

DEPARTMENT OF MECHANICAL ENGINEERING



Milestone II for Ph.D. Program

Discrete Path Planning For Platonic Solids

By NGOC TAM LAM
19107262

Dr. Lei Cui (Supervisor)
Prof. Ian Howard (Co-Supervisor)
AProf. Jonathan Paxman (Chairperson)

March 30, 2020

Contents

1 ABSTRACT	1
2 LITERATURE REVIEW	1
3 MODEL DESCRIPTION	2
4 ALGORITHM\METHODOLOGY	4
4.1 Path Planning Based Rolling Contact	4
4.2 Tree Exploration Algorithm	6
5 EVALUATION	8
5.1 Cube solid	8
5.2 Tetrahedron solid	11
5.3 Octahedron solid	12
5.4 Icosahedron solid	14
5.5 Dodecahedron solid	16
5.6 Result comparisons	18
6 CONCLUSIONS	20
7 ETHICAL ISSUES	20
8 COMPLETION TIMELINES	20
9 DISSEMINATION PLAN	20
A APPENDIX	22

1 ABSTRACT

Background: Rolling geometry between two regular rigid surfaces of objects in continuous space can generate paths, which is a well-known nonholonomic system. However, obtaining the path planning of a compact-closed surface on a general surface is challenging. From the rolling contact viewpoint, path-finding of convex polyhedrons has not received enough attention, and their paths on discrete surfaces have remained unexplored.

Aims: The aim of this project is to develop path planning algorithms for rolling polyhedrons on a surface.

Approaches: The approach presented in this paper is based on implementing the tree exploration technique to solve the closed-path planning with regards to changing the orientation with free-obstacles. The graph search starts from a root node, then expands to different branches which indicates multiple levels of rolling. The tree's branches expand until achieving the target configuration. A close-path planning can be executed with the Rodrigues' rotation matrix for rolling the platonic solids on discrete surface.

Conclusions: The results show multi-suboptimal closed-paths based on rolling for the platonic solids to achieve the desired configuration on the known environments. The results also show success in reaching the closed-path after combining with the point-to-point path planning technique in terms of changing orientation of the polyhedrons. Matlab simulations are carried out to demonstrate the effectiveness of the proposed path-finding algorithm for the regular polyhedra.

Implications: The improvement graph search method for the closed-path planning algorithm based on rolling can be extended to any convex polyhedrons.

2 LITERATURE REVIEW

Introduction: Path planning algorithm is one of the challenging problems in nonholonomic systems which, when resolved achieves the dexterous manipulation of objects in an unknown or part-known environment. This problem is mainly applied to the fields of robotics, artificial intelligence and autonomous vehicles. In robotics, the motivation of path planning is to find a possible path from an initial configuration, avoiding the obstacles and achieving the goal configuration [1]. Based on the task of robot performance, there are mainly two kinds of planning including feasibility and optimality. The former is to find a plan for only achieving the path while the latter is to find an optimal path. In the artificial intelligence fields, searching for actions to attain the desired goal state with receiving reward is employed, including decision-theoretic methods. Each specific path planning algorithm is usually implemented in a parameter space, such as configuration space or free space, which generates the feasible path connecting the two given points. Defining the state space is also one of the important steps for planning purpose. The configuration space or C-space which includes all possible configurations in a physical system is applied for solving path planning problems in n-dimensional. Examples of solving the path planning problems from Lavelle [2] and Kavraki [3] presented the feasible paths avoiding obstacles in the high dimensional configuration space.

Rolling contact: Rolling contact between rigid bodies has been considered as nonholonomic systems in order to solve the problem of dexterous manipulation of industrial parts. The goal of rolling manipulation is to roll the part from an initial configuration to the goal configuration. It can be divided into three types of rolling contacts including point contact [4], [5], line contact [6] and surface contact [7]. A simple experiment of a rolling polyhedral part on a table, mentioned in [8], showed that object manipulation with polyhedral surfaces without sliding can be executed by nonholonomic constraints through rolling. Some cases of rolling polyhedral objects through graspless manipulation have been studied in the robotics field [9], [10]. Due to the lack of complete research of contact kinematics and rolling manipulation with discretized objects, planning for rolling polyhedral parts under reorientation with smooth and non-smooth systems still attracts attention from the research community.

Discrete path planning in continuous space: Planning techniques are categorized into different aspects. The basic idea of discrete path planning in most cases is that state-space models will be used to demonstrate the distinct situation in which the task of a planning algorithm solves the sequence actions, transforming from an initial state to other states [11]. For example, Thomas [12] applied Delaunay triangulations to discretize the environment, and cubic spline representations are proposed to meet robot kinematic constraints. Considering other research on geometry, the paper [13] studied the continuous curvature on smooth curves which has been integrated within the probabilistic approaches in order to compute the piecewise smooth paths for a car-like vehicle as a four-dimensional system. Whereas dealing with nonholonomic constraints, a sampling-based road map technique has been proposed in [14], which

determined trajectories and re-entry trajectories for hovercrafts and rigid space crafts. Based on decomposing space into cells [15], a potential field without local minima was assigned with polygonal partitions of planar environments to solve the Laplace's equation problems in each presence cell. Applications for these techniques in discrete space is limited by a grid.

Polyhedra path planning: Not much work has been done in path planning considering rolling contact constraint. Some types of moving polyhedral parts have been investigated on a plane such as sliding on a face, tumbling through the edges or pivoting [9] through the vertex. The planning motions of rolling polyhedral parts through the edges were clearly represented in [16]. This paper presented some results about changing an orientation of a polyhedron through its edge's contact on a fixed plane without slipping. Experimental works from the article demonstrated that the manipulation of rolling polyhedron on a plane where the set of configurations has different structures of the polyhedral parts can be reached by rolling through its edges. In the experimental validation, a unit cube will reach the next position by rolling along the edges on a square mesh considering the given tolerance which leads to reaching an orientation closer. The paper also proposed a concept of path planning algorithm with the tolerance to achieve the goal configuration, which was considered as an important condition to generate an accurate path. However, the practical application may not be successful on robot manipulation.

Marigo [17] proposed the path planning for polyhedron in the case of an octahedron with eight faces rolling and translating on a plane. For the octahedron rolling algorithm, a list of faces containing the vertices and edges stored parts of the polyhedron. The defect angles are also computed between two connected faces. The algorithm initialized a start configuration, a desired final configuration, and given a polyhedron with a set of geometrical parameters. The steps of planning include displacing and reorienting the polyhedral part until achieving the final configuration. Nevertheless, the algorithm may not satisfy with the accuracy for other types of convex polyhedrons.

Conclusion: Therefore, in this study, we propose a closed-path planning algorithm based on the tree exploration method for the five types of platonic solids including cube, tetrahedron, octahedron, icosahedron, and dodecahedron. The path-finding algorithm focuses on rolling a solid on the associate grid from an original position to its initial position with different orientation. The study is organized as following. Section 3 covers the general properties in geometrical aspect of the five types platonic solids. Section 4 describes the closed-path planning algorithm based on tree exploration technique. Finally, section 5 shows the results of the proposed path planning algorithm under considering their different geometrical properties, then comes the conclusion of the paper.

3 MODEL DESCRIPTION

Platonic solids: The platonic solids are also called regular polyhedra have the convex polyhedra properties. There are exactly five regular polyhedra namely cube, tetrahedron, octahedron, dodecahedron and icosahedron (Figure 1). Some of the equivalent statements are used to describe the platonic solids, including all the vertices lie on a sphere, all the dihedral angle are equal, and all solid angles are equivalent. The tetrahedron is folded by 4-sided pyramid, the octahedron has the double-pyramid with 8 faces and 20-sided pyramid for the icosahedron. The cube is constructed by 6 square faces while the dodecahedron is composed of 12-sided of regular pentagons.

Geometrical parameters: The number of faces, edges, and vertices of each type of platonic solids is described in the Table 1. In mathematics, the Euler's formula shows the relationship between total number of vertices (V), edges (E), and faces (F) by the Eq. 1

$$V - E + F = 2 \quad (1)$$

The icosahedron has the largest number of faces with 20 while the tetrahedron has only 4 faces. If each face of platonic solids has i sides and k edges of the polyhedron meet at each vertex. The conditions are i, k greater than 3 ($i, k \geq 3$) because every face have at least three edges and at least three edges of faces meet at each vertex. The faces are equilateral triangle if $i = 3$ and changing the values of k will yield the other three types of polyhedron, including the tetrahedron ($k = 3$), the octahedron ($k = 4$), and the icosahedron ($k = 5$). When $i = 4$, the faces are square and the cube has $k = 3$ edges which meet at each vertex. The last case with $i = 5$ of regular pentagons and only $k = 3$ will generate the dodecahedron.

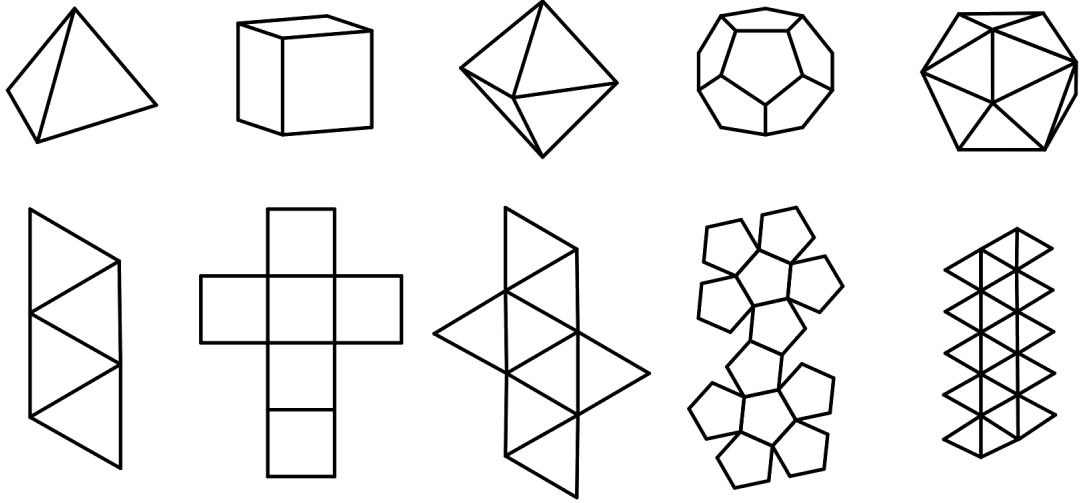


Figure 1: The platonic solids. From left to right with models and unfolding models: the tetrahedron, the cube, the octahedron, the dodecahedron, and the icosahedron

Table 1: Properties of polyhedron

	Faces	Edges	Vertices	Edges on each face	Edges meeting at each vertex
Tetrahedron	4	6	4	3	3
Cube	6	12	8	4	3
Octahedron	8	12	6	3	4
Dodecahedron	12	30	20	5	3
Icosahedron	20	30	12	3	5

The path planning for platonic solids focuses on the rolling of the models through edge-contact. Each edge is shared by two faces and each face may have e edges. Assume that Δ is the quantity faces which contact at an edge or $E = \frac{1}{2}\Delta$. However, each vertex will be shared by f faces. Then, $V = \frac{\Delta}{f}$. After substituting these two quantities into the Eq. 1, the result is:

$$\begin{aligned}
 V - E + F &= 2 \\
 \frac{\Delta}{f} - \frac{1}{2}\Delta + F &= 2 \\
 \rightarrow F &= \frac{4f}{2e - fe + 2f}
 \end{aligned} \tag{2}$$

From the Eq. 2, the condition is that $2e - fe + 2f$ should be positive because F and $4f$ are both positive. As above requirements with $(e \geq 3)$ and $(f \geq 3)$, there are only five pairs of integers (e, f) satisfy this condition. So, the results of these pairs are shown in the last two columns in Figure 1.

