DEPARTMENT OF MECHANICAL ENGINEERING

Curtin University

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)

February 25, 2020

Contents

1 ABSTRACT

In the modern manufacturing industry, path planning in industrial robot applications is an important step to find the shortest direction of achieving a task. Among the path planning algorithms, graph search or exploration methods has been applied for robotic in a high-dimensional space. In the geometrical scenarios, the problem of path planning of polyhedra with rolling contact is considered. However, their rolling behaviour with returning to the initial configuration in different orientation has remained unexplored. To tackle the problem, this study proposes a path planning method for regular platonic solids through rolling contract on a plane based on an improved tree search algorithm. The results reveal that the proposed path planning method can enhance the efficiency of the planning for regular convex polyhedra. Consequently, Matlab simulations are conducted in order to demonstrate the proposed algorithm in terms of finding the shortest path of rolling the regular platonic solids in a discrete environment.

2 INTRODUCTION

Path planning algorithm is one of the challenge problems in nonholonomic systems to achieve the dexterous manipulation of objects in an unknown or part-known environment. This problem is mainly applied for the fields of robotics, artificial intelligent 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 in 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 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.

Planning techniques are categorized into different aspects. The basic idea of discrete path planning in the 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 the continuous curvature on smooth curves 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 [13]. 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 spacecrafts. 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.

Not much work has been done in path planning under considering rolling contact. Some types of moving polyhedral parts have 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]. The paper presented some results of changing an orientation of a polyhedron based on edges under considering the convex polyhedral parts for the planning task 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 under considering the given tolerance which leads to reach an orientation closer. The paper proposed a concept of path planning algorithm where achieving the goal configuration within the tolerance 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 for each of vertices. The algorithm was given a polyhedron with a set of geometrical parameters and a desired final configuration. 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 more general polyhedron.

Therefore, in this study, we propose a discrete path planning algorithm based on tree exploration method for the five types of platonic solids including cube, tetrahedron, octahedron, icosahedron, and dodecahedron. The study is organized as follows. Section 3 covers the general properties in geometrical aspect of the five types platonic solids. Section 4 describes 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 conclude the paper.

3 PROBLEM FORMULATION

Five types of Platonic Solids: Platonic solids properties: The platonic solids are also called regular

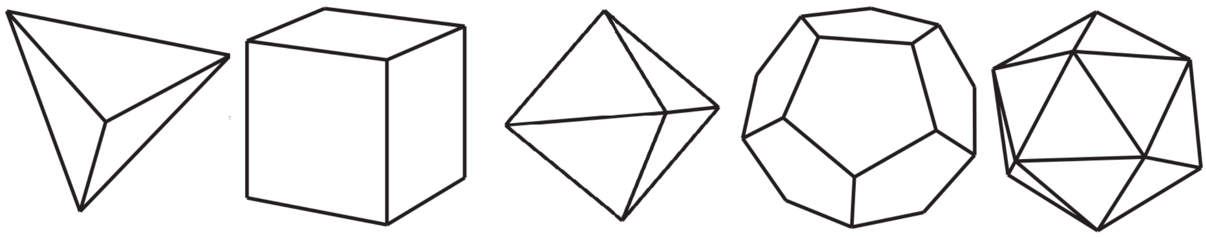


Figure 1: Platonic solids. From left to right: Tetrahedron, Cube, Octahedron, Dodecahedron, and Icosahedron

polyhedra have the convex polyhedra properties. There are only five solids namely cube, tetrahedron, octahedron, dodecahedron and icosahedron. 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.

Here is your table 1

Table 1: Properties of polyhedron

	Faces	Edges	Vertices	Edges on each face	Edges meeting at each vertices
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

Here is your table 2

Table 2: Dimensional of platonic solids

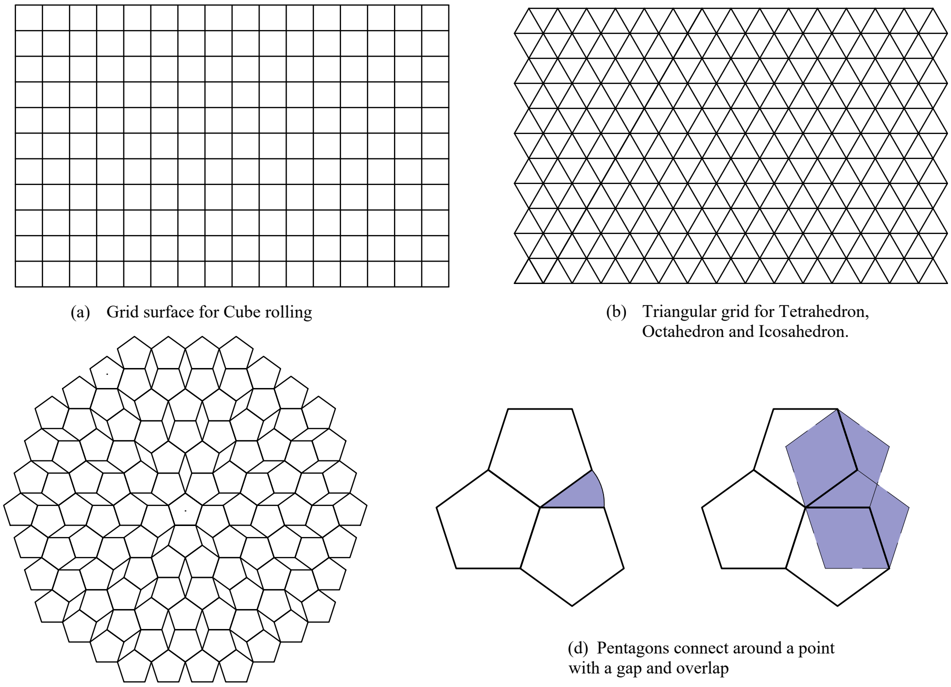
	r_d	ρ	R	dihedral angles (β)
Tetrahedron	$\frac{1}{12}\sqrt{6}$	$\frac{1}{4}\sqrt{2}$	$\frac{1}{4}\sqrt{6}$	$\cos^{-1}(\frac{1}{3})$
Cube	$\frac{1}{2}$	$\frac{1}{2}\sqrt{2}$	$\frac{1}{2}\sqrt{3}$	$\frac{1}{2}\pi$
Octahedron	$\frac{1}{6}\sqrt{6}$	$\frac{1}{2}$	$\frac{1}{2}\sqrt{2}$	$\cos^{-1}(-\frac{1}{3})$
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 shape for cube, triangle shape for tetrahedron, octahedron, and dodecahedron, pentagon shape for 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 triangular grid with different rotation angle. In physics, the rotation angle of the cube is $\pi/2(rad)$ 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, the 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 $3 * 108° = 324°$, which is different $360°$ of the full circle. Another case of four overlap pentagons with $4 * 108° = 432°$ is greater than the circle of $360°$. The path planning through rolling of the dodecahedron solid can be categorized into two these cases. It would be found the possible paths in the first case of dodecahedron without overlap rolling while the second case with overlap rolling cannot guarantee the paths.



