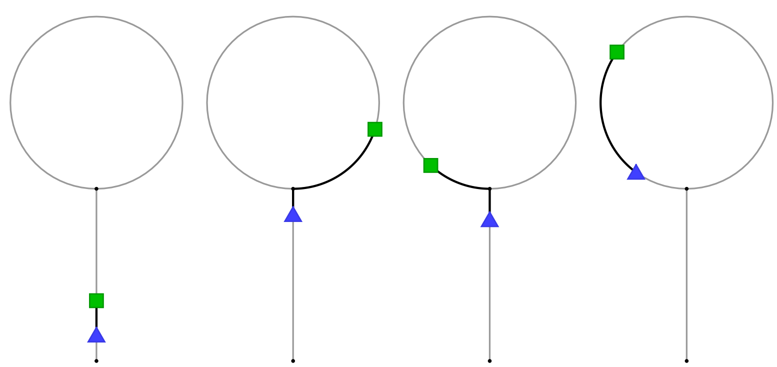

FIGURE 43. Distance shorter than half unit: *close to each other*

**Definition 6.6.** If there is more than one path of minimal length between two positions in $L$, we will say that *the preferred shortest path* is the one that traverses the complete semicircle counterclockwise. Otherwise, the preferred shortest path is just the path of minimal length. See figure 44.

The instructions of the motion planning algorithm in $L$ will be as follows:

(1) Preliminary step:

(a) If both robots are in the interval, then move whichever robot is closer to the center of the lollipop to the center following the shortest path. The other robot moves along the shortest path to the midpoint of $I$.

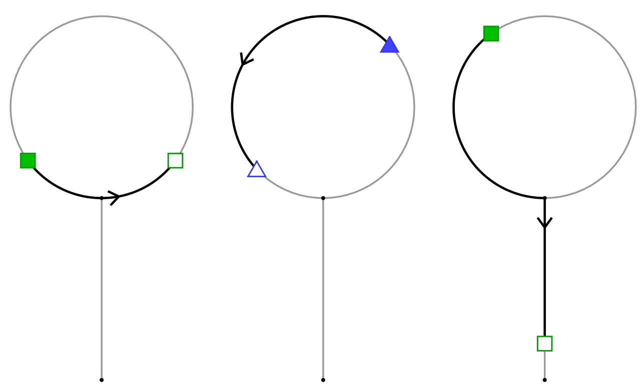

FIGURE 44. The preferred shortest path from initial to final positions in $L$

(b) If at least one robot is in the circle edge and

- they are far from each other, then move the robot in the interval towards the other robot (which remains stationary) until both robots are at generalized antipodal position.
- they are close to each other, then move both robots away from each other without changing edge, until both robots are at generalized antipodal position.

The ratio between the speeds at which the robots are moving is given by the slopes of the traces of the homotopy in figure 31. Apply this same procedure to determine the final generalized antipodal position.

(2) Main step:

(a) If initial and final generalized antipodal positions of each robot are in the same edge but are not all of them in the interval edge,

- move any robot with initial and final positions in the circle edge following the preferred shortest path to final
- the other one follows keeping half unit distance apart and staying in the edge where its final is.

(b) If one robot has generalized antipodal initial and final positions in the circle edge and the other robot in different edges,

- move the robot with initial and final positions in different edges following the preferred shortest path to final
- the other one follows staying in the circle, keeping half unit distance apart and choosing the shortest path when there are two choices.

(c) Otherwise,

(i)
- move the robot that is in the interior of the interval edge shortest path to center
- the other follows keeping half unit distance apart and choosing the preferred shortest path when there are two choices

(ii) move both counterclockwise until they swap positions

(iii) move both following the shortest path to final positions.

(3) Final step: Move both robots to their final positions following the shortest paths. Recall that the ratio of the speeds at which the two robots will be moving in this case is given by the slopes of the traces of the homotopy.

6.4. **Running the algorithm in the physical space $L$.** We will show now some sample cases of two robots moving from initial to final positions following our algorithm under various configurations. A series of demonstration videos of the algorithm implementation is publicly available online at `https://doi.org/10.6084/m9.figshare.12072183`.