In order to execute the transformation stage, a rotation angle needs to determine in the path planning algorithm. In the context of regular convex polyhedra, a rotation angle is supplementary with a dihedral angle which is the angle between two connected faces along an edge inside the polyhedra. The Table 2 shows the radii of each solid with the inradius (r_i), the midradius (ρ), the circumradius (R) and the dihedral angles (β).

Table 2: Geometrical parameters of platonic solids

	r_d	ρ	R	dihedral angles (β)
Tetrahedron	$\frac{1}{12}\sqrt{6}$	$\frac{1}{4}\sqrt{2}$	$\frac{1}{4}\sqrt{6}$	$\cos^{-1}(\frac{1}{3})$
Cube	$\frac{1}{2}$	$\frac{1}{2}\sqrt{2}$	$\frac{1}{2}\sqrt{3}$	$\frac{1}{2}\pi$
Octahedron	$\frac{1}{6}\sqrt{6}$	$\frac{1}{2}$	$\frac{1}{2}\sqrt{2}$	$\cos^{-1}(-\frac{1}{3})$
Dodecahedron	$\frac{1}{20}\sqrt{250 + 110\sqrt{5}}$	$\frac{1}{4}(3 + \sqrt{5})$	$\frac{1}{4}(\sqrt{15} + \sqrt{3})$	$\cos^{-1}(-\frac{1}{5}\sqrt{5})$
Icosahedron	$\frac{1}{12}(3\sqrt{3} + \sqrt{15})$	$\frac{1}{4}(1 + \sqrt{5})$	$\frac{1}{4}\sqrt{10 + 2\sqrt{5}}$	$\cos^{-1}(-\frac{1}{3}\sqrt{5})$

4 ALGORITHM\METHODOLOGY

4.1 Path Planning Based Rolling Contact

Rolling on discretized surfaces: The surface contacts between platonic solids and the plane can be categorized into three types as shown in Figure 2 including square polygon for a cube, triangle polygon for the tetrahedron, the octahedron, and the dodecahedron, pentagon polygon for the dodecahedron. The bottom surface of a cube occupies each square on the grid when the path planning process is executed. This property is applied for a triangular grid with different rotation angles. In physics, the rotation angle of the cube is $\pi/2(\text{rad})$ while the rotation angles of tetrahedron, octahedron, icosahedron and dodecahedron are $\pi - \arctan(2\sqrt{2})$, $\pi - 2\arctan\sqrt{2}$, $\arccos(-\sqrt{5}/3)$, and $\pi - \arccos(-\sqrt{5}/5)$ in radian respectively.

The square grid has $\pi/2$ at all corners while the triangular grid has $\pi/3$ between two arbitrary edges at a vertex. In the case of dodecahedron rolling contact, Figure 2d shows the two types of connections between pentagons where the first case has a gap (Figure 2c) and the other has overlap pentagon connection. A regular pentagon has five interior angles of 108° which generate a gap between three pentagons surrounding because of $3 * 108^\circ = 324^\circ$, which is different 360° of the full circle. Another case of four overlap pentagons with $4 * 108^\circ = 432^\circ$ is greater than the circle of 360° . The path planning through the rolling of the dodecahedron solid can be categorized into two of these cases. However, the environment can be complex due to various types of the grid in the latter case. It would be found possible paths in the first case of dodecahedron without overlap rolling while the other cases with overlap rolling cannot guarantee the paths.

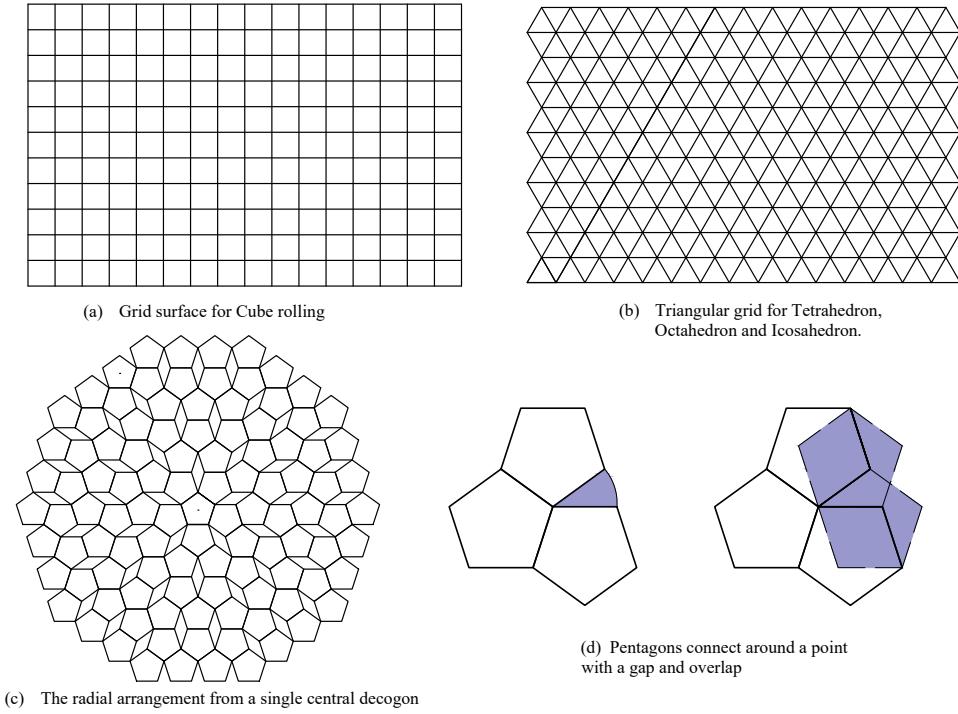


Figure 2: Grid of platonic solids

Rodrigues' roatation: It is assumed that the motion of rolling the platonic solids is a pure rolling without slipping or spinning at a line contact. Rolling the platonic solids means rolling all its vertices in 3D environments indicated by changing the coordinates. There were several algorithms to transform vertices stored in a matrix in 3D space. In this case, the Rodrigues's rotation method in [18] is used to represent the rotation matrix in the path planning algorithm.

Algorithms: Due to the different surface contacts, there are three types of direction for the rolling of platonic solids. As shown in Figure 3, the cube has four directions with the square surface contact while the tetrahedron, the octahedron and the icosahedron have three rolling directions with the triangular surface contact. The dodecahedron with pentagon surface contact has five rolling directions. In the case of a rolling cube, the surface contact is surrounded by four edges which means there are four possible directions through the edges. In this work, the proposed path planning algorithm deals with rolling from initial configuration within the position and orientation to the goal within the same position but different orientation. While rolling on the smooth plane, the platonic solid models will contact to the plane through their edges.

Algorithm 1 shows that path planning for cube rolling based on tree graph search has some important steps. The first step is to initialize the coordinates and the orientations of the initial cube and the target cube which is stored as the initial path. The same as the tree expansion, cube will roll in four different directions including the right, left, up and down is the next step. From these new positions and orientations, the cubes will continue to expand with only three directions to avoid return the previous positions. An example of this step is that from the initial coordinate the cube achieves a new position after doing rolling for the right direction, the new three positions of the cube by rolling through right, up, and down direction. After implementing the expansion steps through rolling, the function of checking whether updated models reach the goal is called through the loop. By that means, the loop will stop when reaching the goal, whereas the loop will continue to execute and store new models to the initial path. While the searching algorithm is executing, the data structure is used to store the positions and orientations from the start to the current. This process runs in time $O(|E|^3)$ (where $|E|$ is the number of updated cubes) which causes the longer the running time of the searching technique.

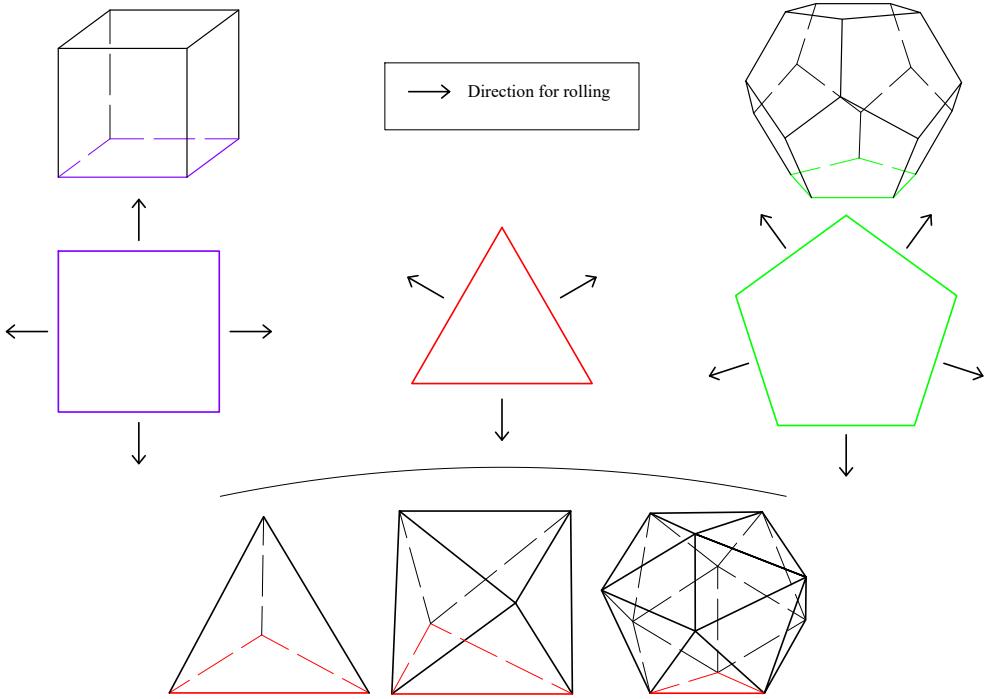


Figure 3: Rolling directions for each type of platonic solids

4.2 Tree Exploration Algorithm

The node tree exploration for the searching algorithm described in Algorithm 1 is similar to the non-recursive depth-first-search algorithm [19]. The graph search in Figure 4 shows the expansion from the *root* with node *S* to multi-level from 1^{st} level ... n^{th} level for the case study of a cube solid. Each node indicates the position of the cube's center and the orientation of the cube. The node *S* means Start-Point while *R*, *L*, *U*, *D* are labelled for four different directions including right, left, up and down respectively. For each iteration, a tree with a node including *3D* coordinate and orientation is stored at each level. At the same time, the algorithm of checking the goal configuration will be called to check whether the current executing level achieves the target.