(a) Grid surface for Cube rolling

(b) Triangular grid for Tetrahedron, Octahedron and Icosahedron.

(c) The radial arrangement from a single central decogon

(d) Pentagons connect around a point with a gap and overlap

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 the line contact. Rolling the platonic solids means rolling all its vertices in $3D$ environment 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. The method is explained in the Section 5.

Algorithms: Due to the different surface contacts, there are three types of direction for the rolling of platonic solids. As shown in the Figure 3, the cube has four directions with the square surface contact while tetrahedron, octahedron and icosahedron have three rolling directions with the triangular surface contact. The dodecahedron with pentagon surface contact has five rolling directions. In the case of 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 though their edges.

The Algorithm 1 shows that path planning for cube rolling based on tree graph search has some important steps. The first step is to initial the coordinates and the orientations of the initial cube and the target cube which is stored as the initial path. The same as 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 expand with only three directions to avoid return the previous positions. An example for this step is that from the initial coordinate the cube achieves a new position after doing rolling for 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.

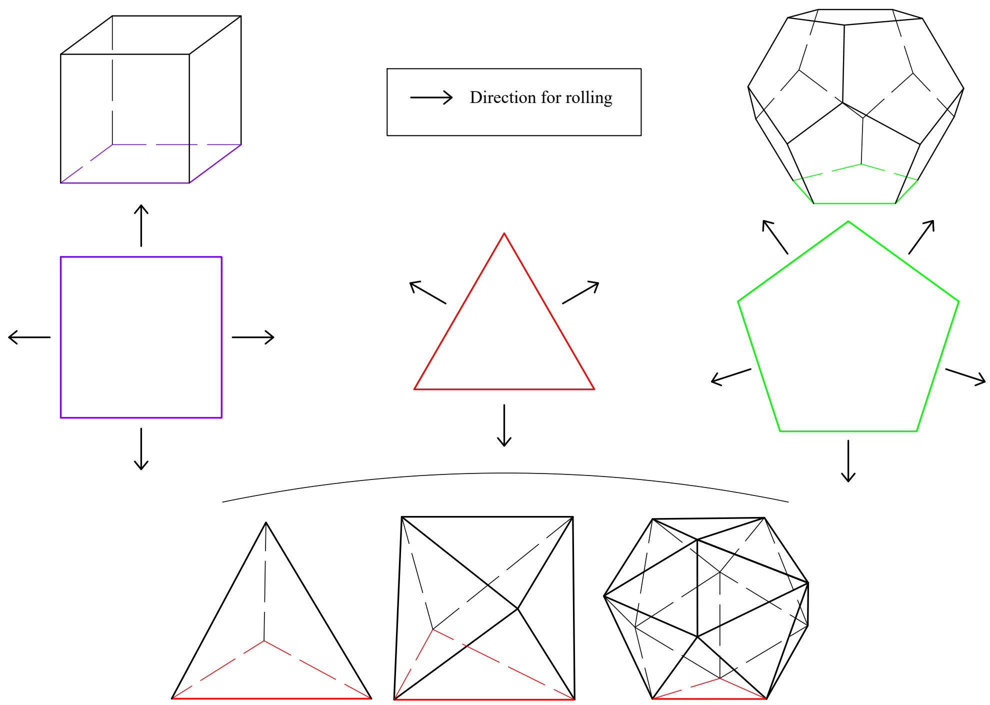


Figure 3: Rolling direction for each types of platonic solids

4.2 Tree Exploration Algorithm

The node tree exploration for searching algorithm described in Algorithm 1 is similar to non-recursive depth-first-search algorithm. The graph search in the 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 nodes 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 iterations, a tree with a node including $3D$ coordinate and orientation is stored in each levels. 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 path-finding process. Only the case of dodecahedron has the 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 the Figure 2b.

