

# TOPOLOGICAL COMPLEXITY FOR DRIVERLESS VEHICLES

RICKY SALGADO AND EMILIANO VELAZQUEZ

ABSTRACT. The topological complexity is a numerical invariant which measures the number of commands an autonomous robot needs in order to move in a space to perform a task.

To explain these ideas, we will walk through the various algebraic definitions and provide physical examples along the way. We calculate the topological complexity for various scenarios, starting with a single robot moving on a simple space, we will add on to this scenario towards a final case including more robots and more complicated spaces where they move.

Finally, we provide an example where two driverless vehicles move in a track joining seven colleges throughout Chicago. We determine the number of instructions as well as their content for this case study.

---

(Ricky Salgado, Emiliano Velazquez) WRIGHT COLLEGE, 4300 N. NARRAGANSETT AVENUE, CHICAGO, IL 60634 USA

(Hellen Colman, Advisor) DEPARTMENT OF MATHEMATICS, WRIGHT COLLEGE, 4300 N. NARRAGANSETT AVENUE, CHICAGO, IL 60634 USA

*E-mail address:* `rsalgado108@student.ccc.edu`, `evelazquez72@student.ccc.edu`, `hcolman@ccc.edu`.

## INTRODUCTION

We examine the idea of topological complexity in the context of robot motion planning through the manipulation of configuration spaces. These ideas were introduced to us during an undergrad project lead by Professor Hellen Colman<sup>1</sup>. We study the works of mathematicians such as Michael Farber and Robert Ghrist. We hope to show the applications of their theorems and conclusions in some real world scenarios.

*Motion planning* is currently one of the most active branches of robotics. Mathematics, computer science and engineering converge in the study of motion planning to create and implement algorithms for a set of robots to move under certain constraints.

The question addressed in this paper is that of deciding how many instructions are needed for a set of two driverless vehicles to move autonomously through a theoretical track joining all the City Colleges of Chicago, and describing what these instructions look like.

The set of two robots together with the track of a specific shape where they can move determine what is called the *configuration space* for our problem.

The motion planning problem in the configuration space  $X$  consists of a process that takes pairs of initial and final configurations  $(a, b) \in X \times X$  as an input and produces as an output a path in  $X$  from  $a$  to  $b$ .

The *topological complexity* of a space,  $TC(X)$ , is a numerical invariant that measures the discontinuity of this process. It reflects the complexity of the problem of choosing a path in a space  $X$  so that the choice depends continuously on its endpoints. The topological complexity is the number of rules (or instructions) required to specify how to move between any two points of  $X$ .

This number is in general hard to calculate. In fact, for some simple spaces like the Klein bottle, the topological complexity is still an open problem.

By using a fundamental result of Farber [3] which establishes that the topological complexity depends only on the homotopy type of  $X$ , our goal is to find a suitable shape for the track  $T$  such that the configuration space of two robots moving on  $T$  has the same homotopy type of a space with known topological complexity.

The City Colleges of Chicago is a network of seven colleges throughout Chicago. We first design a track that passes through the seven colleges, whose positions are fixed. Considering the way the streets and colleges are arranged, we can build a track in a shape homeomorphic to the letter  $T$ . Then we proceed to study the topological complexity of the configuration space of two robots moving on a  $T$ -track following the article of Ghrist on robots moving on different shaped graphs [1].

Finally we translate the theoretical results to our problem and describe the two instructions that we need to provide to the driverless shuttles to perform their task.

Through our research studying papers written on topological complexity, we have gathered enough to write about it ourselves and include some conclusions we drew. In doing so, we want to encourage anyone interested to add on to this new and promising area of study.

---

<sup>1</sup>HC is partially supported by a Simons Foundation Grant

ACKNOWLEDGMENTS

This paper is the result of an independent study course at Wilbur Wright City College supervised by Dr. Hellen Colman. The authors are sincerely grateful for her guidance and support throughout the development of this paper. The authors are also thankful to Marijan Peric for his help during discussions. Finally, the authors appreciate the previously mentioned college and Northeastern Illinois University for providing platforms to present this research.

1. PHYSICAL SPACE

In this paper we study motion planning for  $N$  robots  $r_1, \dots, r_N$  in a space  $\Gamma$ . The physical space  $\Gamma$  is simply the space in which the robots are allowed to be.

**1.1. Examples of Physical Spaces.** A basic example of a space in which robots could move is a segment  $I = [0, 1]$ . The robot could be anywhere between the ends of the segment. This segment is the physical space of the motion planning problem. So we write  $\Gamma = I$ .

We could also connect the two ends of the segment and have our robot move around in a circle  $S^1$ . Here we have no endpoints and our robot could move anywhere around the circle,  $\Gamma = S^1$ .

Our main example is to have the robot move in a  $T$ -shape track. We imagine the shape to be three segments with the beginning of each segment being connected. Here  $\Gamma = T$ .