In other cases of tetrahedron, octahedron, and icosahedron with the triangular grid (Figure 2b), there are three directions at the first rolling and only two directions for the rest of the path-finding process. Only the case of dodecahedron has a different approach from the algorithm. The path planning algorithm depends on the environment including gaps or overlaps between two pentagon connections as can be seen in Figure 2b.

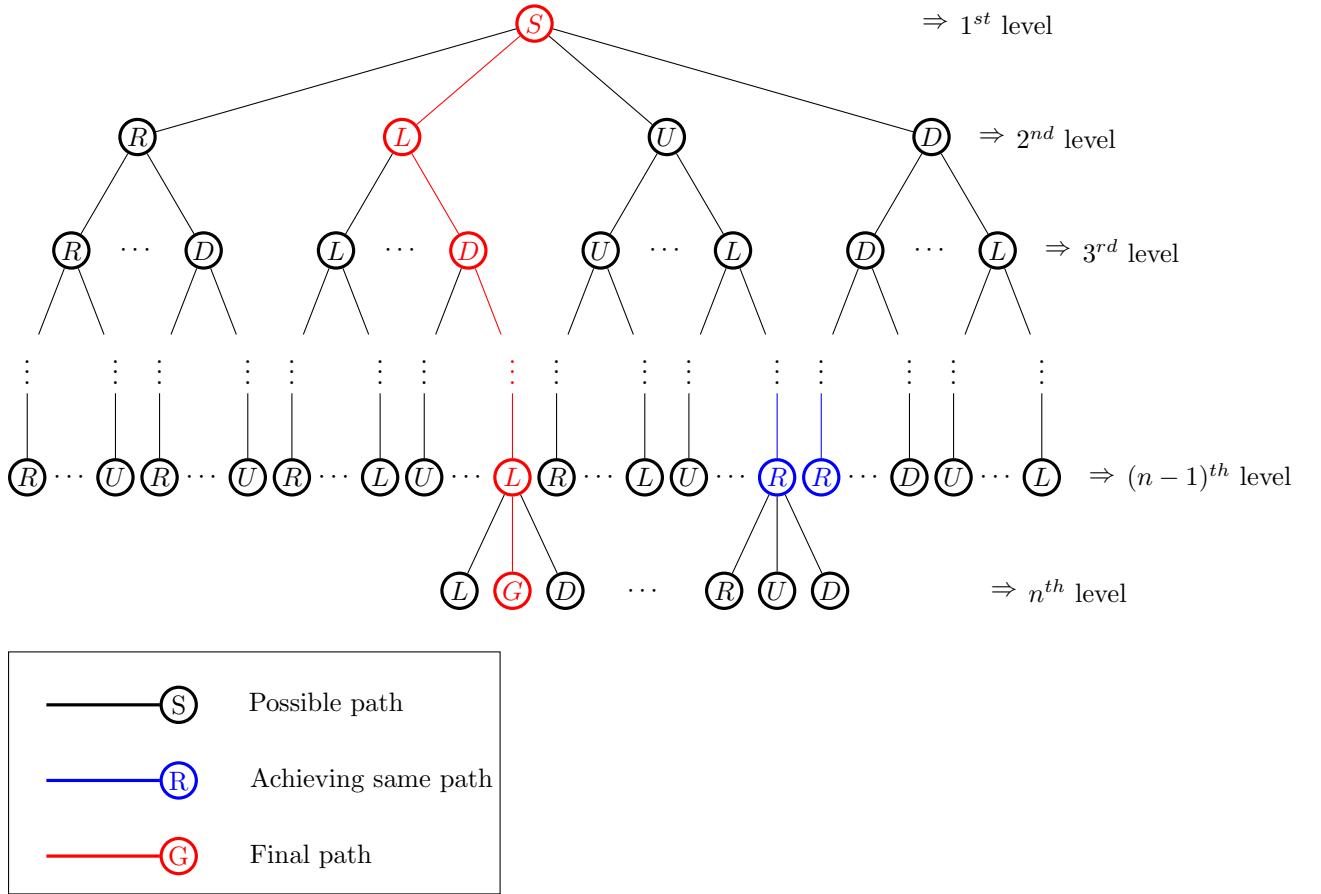


Figure 4: Tree Exploration of Cube Rolling

Starting from the root *S*, path planning based rolling of the cube model at the first level of expansion will generate to four different directions *R*, *L*, *U*, *D*. In the next level, the cube can only roll with three directions without rolling back to the previous position. An example of the second level is that node *R* will roll to right, up, and down directions indicated by node *R*, *U*, *D* respectively.

To eliminate the processing time in the proposed algorithms, whenever any updated points achieved the same position and orientation, these nodes will merge at that *level*. An example from Figure 4 shows that two updated *R* nodes (blue nodes at $(n-1)^{th}$ level) have achieved the same position. The next path is generated from this merged nodes. From the tree exploration algorithm, the result can show only one path or various paths which depends on the initial and goal configuration. The first path is the shortest path because the execution time is shortest based on the condition of achieving goal configuration.

Algorithm 1 Path planning based rolling contact for Cube model.

```

1: procedure CUBE PATH PLANNING( $S_p, G_p$ )  $\triangleright$  Find the shortest path from start to goal position with
   different orientation
2:    $flag \leftarrow false$ 
3:    $Path[S_p] \leftarrow S_p$ 
4:    $newPoints \leftarrow \text{ROLLING4DIRECTIONS}(S_p)$   $\triangleright$  Generate first four updated points
5:   while  $newPoints \neq G_p$  do
6:      $updatedPoints \leftarrow \text{TREEEXPLORATION}(newPoints)$   $\triangleright$  Update new three right rolling models
7:      $n \leftarrow \text{size}(updatedPoints)$ 
8:     for  $i \leftarrow 0, n$  do
9:       for  $j \leftarrow 1, n$  do
10:        if  $updatedPoints[i] = updatedPoints[j]$  then
11:           $\text{remove}(updatedPoints[i])$ 
12:        end if
13:      end for
14:    end for
15:     $flag \leftarrow \text{CHECKINGTARGETPOINT}(updatedPoints)$   $\triangleright$  Compare updated points with goal point
16:    if  $flag = true$  then
17:      return  $Path[S_p, G_p]$   $\triangleright$  Store new point to  $Path$ 
18:    end if
19:     $newPoints = updatedPoints$ 
20:  end while
21:  return "no path found"
22: end procedure
23: procedure ROLLING4DIRECTIONS( $S_p$ )  $\triangleright$  Generate new points in different direction of rolling
24:    $(newRightPoint, newLeftPoint, newUpPoint, newDownPoint) \leftarrow \text{ROLLINGCONTACT}(S_p)$ 
25:   return  $newPoints \leftarrow (newRightPoint, newLeftPoint, newUpPoint, newDownPoint)$ 
26: end procedure
27: procedure TREEEXPLORATION( $newPoints$ )
28:   if  $dir = right$  then
29:      $updatedPoints \leftarrow (newRightPoint, newUpPoint, newDownPoint)$ 
30:   else if  $dir = left$  then
31:      $updatedPoints \leftarrow (newLeftPoint, newUpPoint, newDownPoint)$ 
32:   else if  $dir = up$  then
33:      $updatedPoints \leftarrow (newRightPoint, newLeftPoint, newUpPoint)$ 
34:   else
35:      $updatedPoints \leftarrow (newRightPoint, newLeftPoint, newDownPoint)$ 
36:   end if
37:   return  $updatedPoints$ 
38: end procedure
39: procedure CHECKINGTARGETPOINTS( $updatedPoints, G_p$ )
40:   if  $updatedPoints = G_p$  then  $\triangleright$  Consider both position and orientation
41:      $flag \leftarrow true$ 
42:   end if
43:   return  $flag$ 
44: end procedure

```

5 EVALUATION

The proposed algorithm for platonic solids path planning by rolling through edge contact was implemented in MATLAB environment. In general of path planning, there are three case studies including same location and different orientation between initial configuration and goal configuration, long distance between two configurations, and bi-direction path-finding. To validate the proposed algorithm, this study only considers the first case study of path planning that both initial and goal configuration have the same positions and different orientations.

Assume that the platonic solids' edges have the same length with a ($a = 1$) and one of the faces of the platonic solids contact to OXY . The environment for each type of platonic solids is discretized from the smooth surface. For example, cube solid will roll on the square grid while tetrahedron, octahedron, and icosahedron solids roll on the triangle gird. Only dodecahedron solid rolls on the pentagon grid with the two specific cases including the gaps and overlaps between pentagons. The path planning based on rolling for the five cases of platonic solids is represented as the following.

5.1 Cube solid

Properties: The cube has a length a which is the same as the length of the side of each grid square. Path planning for rolling a cube is not as complex as other cases of platonic solids. The rolling angle β has the value of $\frac{\pi}{2}$ as shown in Figure 5 ($\beta = \angle I K I_1 = \frac{\pi}{2}$). The center of cube will move along the curve $I I_2 I_1$ following the Ox axis. The proposed algorithms will focus on rolling along Ox or Oy axis. However, all the vertices are stored in a matrix will change their coordinates in 3D space. An example from the Figure 5, eight vertices of the cube $ABCDMNOP$ and the center I are stored in a matrix $M(9, 3)$ with 9 rows for nine 3D points, and 3 columns for 3D coordinate Ox, Oy, Oz of each point. After finishing the rotation angle β based on the rotation axis MN , the coordinates of the cube's vertices are updated within a new cube $MA'D'PN'B'C'O'$. This is the basic step of rolling a cube through a line contact.