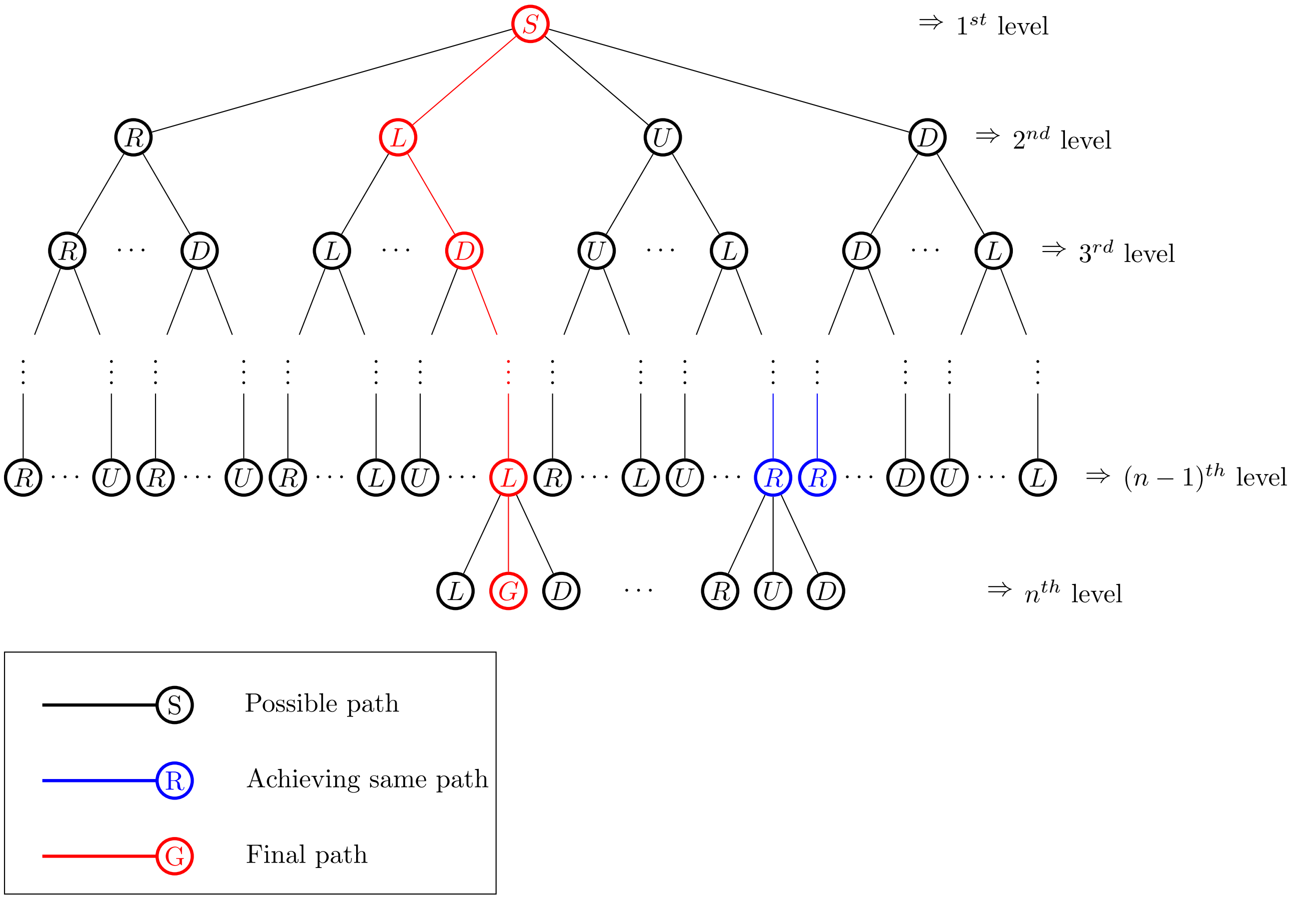


Figure 4: Tree Exploration of Cube Rolling

Starting form the root S, path planning based rolling of the cube model at the first level of expansion will generate to four different direction 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 two updated nodes R (blue node) 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 executing time is shortest based on the condition of achieving goal configuration.