**1.2. Number of Robots.** Increasing the number of robots in our physical space comes with the problem of their movements being restricted by each other. Since it is impossible to have two objects occupying the same space, we cannot have two robots in the same position in our physical space. For example, if we have two robots moving around on  $I$ , the robots are never able to switch positions. A robot in this space could also never reach the end past the other robot. This avoidance of collisions is a limitation on their movements. We want to write a strategy for the robots to move on their own without collisions. To this end we need to take into account the limitations given by the presence of another robot within the same physical space and the shape of the physical space itself. This is where the idea of *motion-planning* comes in. *Motion-planning* is the idea of having two points, initial and final positions, and finding some feasible path between them.

2. CONFIGURATION SPACE  $C^N(\Gamma)$

The physical space does not change with the number of robots, but the space of possible positions of the system as a whole does. The physical space stays the same because it is by definition where the robots are. With only one robot the physical space could also be described as the space in which the robot could move. With an added robot this space of possible positions changes. No longer can the original robot move anywhere it wants. The idea then is to come up with a representation of this new space of possible positions. That is, the space that specifies the position of every robot in the physical space. This notion will be called *configuration space*.

**2.1. Definition of Configuration Space.** Let's assume that a robot  $r_1$  can be placed anywhere in the physical space  $\Gamma$ . The set of points  $x_1$  that represents all possible states of  $r_1$  on the physical space is just  $\Gamma$ . That is, the configuration space of one robot moving on a physical space is just  $\Gamma$  itself. Now if we were to add one more robot,  $r_2$ , on the same physical space, the set of points that represent all of its theoretical states is the Cartesian product  $\Gamma \times \Gamma$ . Here the Cartesian product represents the idea of a *pair* of positions  $(x_1, x_2)$ , one for robot  $r_1$  and another for robot  $r_2$ . Since no two different robots can lie on the same point, we must subtract the pairs of points representing this scenario from our set. We call  $\Delta$  the set of positions where at least two robots lie on the same point. As we add more and more robots, we have to eliminate more and more positions. Formally, we define the configuration space to be  $C^N(\Gamma) := (\Gamma \times \Gamma \times \dots \times \Gamma) - \Delta$  where the diagonal  $\Delta := \{(x_1, x_2, \dots, x_n) \in \Gamma^N : \exists x_i = x_j, i \neq j\}$ .

**2.2. Examples of Configuration Spaces.** We now describe different configuration spaces corresponding to the physical spaces described in the previous section. As we change the number of robots in the same physical space, the configuration space becomes more complicated.

**2.2.1. Robots moving on an interval.** We know that for *one* robot in a physical space  $\Gamma = I$ , where  $I$  is some interval, the configuration space  $X = C^1(I)$  is equal to  $I$ ,  $X = I$ . We want to visualize what this configuration space looks like for two robots.

We know that each robot could be on any point on the interval. So to represent the position of each robot we get two full intervals, from the beginning to end, being multiplied. We can describe this using the Cartesian coordinate system where  $x$  represents the position of one robot and  $y$  represents the position of the other. The position in the coordinate system is then determined by how far along the interval the robots are on the physical space. So for this simple example we are able to visualize its configuration space as a square from which we have to subtract the diagonal  $\Delta$ . Therefore the situation of two robots moving on a physical space  $I$  is modeled by a configuration space  $X = C^2(I) = I \times I - \Delta$ . See Figure 3.

**2.2.2. Robots moving on a circle.** In the same way, the configuration space of one robot moving on a circle is  $X = C^1(S^1) = S^1$ . The configuration space for two robots moving on a circle, is  $X = C^2(S^1) = S^1 \times S^1 - \Delta$ . This gives us a torus with the diagonal  $\Delta$  making a curve around both circles of the torus once. If we cut the torus along this curve, we could bend the torus into a cylinder, where its circular edges, at both ends represent the diagonal  $\Delta$  that we cut off. See Figure 1 and Figure 2.

FIGURE 1. The black line represents the diagonal that is not part of the configuration space  $C^2(S^1)$

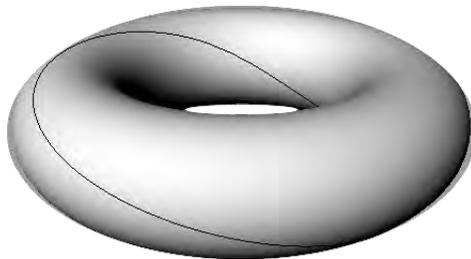
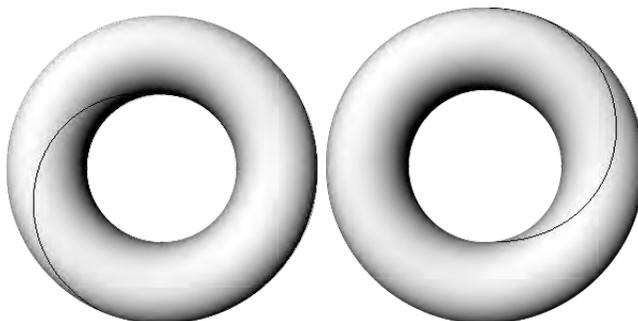