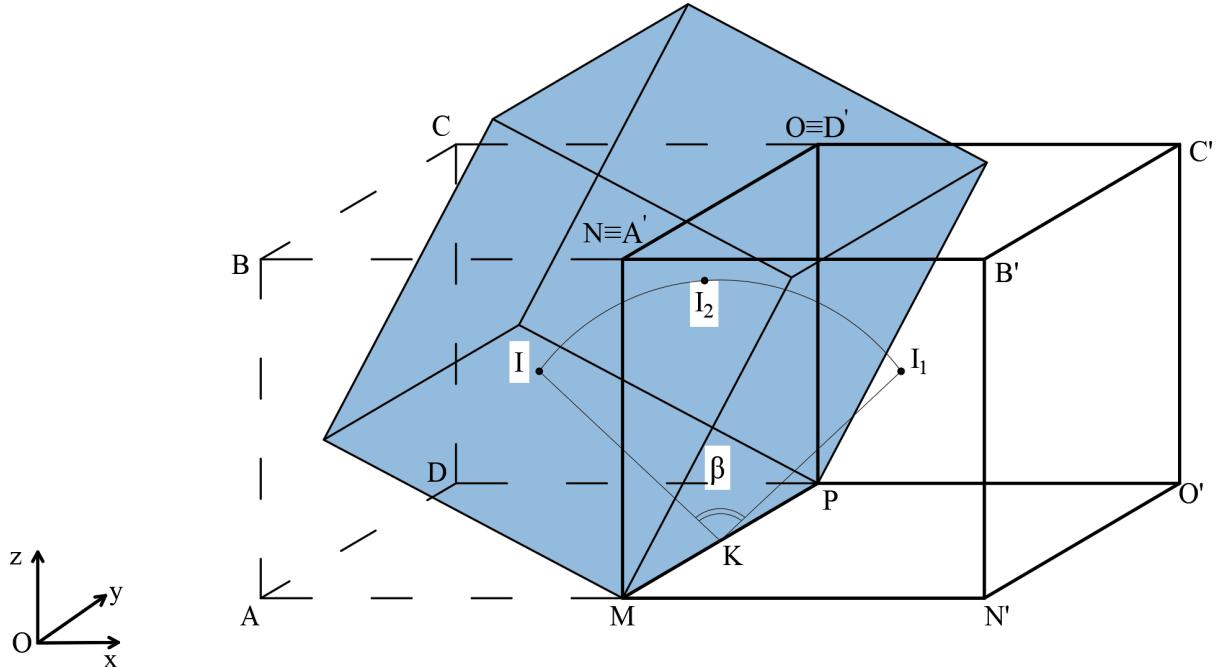


Figure 5: Rotation angle of a cube on a plane

Path planning: The only way to move from the initial position to the goal position of the cube solid is by rolling from square to square on the grid without moving diagonal. Figure 6 shows the first four layers of the cube path-finding on the grid. The algorithm implements in $O(|E|^3)$ running time from the second layer called the expansion three branches from a tree. All 3D coordinates of the cube are stored in a matrix which can affect the computer's storage capacity. When the updated cubes achieved one of the same previous configurations, the execution time can be reduced by releasing this configuration or stopping the expansion of this tree branch. The star position (*) in the grid (Figure 6) is occupied by

the two updated cubes with different orientations in *Layer 2*.

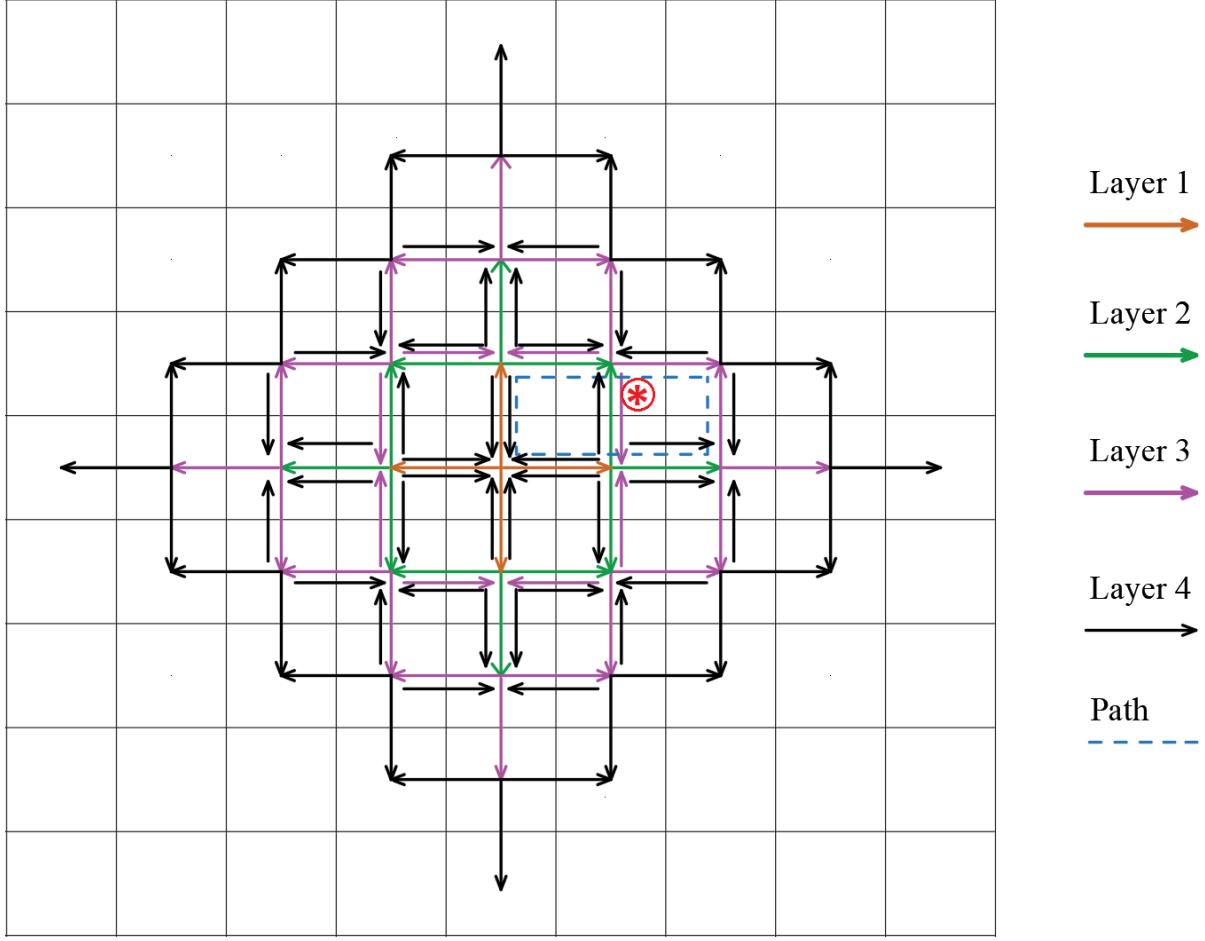


Figure 6: The first four layers of cube rolling

Rotation matrix: In this paper, the Rodrigues' rotation matrix [18] is used in each level of the proposed path planning algorithm to rotate the polyhedrons from a position to other positions. In the 3D coordinate system, the unit vector is given by $\hat{\omega} = (\omega_x, \omega_y, \omega_z) \in \mathbf{R}^3$. The Euler-Rodrigues' rotation formula can be written in a standard form and then expressed as the $[3 \times 3]$ rotation matrix from the axis-angle representation of rotations as below.

$$\begin{aligned}
 R_{\hat{\omega}(\theta)} &= e^{\omega\theta} \\
 &= I + \omega \sin \theta + \omega^2 (1 - \cos \theta) \\
 &= \begin{bmatrix} \cos \theta + \omega_x^2 (1 - \cos \theta) & \omega_x \omega_y (1 - \cos \theta) - \omega_z \sin \theta & \omega_y \sin \theta + \omega_x \omega_z (1 - \cos \theta) \\ \omega_z \sin \theta + \omega_x \omega_y (1 - \cos \theta) & \cos \theta + \omega_y^2 (1 - \cos \theta) & -\omega_x \sin \theta + \omega_y \omega_z (1 - \cos \theta) \\ -\omega_y \sin \theta + \omega_x \omega_z (1 - \cos \theta) & \omega_x \sin \theta + \omega_y \omega_z (1 - \cos \theta) & \cos \theta + \omega_z^2 (1 - \cos \theta) \end{bmatrix} \quad (3)
 \end{aligned}$$

where \mathbf{I} is the 3×3 identity matrix, and the ω denotes the antisymmetric matrix

$$\omega = [\omega \times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

The matrix ω gives Lie algebra $so(3)$ in the form of a skew-symmetric matrix with the rotation matrix $\mathbf{R} \in SO(3)$, which corresponds to a rotation angle θ . All the vertices' coordinates of the polygons will be stored in a matrix A and a new matrix A' is the product of A and \mathbf{R} (Eq. 4). One of the edges of the polyhedrons which contacts the surface is considered as a rotation matrix. For each iteration of the closed-path planning algorithm, it is essential to determine a direction and an edge of the polyhedrons

before performing the rotation matrix to transform the polyhedron into a new configuration.

$$A' = AR_{\hat{\omega}(\theta)} \quad (4)$$

where $A(m, n)$ is the matrix of m rows for vertices and n columns for 3D coordinates each vertex. The updated matrix is $A'(m', n')$.

To be more visualized, Figure 7(a) shows the initial configuration and the first step of rolling the cube with its coordinates in the red, green and blue arrows. The expansion of the cube after running the algorithm six iterations are shown in Figure 7(b).