Algorithm 1 Path planning based rolling contact for Cube model.

```
procedure CubePathPlanning(S_p, G_p) ▷ Find the shortest path from start to goal position with different orientation
    flag ← false
    Path[S_p] ← S_p
    newPoints ← rolling4Directions(S_p)    ▷ Generate first four updated points
    while newPoints != G_p do
        updatedPoints ← treeExploration(newPoints)    ▷ Update new three right rolling models
        n ← size(updatedPoints)
        for i ← 0, n do
            for j ← 1, n do
                if updatedPoints[i] = updatedPoints[j] then
                    remove(updatedPoints[i])
                end if
            end for
        end for
        flag ← checkingTargetPoint(updatedPoints) ▷ Compare updated points with goal point
        if flag = true then
            return Path[S_p, G_p]    ▷ Store new point to Path
        end if
        newPoints = updatedPoints
    end while
    return "no path found"
end procedure
procedure rolling4Directions(S_p)    ▷ Generate new points in different direction of rolling
    (newRightPoint, newLeftPoint, newUpPoint, newDownPoint) ← rollingContact(S_p)
    return newPoints ← (newRightPoint, newLeftPoint, newUpPoint, newDownPoint)
end procedure
procedure treeExploration(newPoints)
    if dir = right then
        updatedPoints ← (newRightPoint, newUpPoint, newDownPoint)
    else if dir = left then
        updatedPoints ← (newLeftPoint, newUpPoint, newDownPoint)
    else if dir = up then
        updatedPoints ← (newRightPoint, newLeftPoint, newUpPoint)
    else
        updatedPoints ← (newRightPoint, newLeftPoint, newDownPoint)
    end if
    return updatedPoints
end procedure
procedure checkingTargetPoints(updatedPoints,G_p)
    if updatedPoints = G_p then    ▷ Consider both position and orientation
        flag ← true
    end if
    return flag
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 has the same length with a ($a = 1$) and one of the faces of the platonic solids contact to OXY.

5.1 Cube solid