FIGURE 2. Top and bottom view of the configuration space  $C^2(S^1)$



### 3. PATHS IN CONFIGURATION SPACES

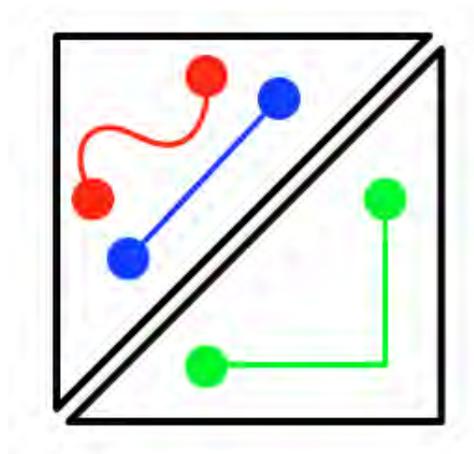
If we want to move  $N$  robots from an initial position to a final one, we need to describe a certain choreography that allows this movement to happen without collisions. Here is where working with configuration space is beneficial. Knowing the shape of the configuration space we can study the paths between different configurations that are possible. This translates to paths on the physical space that avoid collisions, which are our valid paths.

Suppose we have just one robot  $r_1$  in the physical space  $\Gamma$ . The configuration space  $C^1(\Gamma)$  is the same as the physical space,  $C^1(\Gamma) = \Gamma$ . The position  $x_1 \in \Gamma$  for the robot  $r_1$  has a corresponding state  $c_1 \in C^1(\Gamma)$  with  $x_1 = c_1$ . This holds true for the *motion* of  $r_1$ , as well. As we move it from some configuration to another, the path that  $r_1$  follows is the same both in the physical space  $\Gamma$  and in the configuration space  $C^1(\Gamma)$ . Now, if we add another robot  $r_2$ , then our configuration space changes from  $C^1(\Gamma) = \Gamma$  to a new space  $C^2(\Gamma)$  which is not  $\Gamma$  anymore. Our similarity does not carry over; we can no longer picture the spaces to be the same. This is due to the description of  $C^2(\Gamma)$  in which every point in the new space represents a configuration of two robots, so in particular our configuration space has higher dimension than  $\Gamma$ . As we move along some path in  $C^2(\Gamma)$  we are describing a collective change of configurations for  $r_1$  and  $r_2$ .

**3.1. Movements along  $C^2(I)$ .** We can start with our basic example of two robots moving on an interval  $I$  to visualize a path along the configuration space  $C^2(I) = I \times I - \Delta$ . As we have seen before, this configuration space is just the union of two

triangles. Since the space is not connected, the only possible paths are contained in one of those triangles. A path that connects a point on one triangle and another point on the other triangle does not exist. See Figure 3.

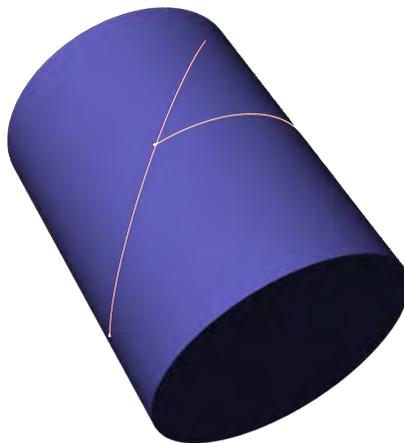
FIGURE 3. Examples of possible paths in  $C^2(I)$



So what does this mean in terms of movement on the physical space  $I$ ? Well, simply that we cannot have one robot move to an end of the interval with the other robot in its way. A path in the configuration space contained in one of the triangles corresponds to a choreography where both robots are moving from an initial configuration to a final one without involving any interchange of positions between them. The robots cannot perform any task involving swapping places just because there is not enough room in the interval for this to happen.

**3.2. Movements along  $C^2(S^1)$ .** Now we can go back to our example of two robots moving on a circle, whose configuration space  $C^2(S^1)$  is a cylinder. The robots can move on the circle in any direction so long as they do not collide. So our paths should be contained in the cylinder  $C^2(S^1)$ . We can see that this space is connected and there is always a path between any two points. Some of the possible paths are shown in Figure 4. All these paths correspond to movements of the two robots on the circle, from some initial position to a final one, without collisions. Observe that the shape of the physical space  $\Gamma = S^1$  provides more room for all the changes of positions to be possible now.