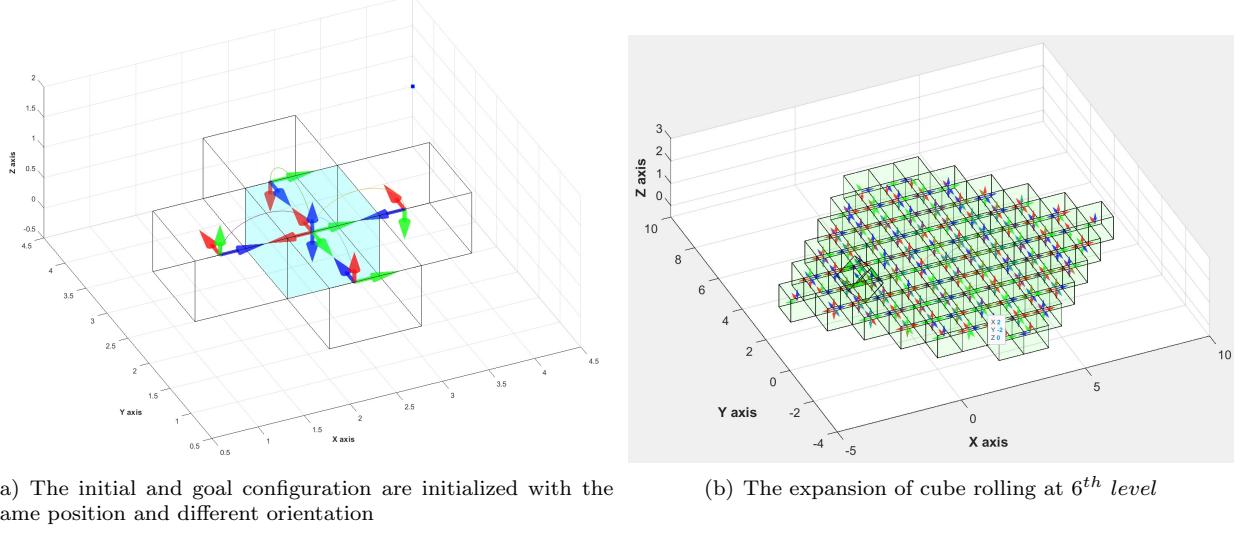


Figure 7: Initial configuration and the 6th level of the cube's expansion

Figure 8 shows that the two closed-paths are found at the same time after executing the proposed path planning algorithm six iterations. The initial configuration includes the position at [2.5, 2.5, 0.5] and the orientation with three arrows (green, red, blue) corresponding to $(Ox, Oy, -Oz)$ while the goal configuration has the same position with the initial position and the orientation with (green, red, blue) corresponding to $(Ox, -Oy, Oz)$. If the initial or the goal configuration change their orientation, the algorithm will implement at different times and generates different paths.

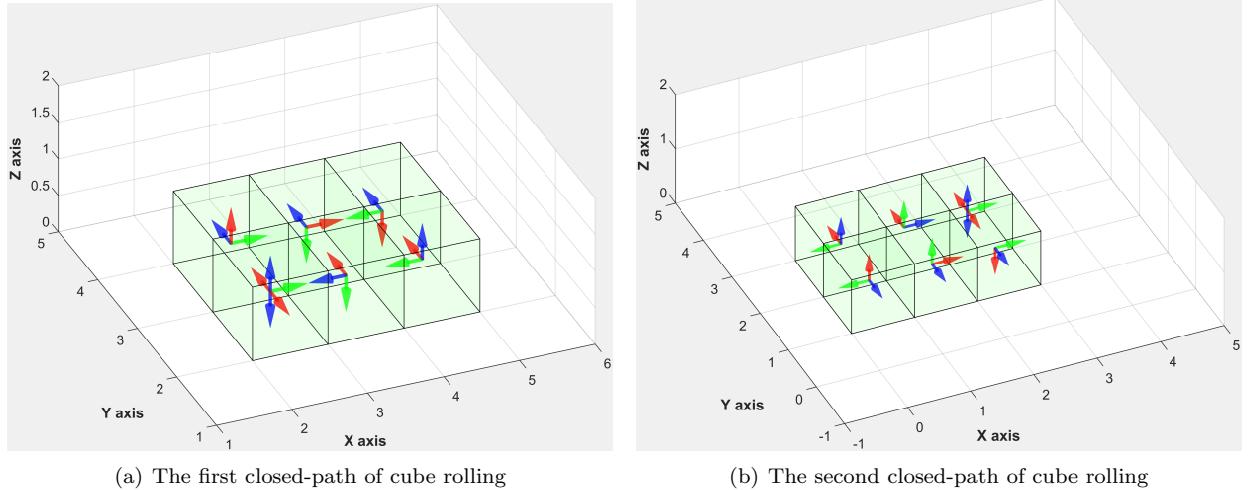


Figure 8: The two closed-path of cube rolling with initial and goal position at [2.5, 2.5, 0.5]

5.2 Tetrahedron solid

Properties As can be seen from Figure 9, the Tetrahedron has constructed by four faces of the equilateral triangles. Then the height of triangle ABC is AM and $AM = DM = a\sqrt{3}/2$. Because of $r = OH = a\sqrt{6}/12$ (the radius of insphere) and $R = OA = a\sqrt{6}/4$ (the radius of circumsphere), the height of tetrahedron is $AH = OA + OH = a\sqrt{6}/3$. The rotation angle is β determined by supplementary angles $\alpha = \arctan(2\sqrt{2})$ or $\beta = \pi - \alpha = \pi - \arctan(2\sqrt{2})$.



Figure 9: Tetrahedron geometrical properties

Path Planning: The case study of the tetrahedron in this study is the same as the rolling cube path-finding which only considering the path planning through rolling from initial configuration to origin coordinate with different orientations. Figure 10 illustrates two of four cases of the tetrahedron path-finding within rolling (red and cyan arrows are pointing down to plane respectively). A tetrahedron has symmetry properties with indistinguishable for any two faces, edges and vertices. To be more specific, dihedral triangles have the same three angles within 60° . In one cycle of rolling a tetrahedron around any vertices, the tetrahedron always achieves the initial configuration due to six times of rolling ($6*60^\circ = 360^\circ$, a full circle).



Figure 10: The two of four cases

5.3 Octahedron solid

Properties: An octahedron has six vertices and twelve edges which generates eight equilateral triangles. Figure 11 shows that an octahedron has length a with the based surface ABC . In each step of a rolling, the octahedron will roll into three direction through edge contacts with same rotation angle which is determined as below.

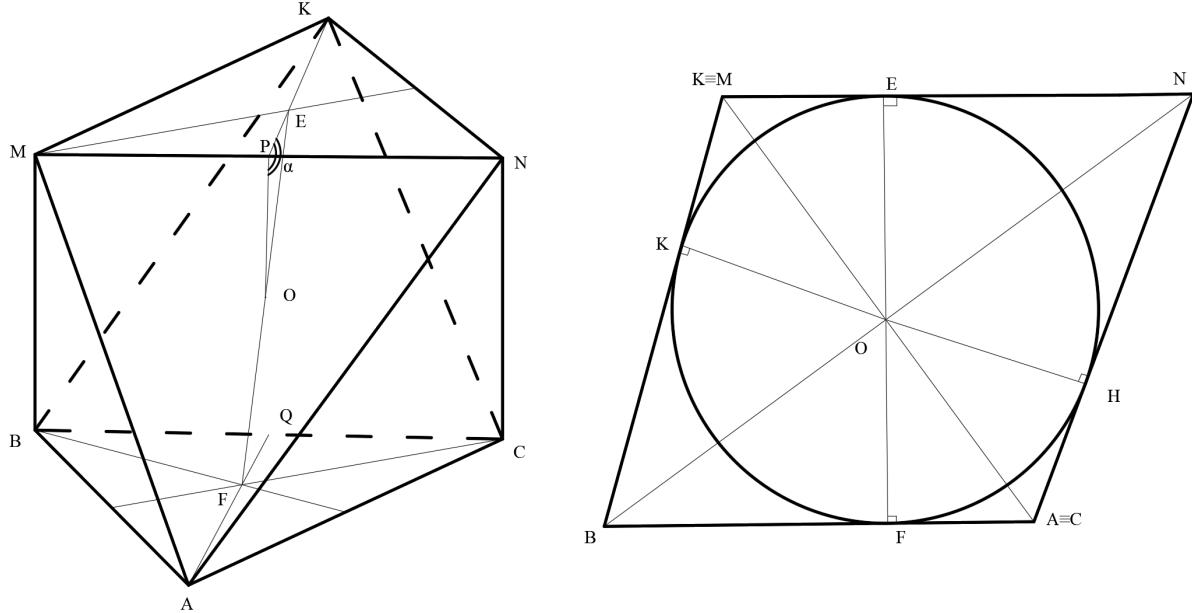


Figure 11: Octahedron geometrical properties

From Figure 11 we have:

$$\begin{aligned} AK &= 2OK = 2\sqrt{(MK^2 - MO^2)} = a\sqrt{2} \\ \Rightarrow OK &= OM = ON = a\frac{\sqrt{2}}{2} \end{aligned}$$

To find the rotation angle, the supplementary angle should be calculated first.

$$\alpha = \angle OPK = \arctan \frac{OK}{OP} = \arctan \sqrt{2}$$

Then, the rotation angle has the result as $\beta = \pi - 2\alpha = \pi - 2 \arctan \sqrt{2}$,

Due to $\angle POK = 90^\circ$, a distance from the center O of the octahedron to the center of bottom triangle is calculated.

$$\begin{aligned} \frac{1}{OE^2} &= \frac{1}{OP^2} + \frac{1}{OK^2} \\ &= \frac{1}{(\frac{\sqrt{2}}{2})^2} + \frac{1}{(\frac{1}{2})^2} \\ \Rightarrow OE &= \frac{\sqrt{6}}{6} \end{aligned}$$

Applying the theory of the equilateral triangle

$$\begin{aligned} EO &= OF = a\frac{\sqrt{6}}{6} \\ KE &= FA = a\frac{\sqrt{3}}{3} \\ EP &= FQ = a\frac{\sqrt{3}}{6} \end{aligned}$$

Path planning: Based on the classical path planning which is the movement from point to other points, the octahedron path planning within rolling has three directions to move. In the Figure 11, the bottom layer $\triangle ABC$ which contacts to OXY can roll with the directions FQ, FO_1, FR . The distance between two shortest positions is $2FR = 2a\frac{\sqrt{3}}{6} = a\frac{\sqrt{3}}{3}$.

As can be seen from Figure 12(a), the shortest path within red line includes ten line segments for rolling the octahedron. The total length of the path equals $10a\frac{\sqrt{3}}{3}$ (edge length of the octahedron is a).