Properties: The cube has a length which is the same as the length of side of each grid square. The only way to move from initial position to goal position is by rolling from square to square without moving diagonal. The Figure 5 shows the first three layers of cube path finding on the grid. The algorithm implements in $O(|E|^3)$ running time from the second layer. The executed time can be reduced when the updated cubes achieved the same configurations. The $*$ position in the grid is occupied by the two updated cubes with different orientations. To be more visualized, the Figure 6(a) indicates the first step of rolling of the cube with its coordinates in three arrows red, green and blue colors. Four paths of cube rolling are shown in the Figure 6(b).

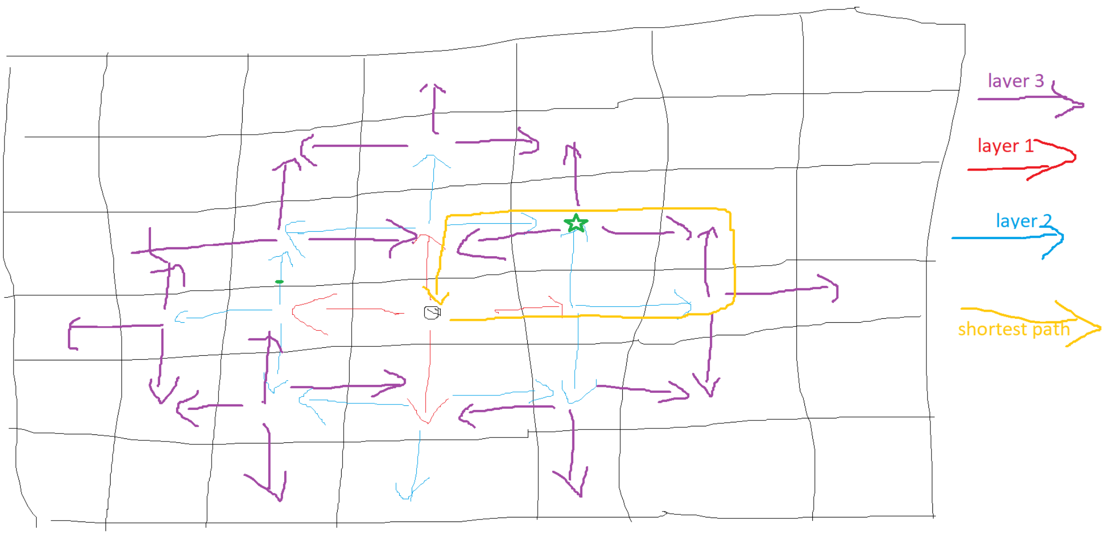
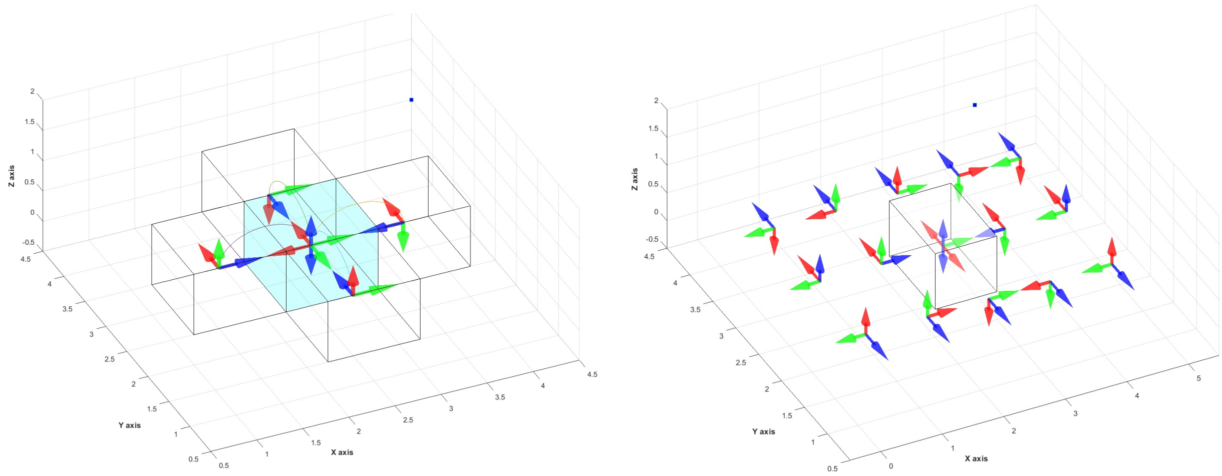


Figure 5: The first three layers of cube rolling

Path planning: ABC here.

Result:



(a) The initial configuration is the same position but different orientation with goal configuration

(b) First four paths of the cube rolling

Figure 6: Blah Blah

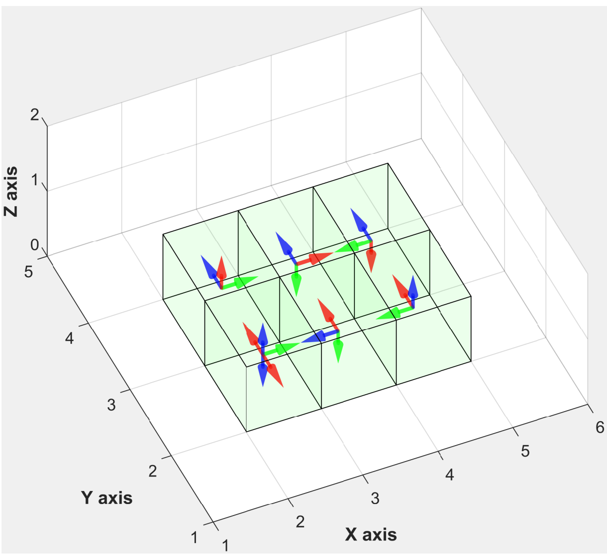
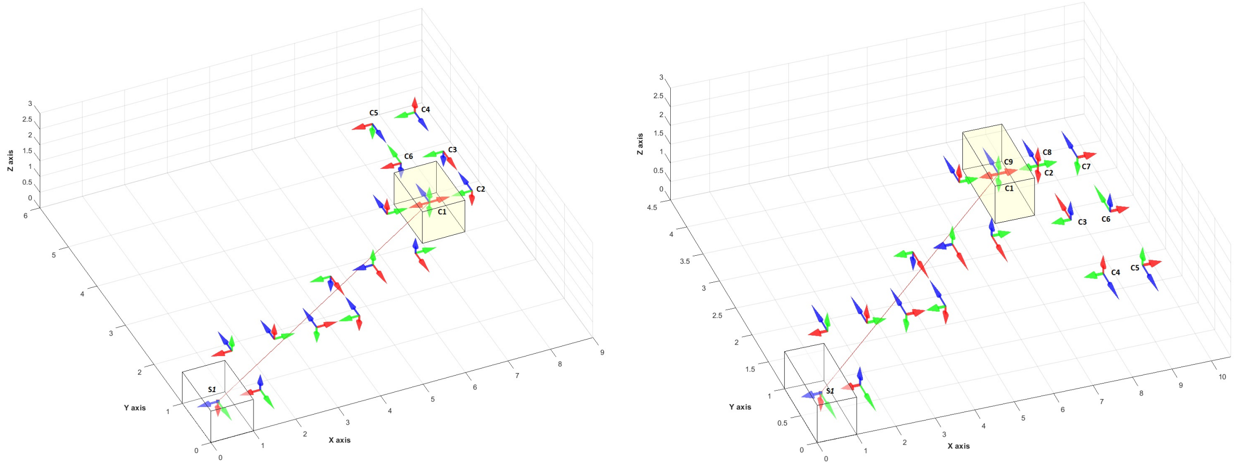


Figure 7: Shortest path of cube rolling

Although considering the case study within

The proposed algorithm in this study can be applied for the case study of long distance between the initial and goal configurations. Figure 8 shows an example with two different paths. Shortest path-finding algorithm is added to the original algorithm to find the shortest path from start point to the goal point. After finishing this step, the cube updated to a new orientation with different orientation at the goal configuration. Then, the original algorithm will be implemented from the updated cube to the goal configuration. Assume that the initial configuration is at S_1 and the goal configuration is at $S2$. In the first step, the red line segment S_1S_2 shows the shortest distance from start position to goal position. The cube will roll through this line segment and achieve the updated orientation at the goal position called shortest path for the case of long distance.



(a) Path1 (b) Path2

Figure 8: The case study of long distance between the initial and goal configurations

5.2 Tetrahedron solid

Properties As can be seen from the 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 α determined by supplementary angles $\beta = \arctan(2\sqrt{2})$ or $\alpha = \pi - \beta = \pi - \arctan(2\sqrt{2})$.