(1) Example of motion in the same circle in $S$

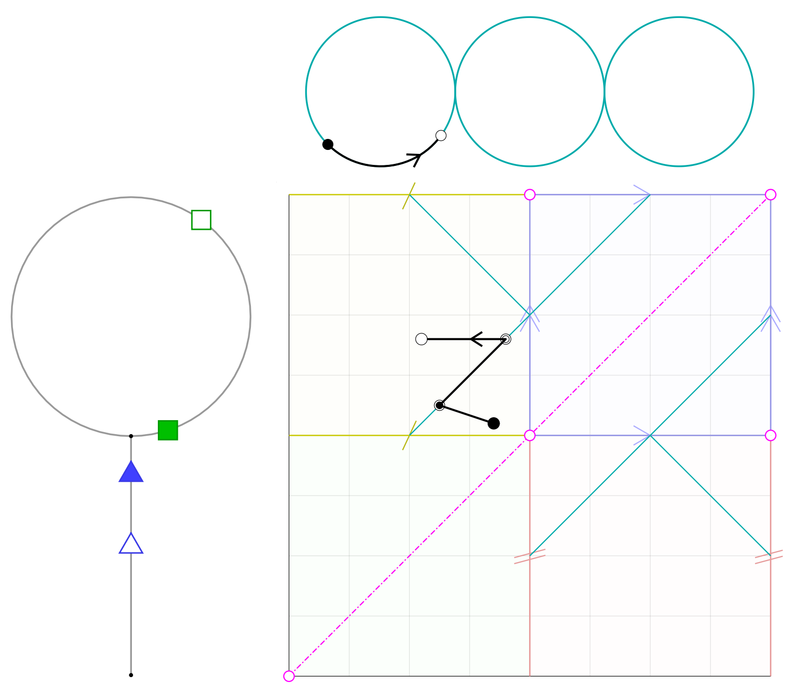

Figure 45. Configuration scenario in $L$ and the paths in $X$ and $S$

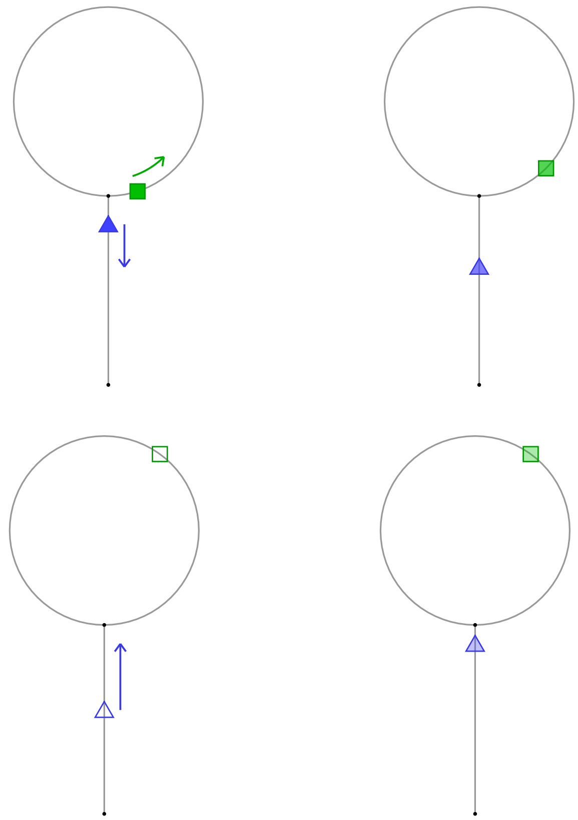

FIGURE 46. Preliminary step of the execution of the algorithm in $L$

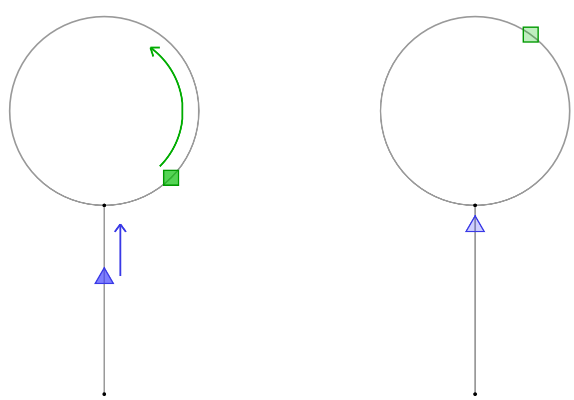

FIGURE 47. Main step of the execution of the algorithm in $L$

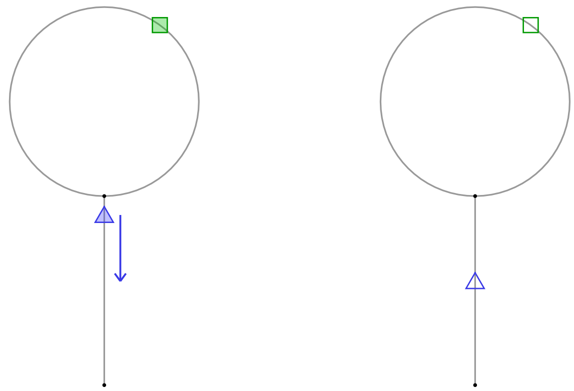

FIGURE 48. Final step of the execution of the algorithm in $L$

(2) Example of motion between adjacent circles in $S$

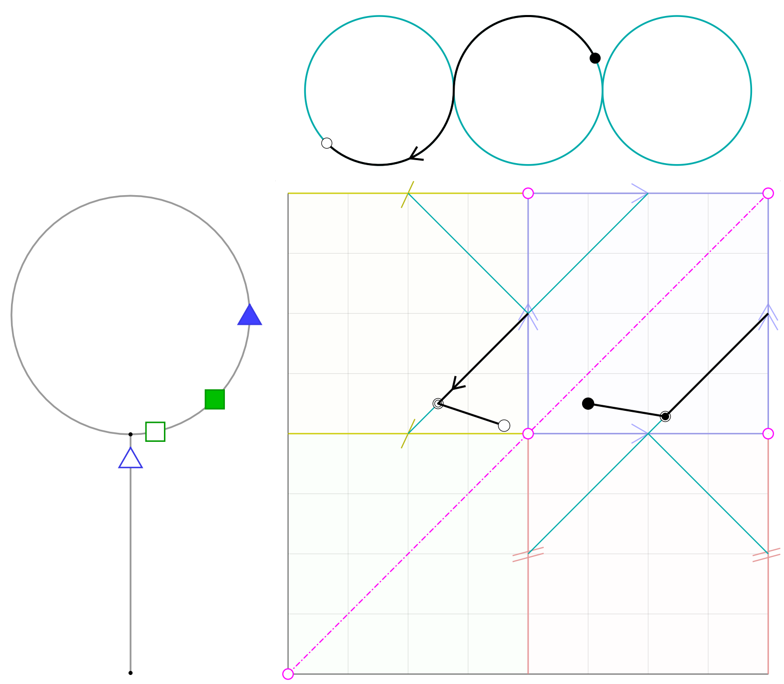

FIGURE 49. Configuration scenario in $L$ and the paths in $X$ and $S$

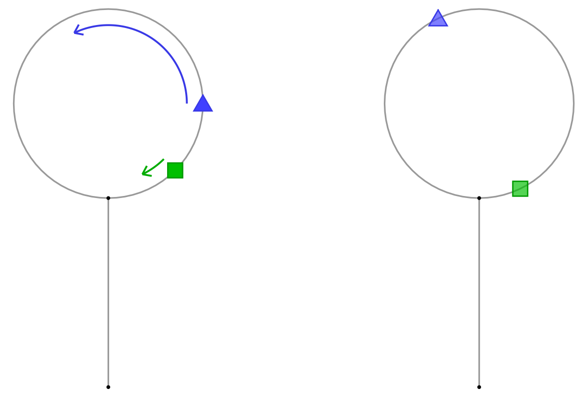

Figure 50. Preliminary step of the execution of the algorithm in $L$

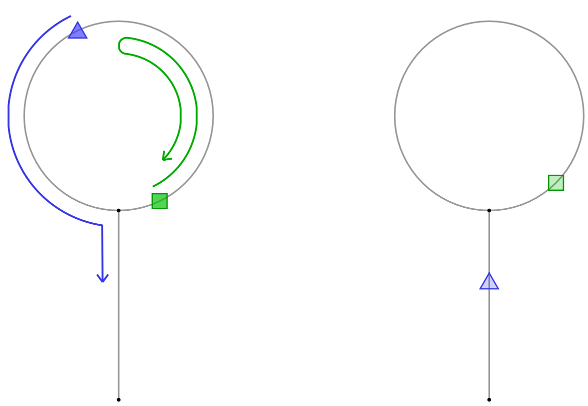

Figure 51. Main step of the execution of the algorithm in $L$

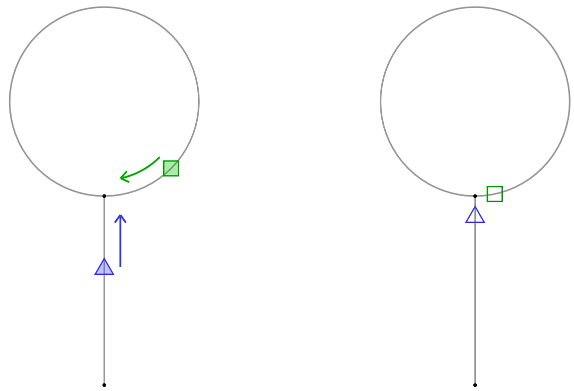

FIGURE 52. Final step of the execution of the algorithm in $L$

(3) Example of motion between disjoint circles in $S$