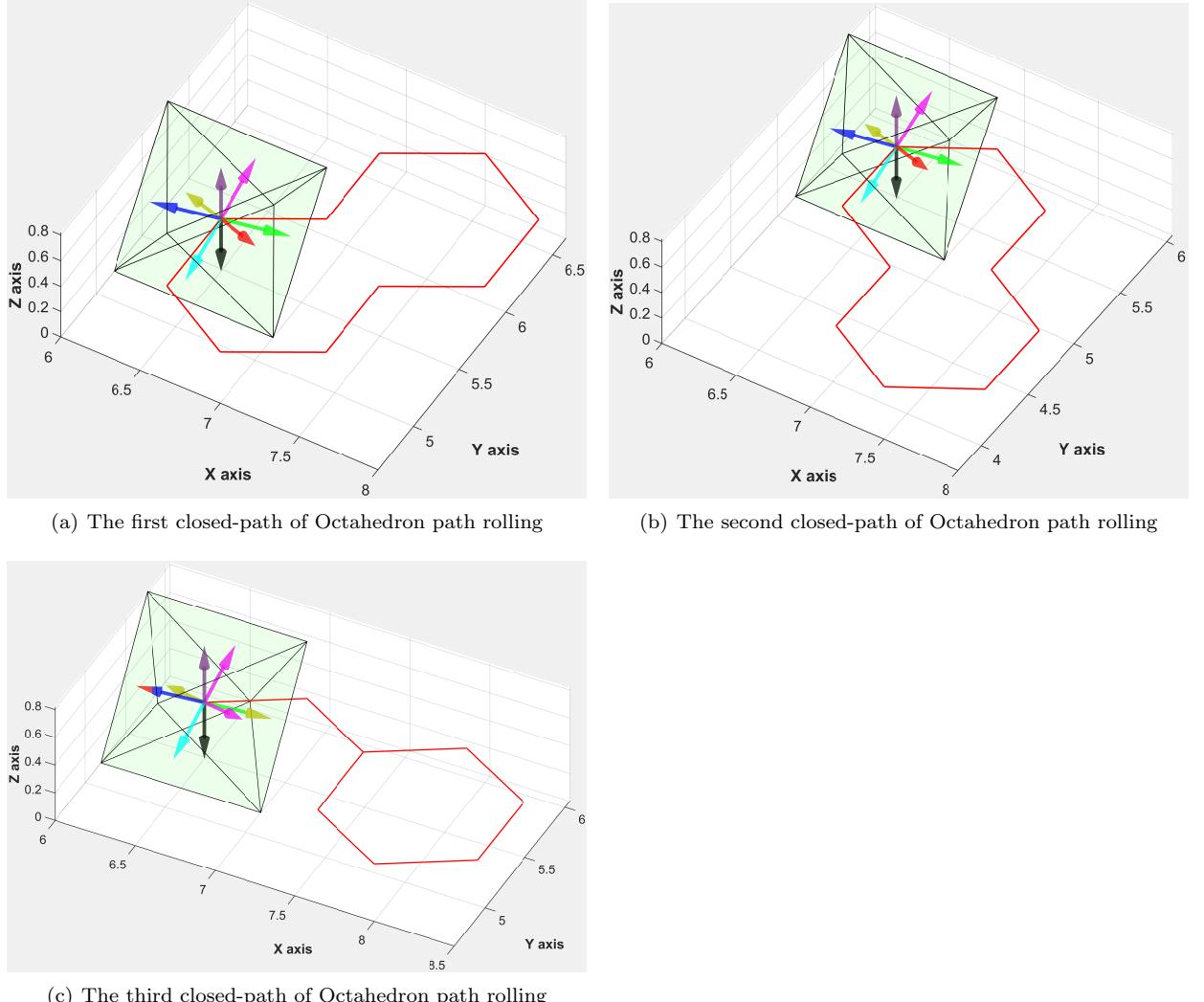


Figure 12: Three shortest paths of octahedron based rolling

5.4 Icosahedron solid

Properties: The convex regular icosahedron in Figure 13 is one of the five regular Platonic solids has 12 vertices, 20 triangular faces, and 30 edges. Assume that the icosahedron has the edge length with a . The crossing surface of the solid at vertex A perpendicular with OD will generate a pentagon $ABCEF$.



Figure 13: Icosahedron geometrical properties

At the right side of the Figure 13, the pentagon $ABCEF$ has circumcircle radius R , inscribed circle r_i , and the height d_{CS} ($R + r_i$). The golden ratio Φ (irrational number) has the value of $\frac{1+\sqrt{5}}{2}$ can be found by:

$$\Phi = \frac{OX}{XB} = \frac{OX}{\frac{1}{2}a}$$

Same as the case of cube's path planning algorithm, the initial step of rolling icosahedron is to find the rotation angle. The rotation angle is a supplementary of the $\angle QRS$. The line OR is perpendicular to QS and $OS = OQ = OM = r_i = \frac{\Phi^2}{2\sqrt{3}}a$. BI is a diagonal of pentagon $BCLIJ$ and $\angle QRS = 2\angle ORS$.

Otherwise, $OR = OX = \frac{1}{2}a\Phi$, and

$$\begin{aligned} \angle QRS &= 2\angle ORS \\ \sin \angle ORS &= \frac{OS}{OR} = \frac{\frac{\Phi^2}{2\sqrt{3}}a}{\frac{1}{2}a\Phi} = \frac{\Phi}{\sqrt{3}} \\ \Rightarrow \angle QRS &= 2 \arcsin \frac{\Phi}{\sqrt{3}} \end{aligned}$$

To be more detail in Figure 14, the rotation angle of icosahedron can be determined from the result of $\angle QRS$, as the following.

$$\begin{aligned} \beta &= \pi - \angle QRS \\ &= \pi - 2 \arcsin \frac{\Phi}{\sqrt{3}} \\ &= \pi - \arccos -\frac{\sqrt{5}}{3} \end{aligned}$$

To calculate the rotation axis for all the cases of the triangular surface contact, there are three axis such as IJ , JK , and IK with the 3D coordinates as following.

$$JK = [a \ 0 \ 0]$$

$$IJ = [-\frac{1}{2}a \ (\frac{\sqrt{3}}{6} + \frac{\sqrt{3}}{3})a \ 0]$$

$$IK = [-\frac{1}{2}a \ -(\frac{\sqrt{3}}{6} + \frac{\sqrt{3}}{3})a \ 0]$$

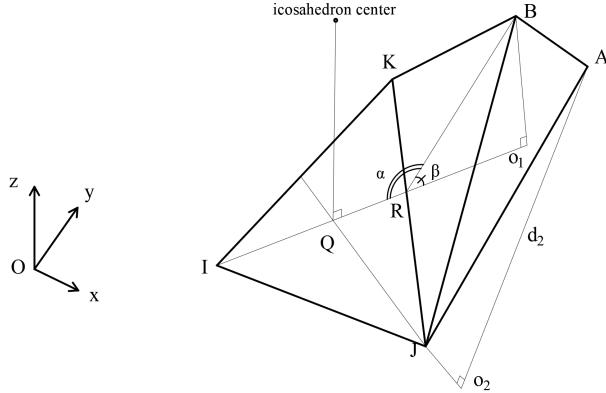


Figure 14: Rotation angle and rotation axis

Path planning Path planning of the regular icosahedron through rolling in known environment has initial coordinate at $[6.5, 5.5, 0]$ and the surface contact with red arrow points to bottom. The goal configuration with same as the initial coordinate has the contact surface with black arrow as shown in Figure 15(a). This figure shows the shortest path with the red line including 14 line segments. It means that the icosahedron rolled 14 times from the initial configuration to the goal configuration. The path can be seen more precisely from the top view in Figure 15(b). The total length of this icosahedron path equals $14a\sqrt{3}$ (edge length of the octahedron is a).

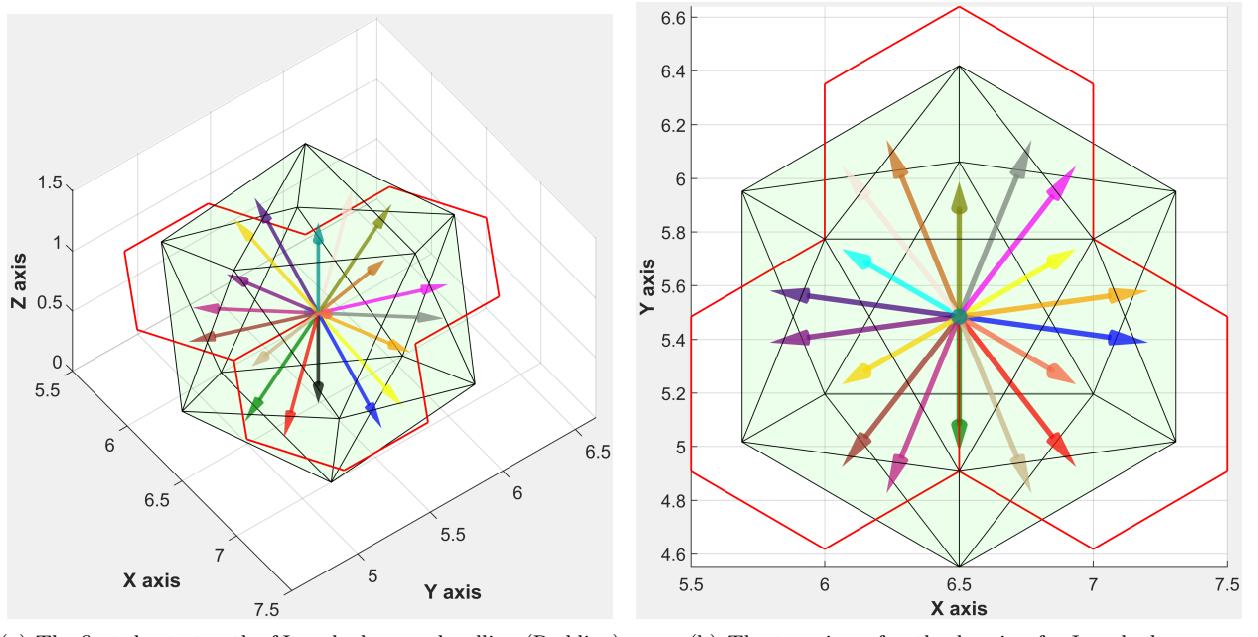


Figure 15: Path of icosahedron rolling though edges

5.5 Dodecahedron solid

Properties: An dodecahedron is constructed from twelve self-intersecting faces with the same pentagon shape. These faces meet at each vertex and there are total of twenty vertices in a dodecahedron as shown in Figure 16. From Eq. 1, the great dodecahedron satisfy the Euler's formula ($V = 20, F = 12, E = 30$). It will be assumed that the coordinates $Oxyz$ lies on $ABCDE$ surface within Oy through A and Oz perpendicular to $ABCDE$. The 30 edges have the same length as a . It should be determined all the vertices' coordinates in the three dimensional system. Figure 16 indicates the lengths of each vertex from l_1 to l_4 and the angles α_1 to α_4 which correspond to the five sides of a pentagon.