FIGURE 4. Paths along the configuration space  $C^2(S^1)$

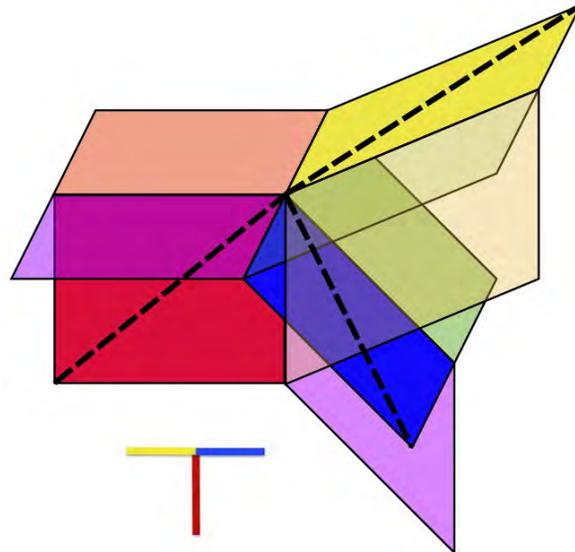


#### 4. MAIN EXAMPLE: $\Gamma = T$

**4.1. Configuration space  $C^2(T)$ .** We now consider the example of two robots moving on a track  $T$ . To construct the space  $C^2(T)$  we begin as follows. First, we build the cartesian product  $T \times T$ . This is the space of pairs of points where the first component belongs to the first space  $T$ , i.e. represents the position of the first robot. Likewise, the second component in the pair represents the position of the second robot.

We construct first a graph  $T$  and extrude it by a single interval corresponding to one of the edges of the second graph  $T$ . At this stage, the points of this space corresponds to the first robot being anywhere and the second being in that particular interval. Next we continue to extrude the first  $T$  by the next edge of the second  $T$  at a slight angle, to the right. Finally, we repeat the extrusion with the third edge to the left. The points on the surface obtained represent all theoretical positions of two robots on a space  $T$ . Our virtual modeling of the space as well as its physical plastic model helped us to visualize the space  $T \times T$  as a surface that cannot be embedded in the three dimensional space. See Figure 5.

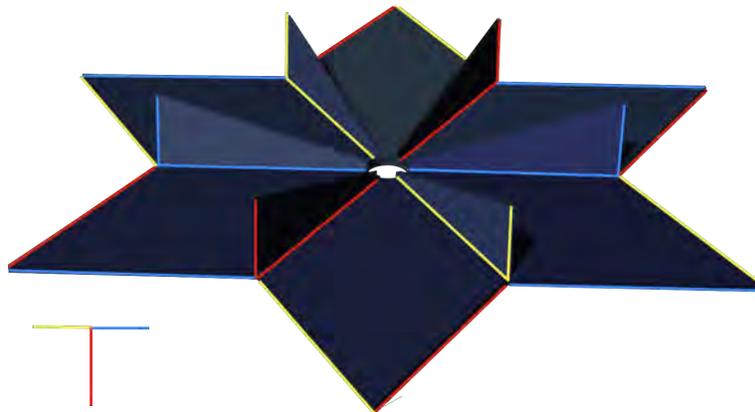
FIGURE 5.  $T \times T$ , where the dashed lines are the diagonal  $\Delta$



From this space of theoretical positions, we need to subtract the ones that are not possible: all the points in  $T \times T$  corresponding to both robots being exactly at the same place in  $T$ .

The diagonal  $\Delta$  in this case can be seen as the union of the diagonals of the squares corresponding to the robots being in the same edge. If we take out these set of points, we can then picture our  $C^2(T) = T \times T - \Delta$  as a surface that we can embed in the three dimensional space. See Figure 6. We actually built this space in Rhino [2], and printed a plastic model of it using a 3D-printer. See Figure 7.

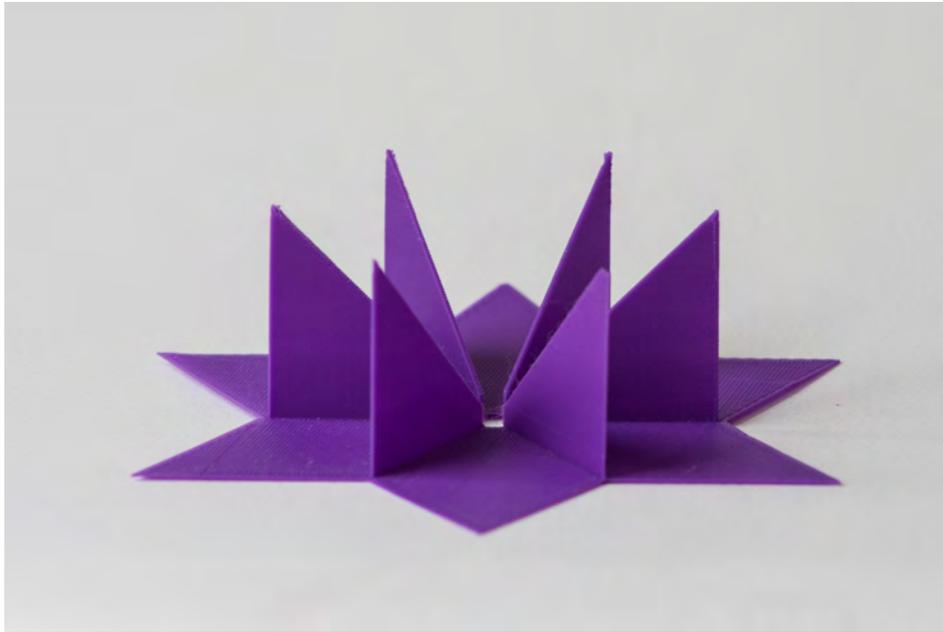
FIGURE 6.  $T \times T - \Delta$



We call this configuration space “flower” and refer to the quadrilaterals and triangles as “petals”. Topologically there is no difference between our original