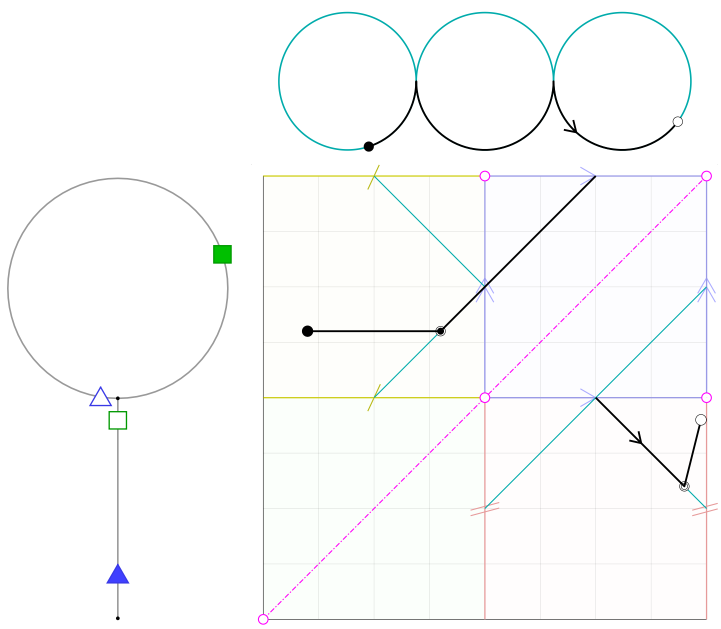

FIGURE 53. Configuration scenario in $L$ and the paths in $X$ and $S$

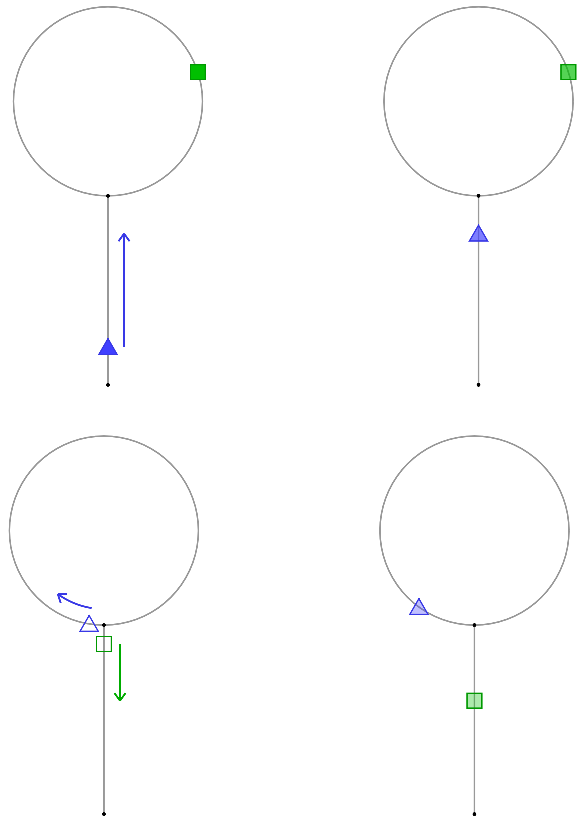

Figure 54. Preliminary step of the execution of the algorithm in $L$

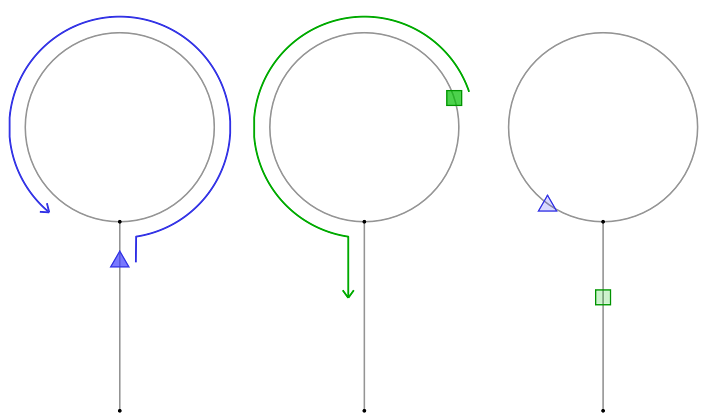

Figure 55. Main step of the execution of the algorithm in $L$

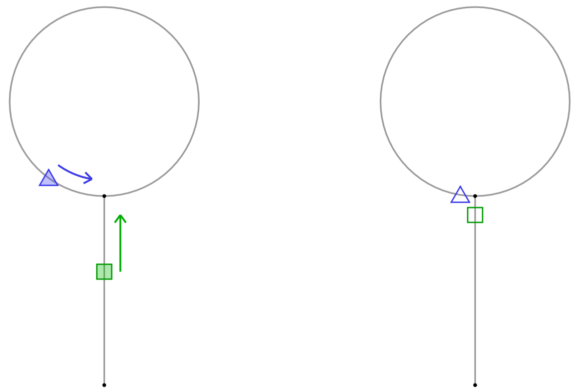

Figure 56. Final step of the execution of the algorithm in $L$


## Acknowledgements

Hellen Colman is supported in part by the Simons Foundation.


## References


[1] Martin R. Bridson and André Häfliger. *Metric Spaces of Non-Positive Curvature*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2011.

[2] Edward Fadell and Lee Neuwirth. Configuration spaces. *MATHEMATICA SCANDINAVICA*, 10:111–118, Jun. 1962.

[3] Shaun M. Fallat, Steve Kirkland, and Sukanta Pati. Minimizing algebraic connectivity over connected graphs with fixed girth. *Discrete Mathematics*, 254(1-3):115–142, 2002.