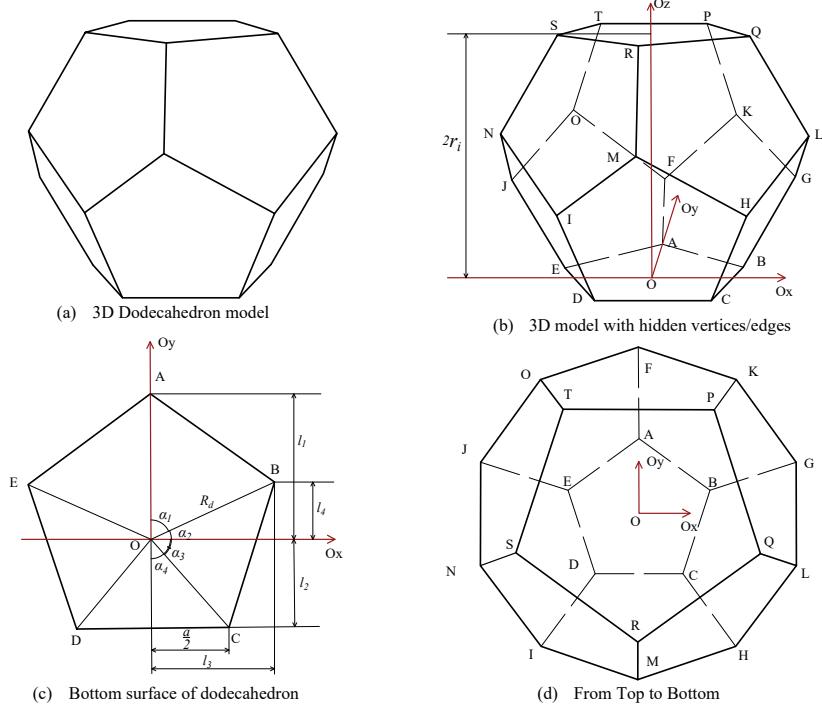


Figure 16: Dodecahedron's vertices.

The path planning will implement on a surface but it will be considered in 3D spaces. Then, each vertex will be determined on 3D coordinates such as the vertex A has coordinate with $[A_x A_y A_z]$. Based on the properties of pentagon, the angle $\alpha_1 = \frac{2\pi}{5}$ and $\alpha_4 = \frac{\pi}{5}$, because the angle between Ox and Oy is $\frac{\pi}{2}$, the sum of α_1 and α_2 is $\alpha_1 + \alpha_2 = \frac{\pi}{2}$. Then the other two angles α_2 and α_3 can be founded as below.

$$\begin{aligned}\alpha_2 &= \frac{\pi}{2} - \alpha_1 = \frac{\pi}{2} - \frac{2\pi}{5} = \frac{\pi}{10} \\ \alpha_3 &= \alpha_1 - \alpha_2 = \frac{2\pi}{5} - \frac{\pi}{10} = \frac{3\pi}{10}\end{aligned}$$

From Figure 16(c), these labelled dimensions can be calculated as $l_1 = R_d = \frac{a}{2\sin\alpha_4}$ with R_d is the circumradius of dodecahedron, $l_2 = l_1 \cos \alpha_4$, $l_3 = l_1 \cos \alpha_2$, and $l_4 = l_1 \sin \alpha_2$. The golden ratio Φ with the value of $\frac{1+\sqrt{5}}{2}$ which is the length of the diagonal of a square with 1 unit length of sides is used to calculate the circumscribed sphere radius of dodecahedron. It can be assumed that the dodecahedron has the length a , the radius of an inscribed sphere (r_i) and the circumscribed sphere radius (r) are shown as below.

$$\begin{aligned}r_i &= \frac{a}{20} \sqrt{10(25 + 11\sqrt{5})} \\ r &= a \frac{\sqrt{3}(1 + \sqrt{5})}{2}\end{aligned}$$

In the path-finding through rolling, the proposed algorithm focuses on the transformation and translation of 20 vertices of a dodecahedron. Although rolling dodecahedron's faces on 2D surface, the bottom surface which integrates to the Oxy will contact to the surface in the three dimensional space. This condition expresses the Oz dimension of all the (A, B, C, D, E) vertices which equal to 0 or $A_z = B_z = C_z = D_z = E_z = 0$. Then,

$$\begin{aligned} P_z &= Q_z = R_z = S_z = T_z = 2.r_i \\ &= \frac{a}{10} \sqrt{10(25 + 11\sqrt{5})} \end{aligned}$$

It can be seen that the distance $|AF|$ is a and the distance $|BF|$ is $2l_3$. Using the distance properties and squaring the results give:

$$\begin{aligned} AF^2 &= a^2 = (A_x - F_x)^2 + (A_y - F_y)^2 + (A_z - F_z)^2 \\ BF^2 &= (2l_3)^2 = (B_x - F_x)^2 + (B_y - F_y)^2 + (B_z - F_z)^2 \end{aligned}$$

Figure 16(d) shows that $A_x = F_x = 0$, $A_y = l_1$, $B_y = l_4$, $B_x = l_3$, $A_z = B_z$. Define $l_5 = A_z - F_z$, the relations of these equations are:

$$\begin{aligned} a^2 &= (F_y - l_1)^2 + l_5^2 \\ (2l_3)^2 &= (F_y - l_4)^2 + l_5^2 + l_3^2 \end{aligned}$$

Solving F_y and l_5 gives:

$$\begin{aligned} F_y &= \frac{a^2 - (2l_3)^2 - (l_1^2 - l_3^2 - l_4^2)}{2(l_4 - l_1)} \\ l_5 &= \frac{1}{\sqrt{2}} \sqrt{a^2 + (2l_3)^2 - (F_y - l_1)^2 - (F_y - l_4)^2 - l_3^2} \end{aligned}$$

From these above equations, all the vertices can be determined and stored in a 3D matrix. In every single position, another matrix stores the edge's contact and the rotated-direction. Using these information with the rotation matrix $R_{\omega(\theta)}$, the path planning algorithm will generate a new matrix for the dodecahedron at new position after rolling through its edge.

Path Planning: As mentioned in Section 4, the dodecahedron can roll on the two grid types. The simple case has a gap at the connection between three regular pentagons as shown in Figure 17. In this paper, we do not apply the proposed algorithm to find the path of dodecahedron with rolling on the case of overlaps between four pentagons. This overlap grid is complicated environment which can not guarantee to generate paths with proposed algorithm. In the case of grid with gaps (Figure 17), attaching other five pentagons generates five 36° gaps from a regular pentagon. The six pentagons has a shape of large pentagon and the cycle of attaching more five larger pentagons will shape a grid of pentagons.

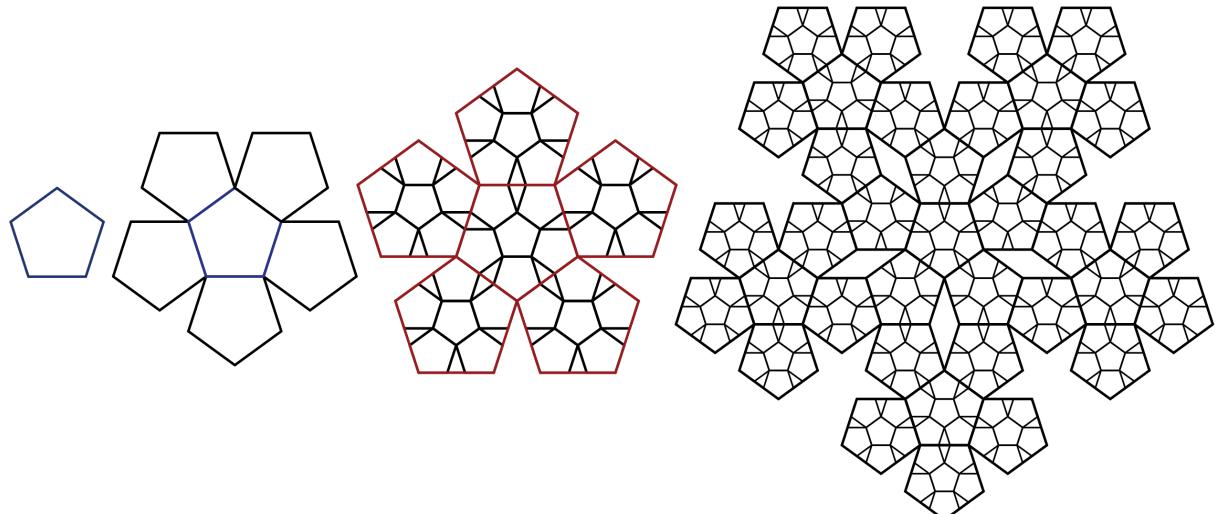


Figure 17: Initial dodecahedron grid with gaps

The rotation angle of dodecahedron on the above grid equals to $\pi - \arccos\left(\frac{-\sqrt{5}}{5}\right)$. One of the closed-paths shown in Figure 18 is initialized for both the initial and goal coordinates with [3.0 2.25 0.0]. Each step of the rolling dodecahedron generates a new dodecahedron which has a distance of $d = 2l_2$ from the previous coordinate. From Figure 16b, the distance d_d can be determined as following.

$$\begin{aligned} l_2 &= l_1 \cos \alpha_4 \\ &= \frac{a}{2 \sin \alpha_4} \cos \alpha_4 \\ &= \frac{1}{2} a \cot \alpha_4 \\ \rightarrow d_d &= a \cot \alpha_4 \end{aligned}$$

Then, the total length of the first path with the red line equals $15a \cot \alpha_4$ as shown in Figure 18.

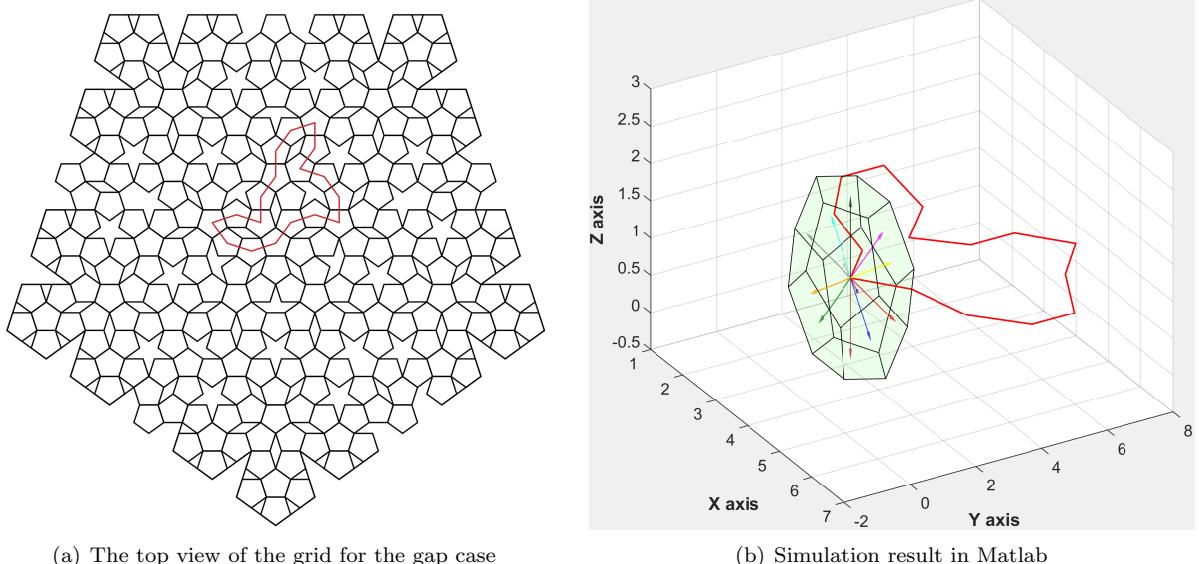


Figure 18: The rolling path of dodecahedron at an coordinate with [3.0 2.25 0.0]

According to the simulation results, implementing the proposed algorithm for other four types of platonic solids including cube, tetrahedron, octahedron, and icosahedron is more effective than for the dodecahedron case. The closed-path planning algorithm through rolling contact for the case of dodecahedron is complex due the various initial environments. Executing the algorithm is no guarantee of path-finding results with the other cases of overlap grid.