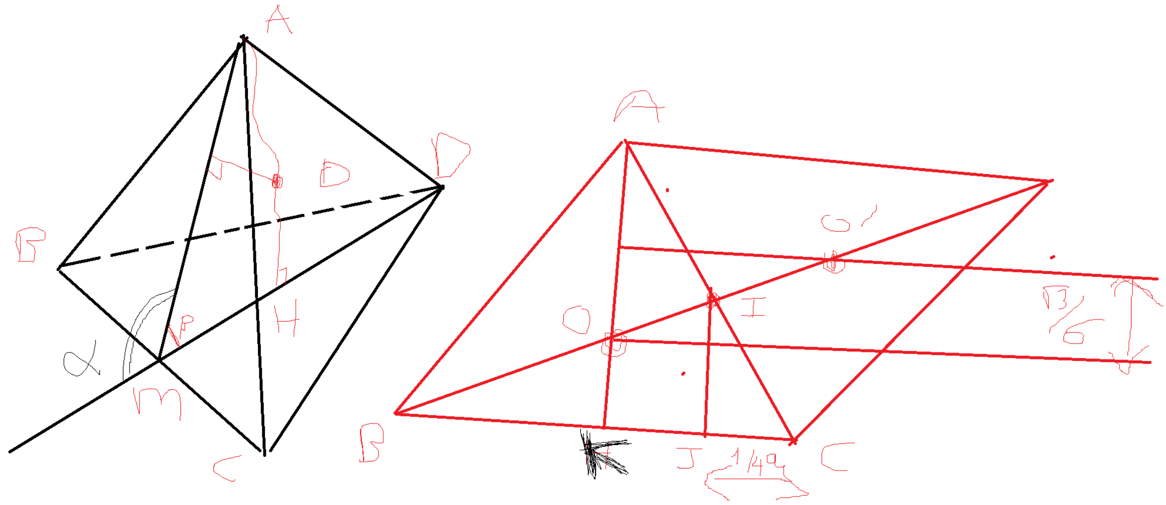
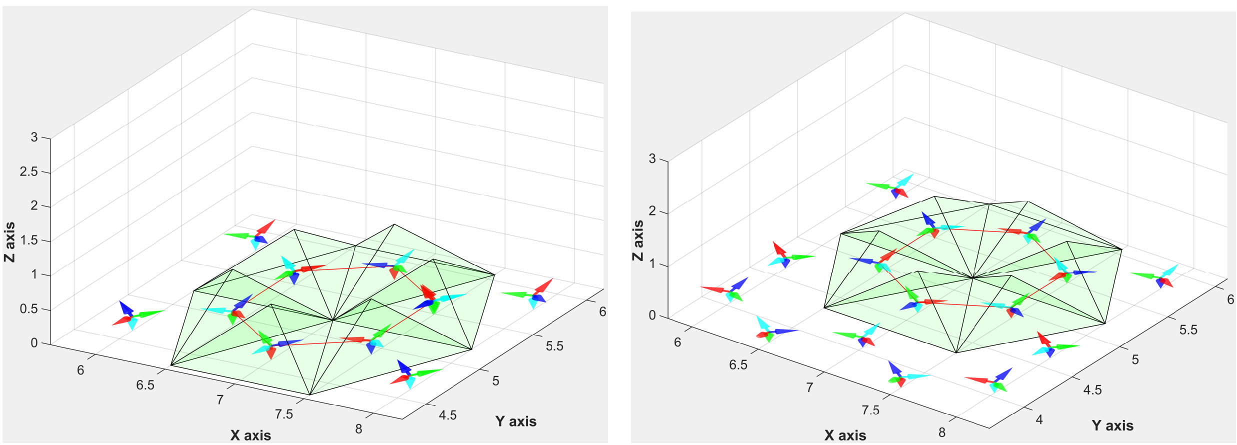


Figure 9: Tetrahedron geometrical properties

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



(a) Tetrahedron path 1

(b) Tetrahedron path 2

Figure 10: The two of four cases

5.3 Octahedron solid

Properties: Dennis also went his own way and divided the sides of the triangles into equal-angles (as measured from the center of the geodesic), instead of equal-length pieces. This technique is slightly more effective at evenly distributing the triangles across the surface of the sphere. For example, compare an octahedron subdivided with frequency 20, using the linear technique (as outlined by the quiz) versus the angular technique Dennis used in this picture.

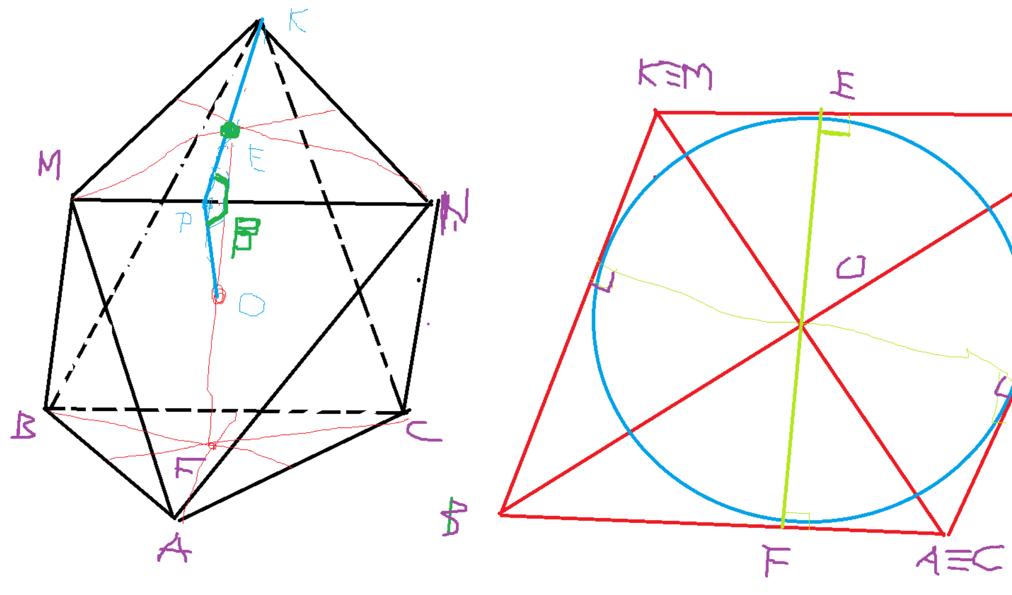
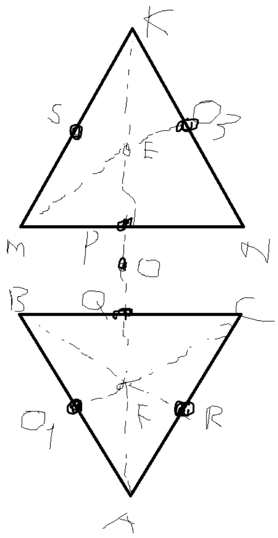


Figure 11: Octahedron geometrical properties

From Figure 11 we have:



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

WTF here

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

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

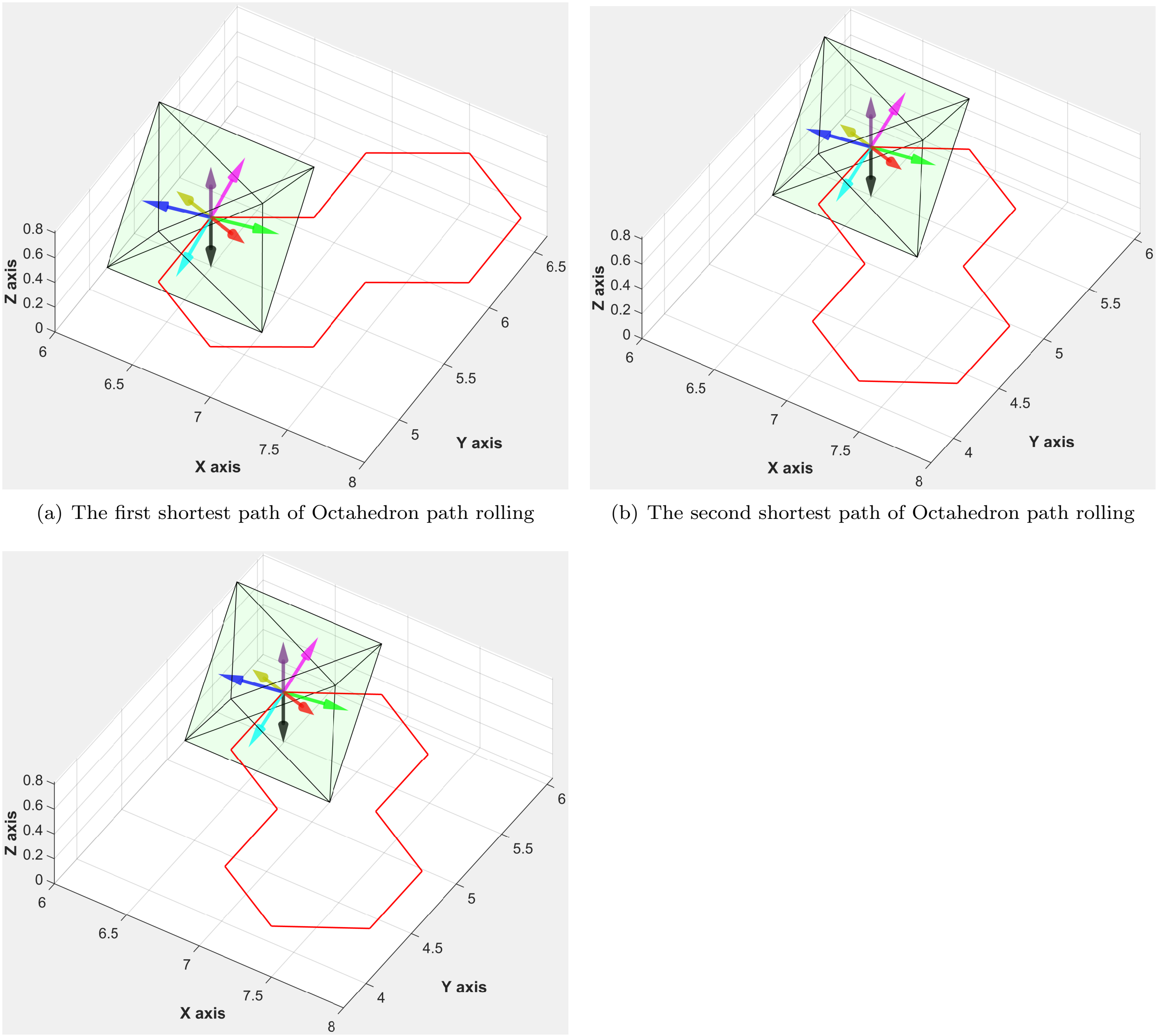
Due to $\angle POK = 90°$, we have

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

Applying the theory of the equilateral triangle

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

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 the 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).



(a) The first shortest path of Octahedron path rolling

(b) The second shortest path of Octahedron path rolling

(c) The second shortest path of Octahedron path rolling

Figure 12: Three shortest paths of octahedron based rolling

5.4 Icosahedron solid

Properties: The convex regular icosahedron in the 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 $ABCEF$.

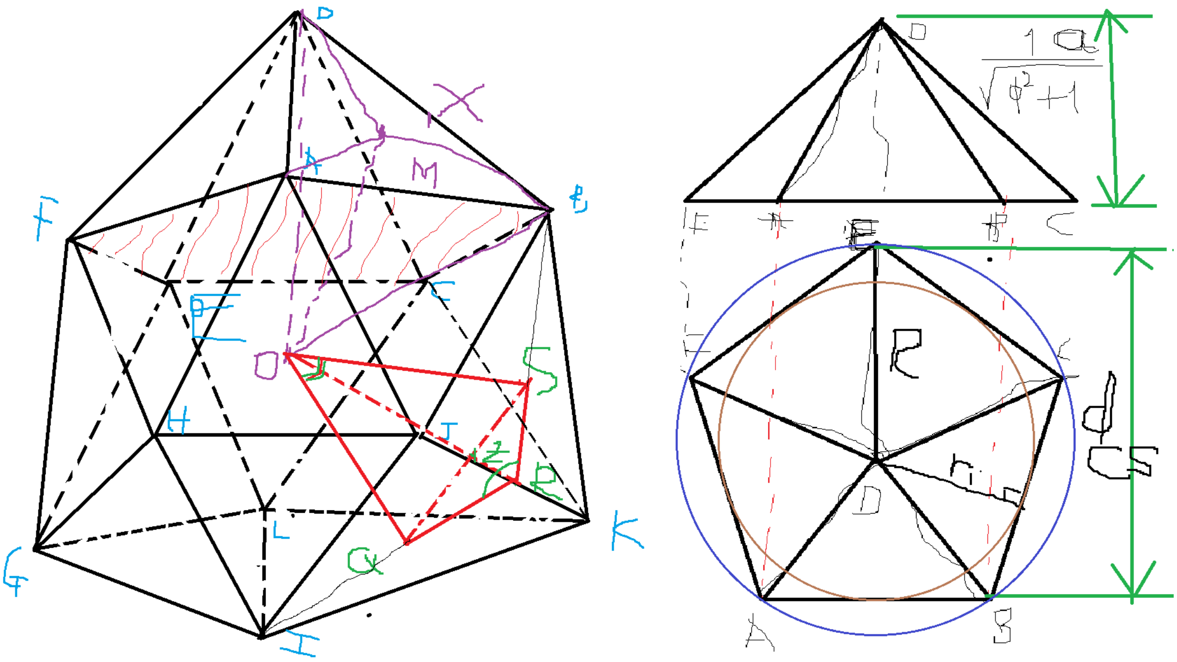


Figure 13: Icosahedron geometrical properties

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 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 the 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.

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

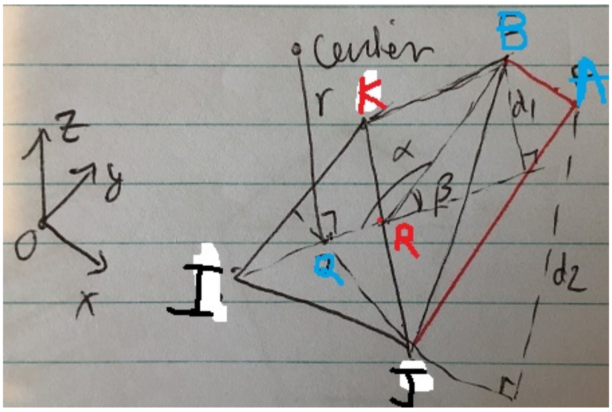
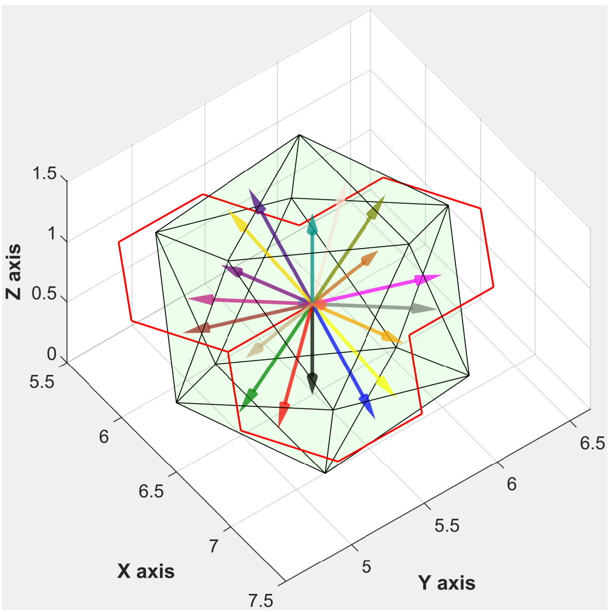
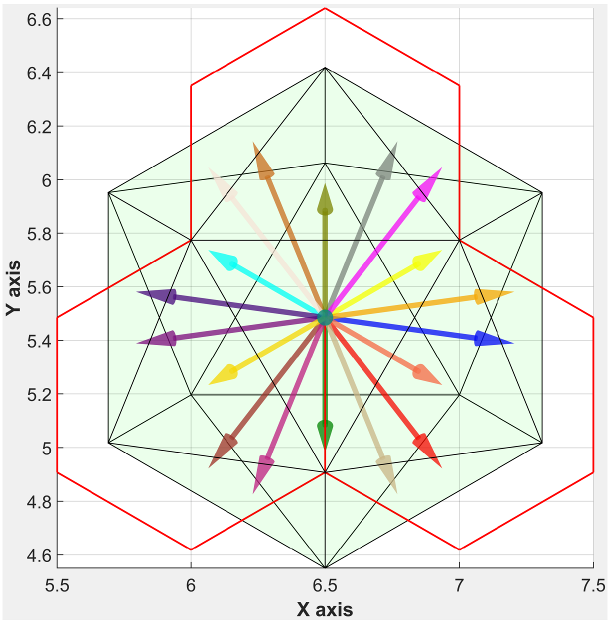