[4] Michael Farber. Topological complexity of motion planning. *Discrete and Computational Geometry*, 29(2):211–221, 2003.

[5] Michael Farber. Collision free motion planning on graphs. In *Algorithmic Foundations of Robotics VI*, pages 123–138. Springer, 2004.

[6] Michael Farber. Instabilities of robot motion. *Topology and its Applications*, 140(2-3):245–266, 2004.

[7] Michael Farber. Topology of robot motion planning. In *Morse theoretic methods in nonlinear analysis and in symplectic topology*, volume 217 of *NATO Sci. Ser. II Math. Phys. Chem.*, pages 185–230. Springer, Dordrecht, 2006.

[8] Michael Farber. Configuration spaces and robot motion planning algorithms. In *Combinatorial and toric homotopy*, volume 35 of *Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap.*, pages 263–303. World Sci. Publ., Hackensack, NJ, 2018.

[9] Robert Ghrist. Configuration spaces and braid groups on graphs in robotics. braids, links, and mapping class groups: the proceedings of Joan Birman's 70th birthday, ams. *IP Studies in Mathematics*, 24:29–40, 2001.

[10] Robert W. Ghrist and Daniel E. Koditschek. Safe cooperative robot dynamics on graphs. *SIAM J. Control Optim.*, 40(5):1556–1575, 2002.

[11] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[12] Tomas Lozano-Perez, Matthew Mason, and Russell H. Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984.

[13] James R. Munkres. *Topology*. Prentice Hall Upper Saddle River, NJ, 2000.

STUDENT BIOGRAPHY

**Allaoua Boughrira:** (*Corresponding author:* allaoua.boughrira@gmail.com) Allaoua graduated from Wilbur Wright College in Spring 2019 with a Computer Science major. He is currently working as a software engineer at a private firm. Albeit not affiliated with any higher education institution at the time, he intends to pursue his immense interest in mathematics as an independent researcher. He is interested in particular in algebraic topology and graph theory. His main drive is exploring the beauty of mathematics and its elegant tools to help derive practical solutions to complex problems in the real world.