5.6 Result comparisons

Extension case: The proposed algorithm in this study can be applied for the case study of long distance between the initial and goal configurations. Figure 19 shows an example with two different paths. The closed-path planning algorithm is added to the point-to-point algorithm based on rolling in order to find the path from start point to the goal point. Assume that the initial configuration is at S_1 and the goal configuration is at S_2 . At the stage of point-to-point planning, the red line segment S_1S_2 shows the shortest distance from start position to goal position and the path can be generated along this red line. After achieving the goal position without considering the orientation, the cube updates this configuration as a new start configuration. Then, the proposed closed-path planning algorithm will implement with this updated configuration to achieve the original goal configuration.

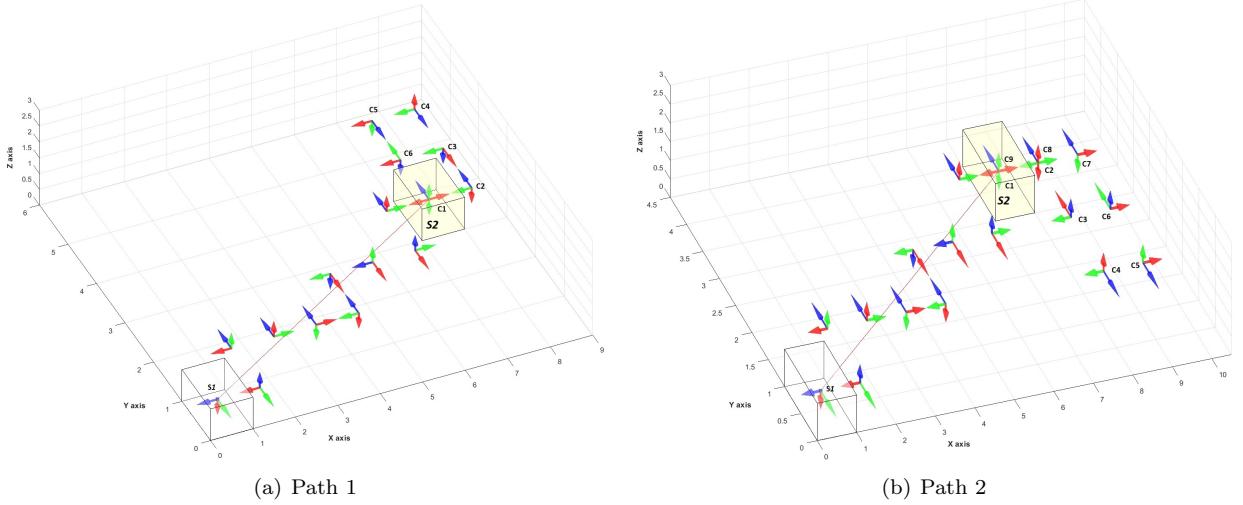


Figure 19: Caption for this figure with two images

Execution time: As can be seen in Figure 19, the point-to-point path planning in the first stage generates two same path from S_1S_2 . In the second stage, the closed path planning algorithm finds two different closed-paths. The first closed-path has 16 steps while the second closed-path has 18 steps of rolling to achieve the goal. The execution times are shown in Table 3. The shorter path has less time to implement the closed-path planning algorithm than the longer path with more steps.

Table 3: The execution times of two paths of the rolling cube shown in Figure 19

Execution Time (ms)	Path 1	Path 2
Poin-to-point	0.25	0.25
Proposed algorithm	2.02	2.34

Considering the execution times for the five types of platonic solids, Table 4 shows that the proposed algorithm runs in a short time to find the closed-paths for the rolling cube and tetrahedron. The longer execution times depend on the complex geometry of each polyhedron. In this case, the running time of the closed-path planning for the icosahedron implements extensively.

Table 4: The execution times of the first two paths are found for the platonic solids

Execution Time (ms)	Cube	Tetrahedron	Octahedron	Icosahedron	Dodecahedron
First path	4.15	5.02	48.45	135.34	89.74
Second path	4.15	5.02	62.78	179.35	N/A

6 CONCLUSIONS

In this paper, we have proposed a method of closed-path planning for the platonic solids, which successfully generates closed-paths under rolling constraint without sliding. In particular, the algorithm works by developing the tree exploration method to efficiently search possible paths for cube, tetrahedron, octahedron and icosahedron solids. In the case of the dodecahedron, the initial environments with a gap between two pentagons are varied, which may lead to different paths. The algorithm is applied to different initial environments with the overlap case is not guaranteed to find a path.

The results of this study concern the closed-path planning problem in discrete environments through rolling for only the platonic solids with free-obstacles. In future work, some more case studies will be considered such as rolling platonic solids and the general convex polyhedra with obstacles in known environments. Future work concerns the extensions of the proposed algorithm for rolling convex polyhedra on 3D convex surfaces such as rolling a soccer ball on a convex surface. Finally, the algorithm will be optimized in terms of storage capacity and executing time.

7 ETHICAL ISSUES

This research paper will be conducted by the Ph.D. candidate under the guide of supervisors without any ethical issues. No any sensitive data and no harmful chemicals will be used in this research.

8 COMPLETION TIMELINES

Activity	Year				2020				2021			
	Jan	Mar	May	Aug	Oct	Dec	Jan	Mar	May	Aug	Oct	Dec
Milestone 2												
Optimal Algorithm												
Simulated approaches												
Result & Validation												
Research papers												
Milestone 3												
Thesis preparation												

9 DISSEMINATION PLAN

This study solved the problem of closed-path planning for the platonic solids considered for over two decades. The proposed algorithm significantly generates closed-paths faster than RRT* or breath-first search algorithms. Based on the successful results of the closed-path planning algorithm for rolling polyhedrons on discrete surfaces, this study can be submitted to Science journal in the Report category.

References

1. Zhang, H.-y., Lin, W.-m. & Chen, A.-x. Path Planning for the Mobile Robot: A Review. *Symmetry* **10**, 450 (2018).
2. LaValle, S. M. *Planning Algorithms* ISBN: 0521862051 (Cambridge University Press, New York, NY, USA, 2006).
3. Kavraki, L. E., Svestka, P., Latombe, J. & Overmars, M. H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* **12**, 566–580. ISSN: 1042-296X (1996).
4. Chun Sheng, C. & Roth, B. On the planar motion of rigid bodies with point contact. *Mechanism and Machine Theory* **21**, 453–466. ISSN: 0094-114X. <http://www.sciencedirect.com/science/article/pii/0094114X8690128X> (1986).
5. Chun Sheng, C. & Roth, B. *On the spatial motion of a rigid body with point contact* in *Proceedings. 1987 IEEE International Conference on Robotics and Automation* **4** (1987), 686–695.
6. Cai, C. & Roth, B. *On the spatial motion of a rigid body with line contact* in *Proceedings. 1988 IEEE International Conference on Robotics and Automation* (1988), 1036–1041 Vol.2.
7. Borisov, A. V., Fedorov, Y. N. & Mamaev, I. S. Chaplygin ball over a fixed sphere: an explicit integration. *Regular and Chaotic Dynamics* **13**, 557–571. ISSN: 1468-4845. <https://doi.org/10.1134/S1560354708060063> (2008).
8. A. Bicchi, Y. C. & Marigo, A. Reachability and steering of rolling polyhedra: a case study in discrete nonholonomy. *IEEE Transactions on Automatic Control*, 710–726 (2004).
9. Aiyama, Y., Inaba, M. & Inoue, H. *Pivoting: A new method of graspless manipulation of object by robot fingers* in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems* **1** (), 136–143.
10. Erdmann, M., Mason, M. T. & Vanecek, G. *Mechanical parts orienting: the case of a polyhedron on a table* in *Proceedings. 1991 IEEE International Conference on Robotics and Automation* (Apr. 1991), 360–365 vol.1.
11. Lavalle, S. M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning* tech. rep. (Department of Computer Science; Iowa State University, 1998).
12. Thomas, J., Blair, A. & Barnes, N. *Towards an efficient optimal trajectory planner for multiple mobile robots* in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* **3** (2003), 2291–2296.
13. Lamiraux, F. & Lammond, J.-P. Smooth motion planning for car-like vehicles. *IEEE Transactions on Robotics and Automation* **17**, 498–501 (2001).
14. Cheng, P., Shen, Z. & Lavalle, S. M. RRT-Based Trajectory Design for Autonomous Automobiles and Spacecraft. *Archives of Control Sciences* **11**, 167–194 (2001).
15. Conner, D. C., Rizzi, A. A. & Choset, H. *Composition of local potential functions for global robot control and navigation* in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)* **4** (), 3546–3551 vol.3.
16. Marigo, A., Chitour, Y. & Bicchi, A. *Manipulation of polyhedral parts by rolling* in *Proceedings of International Conference on Robotics and Automation* **4** (1997), 2992–2997 vol.4.
17. Marigo, A., Ceccarelli, M., Piccinocchi, S. & Bicchi, A. Planning Motions of Polyhedral Parts by Rolling. *Algorithmica* **26**, 560–576. ISSN: 1432-0541. <https://doi.org/10.1007/s004539910024> (2000).
18. Dai, J. S. Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory* **92**, 144–152 (Oct. 2015).
19. Tarjan, R. Depth-first search and linear graph algorithms. *SIAM Journal of Computing* **1**, 146–160 (Mar. 1972).

A APPENDIX