$C^2(T)$  and our flower. In fact,  $C^2(T)$  is homeomorphic to the flower. Each square of  $T \times T$  that was cut in half by the diagonal is homeomorphic to the triangular petals, and the remaining uncut squares are the six quadrilateral petals of the flower. Notice that in the center of our flower there is a hole corresponding to the intersection of the three diagonals that we cut off. We refer to this hole as the *origin* of  $C^2(T)$ . Also notice that although both the product  $T \times T$  and the flower are two dimensional spaces, the flower can be fully viewed in a three dimensional space whereas the original space  $T \times T$  requires a four dimensional space to live in. We use this visualization of the flower to describe our movements along  $C^2(T)$ .

FIGURE 7. 3d-printed model of  $C^2(T)$



**4.2. Movements along  $C^2(T)$ .** Points in the triangular petals correspond to the positions in the physical space where both robots are in the same edge. The remaining quadrilateral petals correlate with the positions in which the robots are in different edges. Since the flower is connected, we can draw a path from a point to any other point in the flower.

For instance, if we place two points on the flower so that the first point lies on the outermost point on a petal and we place the final point on the outermost point of the petal that lies directly antipodal to the first, then we can draw a path in  $C^2(T)$  between these two points. The shortest path in the three dimensional space is a straight line segment that passes through these two points. The problem with this path however is that it passes through the origin in the middle, which is part of the diagonal  $\Delta$ . So, this path is not in  $C^2(T)$ . To avoid the origin, we can go around it in the left or right direction. Which do we choose? This will be part of the instructions that we will give to the robots.

In this case we do not really care, so we choose for instance the one that goes clockwise. The question now becomes, what does this mean physically, how does it connect to our physical space  $\Gamma$ ?

If we look at our initial point on the flower  $C^2(T)$  we observe that it is a point in the petal located the furthest away from the origin. This means that the point is furthest away from the diagonal in  $T \times T$  and as such the robots are furthest away from one another on  $\Gamma$ . We can imagine this to be each robot at the end of different intervals. Analogously, the final point in the flower corresponds to the robots being at the end of the same intervals, but now switched.

Then, as we move by the petal on our straight line towards the center, the two robots on  $\Gamma$  are moving closer to each other. When the robots get sufficiently close to that same intersection position in  $\Gamma$ , they have a choice either the first robot can stop and allow the second to move on ahead, or vice versa, the second stops and allows the first too move ahead. These possible choices correspond to the two possible curves in  $C^2(T)$  described before.

## 5. TOPOLOGICAL COMPLEXITY

**5.1. Continuity of the Motion Planning.** A *motion planning algorithm* is a set of rules that to each pair of positions (initial and final) assigns a path between them.

In our case, we want to provide the initial and final positions of our robots and we want to obtain the path that they should follow to move from the initial position to the final one, without collisions.

A path in a space is defined as a function from an interval to the space. A path in  $X$  between  $a$  and  $b$  is a function  $\alpha : I \rightarrow X$  such that  $\alpha(0) = a$  and  $\alpha(1) = b$ . Let  $PX$  be the space of all possible paths in  $X$ . The evaluation map  $\pi : PX \rightarrow X \times X$  assigns to each path its initial and final points, i.e.  $\pi(\alpha) = (\alpha(0), \alpha(1))$ . We say that  $s : X \times X \rightarrow PX$  is a section of  $\pi$  if  $\pi \circ s = id$ .

**Definition 5.1.** [3] A *motion planning algorithm* is a section  $s$  of the evaluation  $\pi$ . That is, a map  $s : X \times X \rightarrow PX$  which assigns to each pair  $(a, b) \in X \times X$ , a path in  $X$  between  $a$  and  $b$ .

This section does not always exist. For an example of a situation in which the section does not exist, consider the case of two robots moving on an interval  $I$ . Since we know that the configuration space  $X = C^2(I) = I \times I - \Delta$  is the disjoint union of two triangles, we cannot define any section  $s$  at a point  $(a, b)$  when  $a$  belongs to one of the triangles and  $b$  belongs to the other.

If a space is not connected, the section does not exist. Since there is nothing more to say about these spaces we will work with connected configuration spaces from here onward.