Figure 14: Rotation angle and rotation axis

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 the 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 the Figure 15(b). The total length of this icosahedron path equals $14a\frac{\sqrt{3}}{3}$ (edge length of the octahedron is a).



(a) The first shortest path of Icosahedron path rolling (Red line)



(b) The top view of path planning for Icosahedron

Figure 15: Path of icosahedron rolling though edges

5.5 Dodecahedron solid

Properties: An dodecahedron has 12 faces and 20 vertices of which generates a pentagon as shown in Figure 16. It will be assumed that the coordinates $Oxyz$ lie 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. The Figure 16 indicates the lengths of each vertices from l_1 to l_4 and the angles α_1 to α_4 which correspond to the five sides of a pentagon.

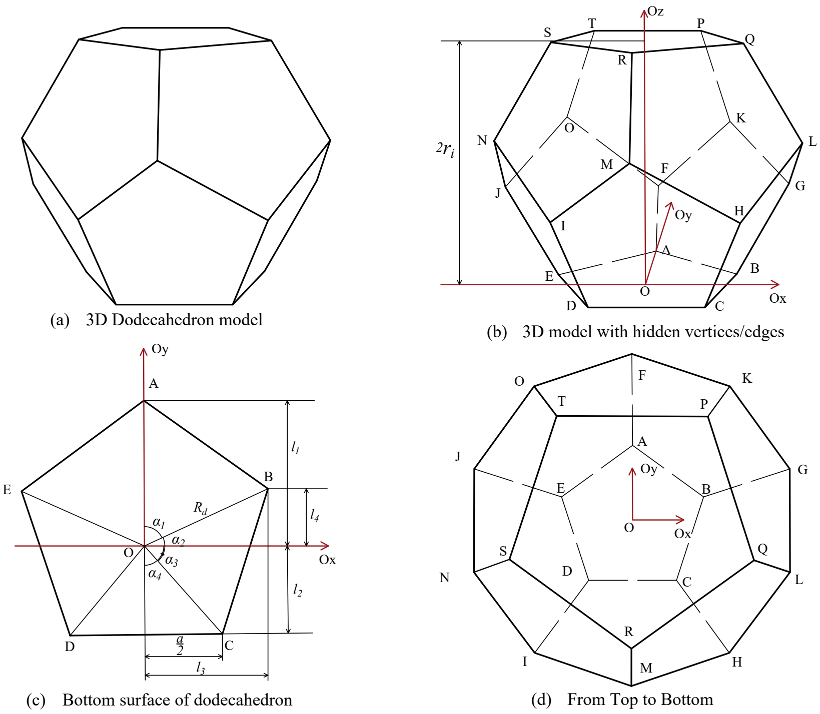


Figure 16: Dodecahedron's vertices.

The path planning will implement on a surface but it will be considered in $3D$ spaces. Then, each of vertices will be determined on $3D$ coordinates such as the vertices A has coordinate with $[A_xA_yA_z]$. 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 determined by $\alpha_2 = \frac{\pi}{2} - \alpha_1 = \frac{\pi}{2} - \frac{2\pi}{5} = \frac{\pi}{10}$ and $\alpha_3 = \alpha_1 - \alpha_2 = \frac{2\pi}{5} - \frac{pi}{10} = \frac{3\pi}{10}$.

From the 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 $l_4 = l_1 \sin\alpha_2$. Referencing to the properties of a dodecahedron with length a, the radius of an inscribed sphere is $r_i = \frac{a}{20}\sqrt{10(25 + 11\sqrt{5})}$ and the circumscribed sphere radius is $r = a\frac{\sqrt{3}}{2}\frac{1+\sqrt{5}}{2}$.

There are total 20 vertices of a dodecahedron. This article focuses on the rolling contact to $2D$ surface, the bottom surface of the dodecahedron integrated to the Oxy which contact to the $2D$ surface. This condition express the Oz dimension of the vertices $ABCDE$ equal to 0 or $A_z = B_z = C_z = D_z = E_z = 0$. Then $P_z = Q_z = R_z = S_z = T_z = 2.r_i = \frac{a}{10}\sqrt{10(25 + 11\sqrt{5})}$.

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:

$$\begin{aligned} AF^2 &= a^2 = (A_x - F_x)^2 + (A_y - F_y)^2 + (A_z - F_z)^2 \\ BF^2 &= (2l_3)^2 = (B_x - F_x)^2 + (B_y - F_y)^2 + (B_z - F_z)^2 \end{aligned} \tag{1}$$

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:

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

Solving F_y and l_5 gives:

$$
\begin{aligned}
F_y &= \frac{a^2 - (2l_3)^2 - (l_1^2 - l_3^2 - l_4^2)}{2(l_4 - l_1)} \\
l_5 &= \frac{1}{\sqrt{2}} \sqrt{a^2 + (2l_3)^2 - (F_y - l_1)^2 - (F_y - l_4)^2 - l_3^2}
\end{aligned}
\tag{3}
$$

From these equations 1,2,3, all the vertices will be founded in the three-dimensional space. Path planning based on rolling is the motion of all these vertices through edges' contact.

<u>Path Planning</u>:

Experiments: Writing about cube solid properties

Discussion: Q2 & Q3
- Q2: What are the new things you learned after you did whatever you did?
- Q3: What exactly did you do?

- **Discussion**
- *What your results mean*
- *Why it makes a difference*
- **Conclusion**
- *Broader implications*
- *Areas for further study*

6 CONCLUSION AND FUTURE PROSPECTS

In this paper, we have proposed a method of path-finding for platonic solids - regular convex polyhedra - under rolling constraint without sliding. Using tree exploration algorithm can find the path for cube, tetrahedron, octahedron and icosahedron solids. In the first case of dodecahedron with a gap between two pentagons in the initial environment, the algorithm is be applied to find shortest paths while the other case with overlaps is not guaranteed to find a path.

The results of this study concern path planning in discrete environment through rolling for only platonic solids with free-obstacles. In future work some more case studies can be included such as rolling platonic solids with obstacles in the environment, optimal algorithm with reducing the cost of memory while executing the algorithm.

7 TIME LINE

Year	2018			2019				2020				2021		
Activity	Sep	Oct	Dec	Feb	May	Aug	Nov	Feb	May	Aug	Nov	Feb	May	Aug
Candidacy proposal														
Literature review														
Mathematical model														
Simulated approach														
Result & Validation														
Thesis preparation														

Refer to the Data Management Plan in Appendix A.

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).

A APPENDIX