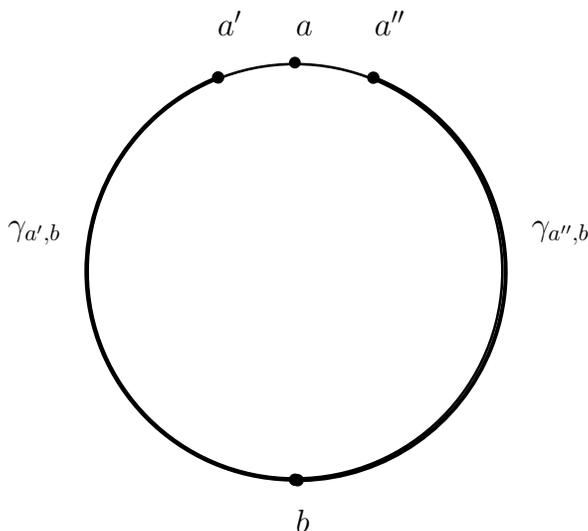
We are interested in motion planning algorithms in which the choice of the path is made in a continuous manner.

Informally, when  $X$  is a metric space, a motion planning algorithm  $s : X \times X \rightarrow PX$  is said to be *continuous* if and only if, for any small perturbations of initial and final positions, we obtain just a small change in the output path.

For an example of a section that is not continuous, consider one robot moving on a circle  $S^1$  ( $X = C^1(S^1) = S^1$ ) and the motion planning  $s : S^1 \times S^1 \rightarrow PS^1$  that assigns to each pair of positions, the shortest path  $\gamma$  between them. Let's say we

have initial and final points  $(a, b)$  at antipodal positions in the circle  $S^1$ . If we ask the motion planner to give us a path between these points, it gives us a clockwise path  $\gamma_{a,b}$ . But if we move  $a$  slightly over to  $a'$  or to  $a''$ , the motion planner redirects the path to now give us two entirely different paths  $\gamma_{a',b}$  and  $\gamma_{a'',b}$  that follows the opposite side of the circle as in Figure 8.

FIGURE 8. Discontinuity



There is a discontinuity in the case of this motion planner for a robot moving on a circle.

**5.2. Homotopy Invariance.** We will briefly discuss the idea of homotopy [4]. Two spaces are *homotopically equivalent* if one can be continuously deformed into the other. We denote this homotopy equivalence as  $X_1 \simeq X_2$ . For instance, a disk  $D$  is homotopically equivalent to a point  $p$  and a cylinder  $S^1 \times I$  is homotopically equivalent to a circle  $S^1$ . We write  $D \simeq p$  and  $S^1 \times I \simeq S^1$ .

**Definition 5.2.** A space  $X$  is contractible if it is homotopically equivalent to a point.

For example, a disk  $D$  is contractible.

Farber proved that there does not exist any continuous section in the case of the circle.

**Theorem 5.3.** [3] *A continuous motion planner  $s : X \times X \rightarrow PX$  exists if and only if the space  $X$  is contractible.*

In the example described above of a robot moving on a circle, the configuration space  $X = S^1$  is not contractible and the motion planner needs another rule to account for all antipodal points, which takes us to the idea of topological complexity.

**5.3. Topological Complexity.** In our example of a motion planner for one robot on a space  $S^1$ , we found that we needed two rules to cover all possible points in  $X \times X$ , such that the local section is continuous. The minimum number of these rules is called topological complexity.

**Definition 5.4.** [3] The topological complexity  $TC(X)$  is the least integer  $k$  such that  $X \times X$  may be covered by  $k$  open sets  $\{U_1, \dots, U_k\}$ , on each of which there is a continuous section  $s_i : U_i \rightarrow PX$  of the evaluation map  $\pi$ . These sets  $U_i$  are called domains of continuity.

For our robot moving on a circle, these sets consist of the two subsets  $U_1$  and  $U_2$ , described below. One set covers all antipodal points, the other covers all other points.

- (1)  $V_1 = \{(x, y) | x \text{ and } y \text{ are antipodal points}\}$
- (2)  $U_2 = \text{all other pairs}$

Let  $U_1$  be a small neighborhood of  $V_1$ . Then  $\{U_1, U_2\}$  is the covering by domains of continuity.

So our topological complexity  $TC(S^1) = 2$ , since we know already that is not 1 because  $S^1$  is not contractible. Farber studied the topological complexity for all spheres:

**Theorem 5.5.** [3]  $TC(S^n) = 2$  if  $n$  is odd or 3 if  $n$  is even.

In our case  $n = 1$ , so the previous theorem guarantees that the topological complexity is equal to 2.

**5.4. Homotopy Invariance.** We will briefly discuss the idea of homotopy. Two spaces are *homotopically equivalent* if one can be continuously deformed into the other. We denote this homotopy equivalence as  $X_1 \simeq X_2$ . For instance, a disk  $D$  is homotopically equivalent to a point  $p$  and a cylinder  $S^1 \times I$  is homotopically equivalent to a circle  $S^1$ . We write  $D \simeq p$  and  $S^1 \times I \simeq S^1$ .

Farber proved that the topological complexity of a space is an invariant of homotopy type. That is, if we can deform a space into another which is homotopically equivalent, the topological complexity of the spaces does not vary.

**Theorem 5.6.** [3] If  $X_1 \simeq X_2$ , then  $TC(X_1) = TC(X_2)$

This theorem allows us to calculate the topological complexity of any space with the same homotopy type than a circle, since we know already that  $TC(S^1) = 2$ .

## 6. TOPOLOGICAL COMPLEXITY OF $C^2(T)$

We know that the topological complexity of one robot moving in a circle,  $TC(C^1(S^1))$ , is 2 because the configuration space is again the circle  $C^1(S^1) = S^1$  and we showed that  $TC(S^1) = 2$ . For more complicated configuration spaces it is harder to calculate the topological complexity. We can use Farber's homotopy invariance theorem to this end. Recall that if two configuration spaces are homotopically equivalent, then their topological complexity is the same. We can say that if  $C^2(\Gamma) \simeq X$ , then  $TC(C^2(\Gamma)) = TC(X)$ .

So we can try to homotopically deform the space  $C^2(\Gamma)$  into another space  $X$  whose topological complexity we already know. To visualize this we can imagine the flower  $C^2(T)$  pictured in Figure 6 being squished down to form a circle. We can bring down the triangles to give you a flat shape and because the space has a hole

in the middle, we can deform it into the circle we want. Using Farber’s theorem, we conclude that our configuration space of two robots moving on a  $T$ -track has topological complexity of 2, since it is homotopically equivalent to  $C^1(S^1)$ .

**6.1. Domains of Continuity.** Consider the embedding of the flower in  $R^3$  as in Figure 6. By identifying the flower with the image of this embedding we can say  $F \subset R^3$ .

Let  $\bar{F}$  be the projection of  $F$  on  $R^2$ , .i.e.  $\bar{F} = \{(x, y) \in R^2 | (x, y, z) \in F\}$ .

We call *ray* in  $\bar{F}$  the half-line  $l$  determined by the origin and any point in the projection of the flower  $F = C^2(T)$ . The *antipodal ray* of  $l$  is the half-line  $l'$  determined by the origin and the antipodal point of any point in  $l$ .

Since we proved that  $TC(F) = 2$ , we know that there is some discontinuity in the motion planning for this configuration space. Before, for  $C^1(S^1)$ , we were able to describe the domains of continuity by the two subsets over which there is a continuous section  $s$ . One subset contains all pairs of antipodal points and the other contains all other pairs. Since we know that the flower  $F = C^2(T)$  can be deformed into  $S^1 = C^1(S^1)$ , we can translate these sets into the configuration space  $C^2(T)$  homotopically. So the subset containing all antipodal pairs extends to  $C^2(T)$  to be the subset of pairs of points that lay on antipodal rays in the flower.

These points in the flower  $F$  translate to several different scenarios in the physical space  $T$ . Given an initial configuration (pair of points representing position of each robot), the goal is to identify what final configurations in the physical space correspond to the points laying in antipodal rays.

The initial position of the robots determines a point  $P = (a, b, c)$  in the flower  $F$ . We have a ray  $l$  joining the origin and the projection of the point  $P$ .

We call *antipodal set* of  $P$ ,  $A(P)$ , the set of points in the flower whose projection is in the antipodal ray  $l'$ ,

$$A(P) = \{(x, y, z) \in F | (x, y) \in l'\}$$

Then we describe our first set of initial and final configurations  $U_1$  as a small neighborhood of  $V_1 = \{(P, P') \in F | P' \in A(P)\}$ . The set  $U_2$  contains all other pairs.

To identify the corresponding positions in the physical space, we observe that if the initial and final configurations are in antipodal petals of  $F$ , then the four positions in the physical space corresponding to the initial and final positions of the first and second robots must lie in a single leg or in the union of *two* legs of the track  $T$ . We can consider the latter union of two legs as a single interval. We then describe the antipodal positions in terms of their order in the interval:

- (1) The initial position of robot 1, then the final position of robot 2, then the final position of robot 1, then the initial position of robot 2.
- (2) The final position of robot 1, then the initial position of robot 2, then the initial position of robot 1, then the final position of robot 2.

with exact placements determined by the slope given by initial positions of both robots. We refer to these two conditions as the *antipodal conditions*.

So, the subset  $U_1$  contains these pairs of initial and final positions. The other subset  $U_2$  contains all other pairs.

**6.2. Motion Planning Algorithm.** So we can say that for our problem of planning the movement of two robots on a  $T$ -track from any initial positions to any

final positions, we need the *two* continuous motion planners,  $U_1$  and  $U_2$ , explained above.

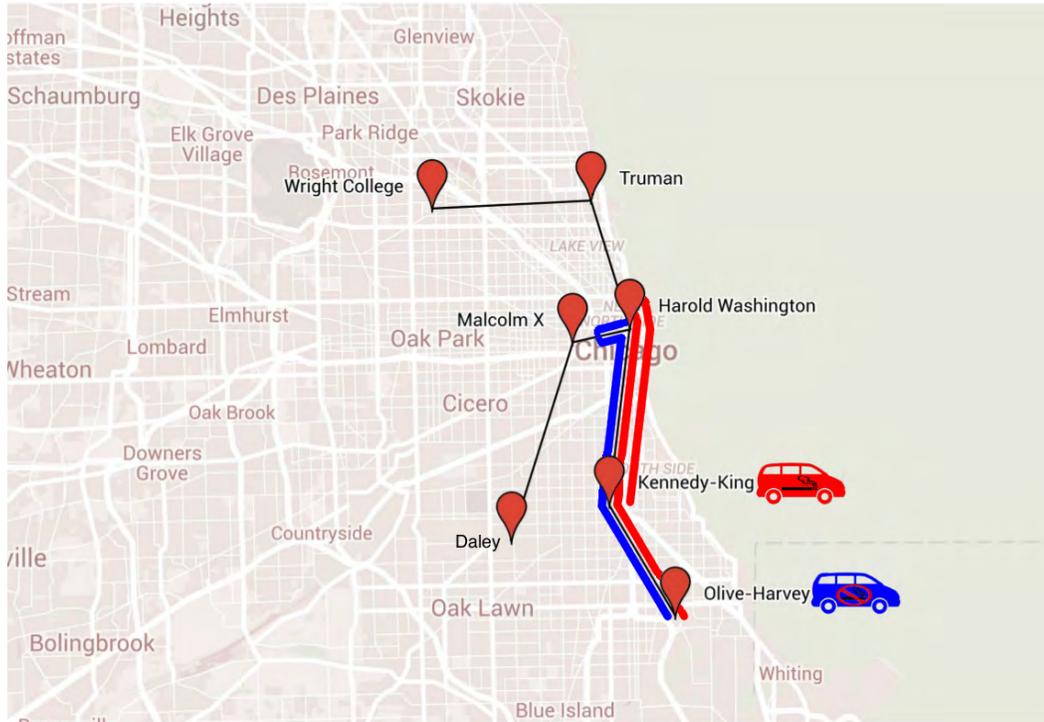
Each robot knows how to move from any initial position to any final one. Given these two positions, it would first check if any of the antipodal are met. If this is not true, the planner would tell the robot to take the shortest path on the configuration space  $C^2(T)$ , giving us the shortest distance traveled in the physical space  $T$ . If the contrary is true, then the planner tells the robots to take the shortest path that goes clockwise on  $C^2(T)$ .

On the physical space, this translates to something like a right-of-way rule where the robot that is on the right-most leg always moves into the unoccupied leg first. If both robots occupy the same leg, the robot closer to the origin moves into the unoccupied, right-most leg first. This results in the shortest possible distance traveled by the robots.

### 7. CITY COLLEGES OF CHICAGO DRIVERLESS SHUTTLE

To conclude, we take a look at our real world example as an application of *motion planning* of  $C^2(T)$ . First, we let all the colleges be joined by a  $T$ -track as in Figure 9.

FIGURE 9. City Colleges of Chicago



Then, we place two distinct robot shuttles in the track. One shuttle is red and will transport only smokers. The other is blue and will only transport non-smokers. Any smoker or non-smoker is allowed to jump into their respective shuttles at any point on the track. Once in the shuttle, the passenger will tell it where he wants to go. The two shuttles are able to transport the passengers using the shortest route while avoiding collisions.

We know from the previous section that the *motion planning algorithm* needed to accomplish this requires only two instructions. The instructions need to account for all possible initial and final positions for both shuttles. The initial positions for the shuttles are wherever the shuttles were left last. The final positions are wherever the passengers would like to go. Because of this the final position of one instance will always be the initial position of another. Using these bases, we can describe the two instructions as such:

- (1) If the antipodal conditions are not met for the shuttles, take the shortest path.
- (2) Otherwise, the shuttles follow a right-of-way rule, where the right-most shuttle moves into the unoccupied leg first.

The two shuttles can now transport their respective passengers around autonomously, using this basic algorithm. For instance, the red car is transporting a smoker from Kennedy-King to Olive-Harvey and the blue car is transporting a non-smoker from Olive-Harvey to Harold Washington at the same time. Given our algorithm, the red car would move into the leg between Harold and Wright, so as to let the blue car move into the leg between Harold and Daley. The red car would then move to deliver the smoker to Olive-Harvey and the blue car would deliver the non-smoker to Harold.

#### REFERENCES

- [1] A. Abrams and R. Ghrist. Finding topology in a factory: configuration spaces. *Amer. Math. Monthly*, 109:140–150, 2002.
- [2] Development Team. *Rhinoceros: design, model, present, analyze, realize...* Robert McNeel and Associates, Seattle, USA, 2015.
- [3] M. Farber. Topological complexity of motion planning. *Discrete and Computational Geometry*, 29(2):211–221, 2003.
- [4] Hatcher, A. Algebraic Topology *Cambridge University Press*, 2002.