

Contents

1 ABSTRACT	1
2 INTRODUCTION	1
3 PROBLEM FORMULATION	2
4 ALGORITHM\METHODOLOGY	4
4.1 Path Planning Based Rolling Contact	4
4.2 Tree Exploration Algorithm	6
5 EVALUATION	7
6 CONCLUSION AND FUTURE PROSPECTS	14
7 Reviews from Prof. Jonathan Paxman for PhD Candidacy Proposal	17
A APPENDIX	19

1 ABSTRACT

250 words or less, concise summary of research conducted, results obtained, and conclusion reached

Background: Place the question addressed in a broad context and highlight the purpose of the study.

Aim:

Approach: Methods: Describe briefly the main methods or treatments applied;

Significance: Results: Summarize the article's main findings;

Conclusion: Indicate the main conclusions or interpretations. The abstract should be an objective representation of the article, it must not contain results which are not presented and substantiated in the main text and should not exaggerate the main conclusions.

Examples: from "2018 Path Planning of Industrial Robot - RRT"

With the development of modern manufacturing industry, the application scenarios of industrial robot are becoming more and more complex. Manual programming of industrial robot requires a great deal of effort and time. **Therefore**, an autonomous path planning is an important development direction of industrial robot.

Among the path planning methods, the rapidly-exploring random tree (RRT) algorithm based on random sampling has been widely applied for a high-dimensional robotic manipulator because of its probability completeness and outstanding expansion. **However**, especially in the complex scenario, the existing RRT planning algorithms still have a low planning efficiency and some are easily fall into a local minimum.

To tackle these problems, this paper proposes an autonomous path planning method for the robotic manipulator based on an improved RRT algorithm. The method introduces regression mechanism to prevent over-searching configuration space. **In addition**, it adopts an adaptive expansion mechanism to continuously improve reachable spatial information by refining the boundary nodes in joint space, avoiding repeatedly searching for extended nodes. **Furthermore**, it avoids the unnecessary iteration of the robotic manipulator forward kinematics solution and its time-consuming collision detection in Cartesian space. The method can rapidly plan a path to a target point and can be accelerated out of a local minimum area to improve path planning efficiency.

The improved RRT algorithm proposed in this paper is simulated in a complex environment. The results reveal that the proposed algorithm can significantly improve the success rate and efficiency of the planning without losing other performance.

2 INTRODUCTION

- Novelty: Literature review
- Goal: What question you're trying to answer
- Motivation: Why you're asking the question

Guide: *Goal: provide context and encourage reader to read the paper.*

1. *Background and motivation (1 paragraph)*
2. *Overview of the paper and contributions (1-2 paragraphs)*
3. *More details and summary of the approach*
4. *Summary of the results and conclusions.*

Overview: Q4. Why should the community care?

Related work: Q1. What did the community know before you did whatever you did?

Contribution: Q3. Why exactly did you do?

We focus on....

We propose ABC algorithm...

We prove that

We demonstrate the EFG problem through x case studies (Section 3.4). We evaluate the ... (Section 4,5).

In this paper, we present discrete path planning of platonic solids including cube, tetrahedron, octahedron, icosahedron, and dodecahedron. These are types of convex polyhedra with equivalent faces constituted to congruent convex regular polygons....

Not much work has been done in path planning under considering rolling contact. [1] and [2] proposed XYZ method. In their work, they did XYZ (how they did).... However, they did not perform ABC.... => mention Types of rolling contact, and the paper of Z.Li

Literature in the path planning domain describes obstacles avoiding of two general types - continuous and discrete. Continuous path planning[] Discrete path planning[]. However, bla bla bla ...

Bla bla

On the other hand, bla bla bla...

Therefore, in this study, we present three cases of platonic path planning in terms of path finding for the same position and different orientation of initial configuration and goal configuration, direct searching for the long distance between two configuration, and bidirect search within obstacles.

Or: This paper presents a methodology for path planning of platonic solids in known environment. Bla bla ... ref Introduction from "Path planning in multi-scale ocean flows..."

A second contribution of this paper is a technique to compute

We explain our algorithms in Section II. We go over experiments and results in simulation in Section III. We verify our algorithms by executing them on a 3D model of the Statue of David and confirming that collision-free trajectories are efficiently generated. Our primary evaluation metric is time taken for the search. We discuss the performance of each individual search, as well as the advantages and shortcomings. Finally, we discuss possible future steps for this work in Section IV.

3 PROBLEM FORMULATION

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

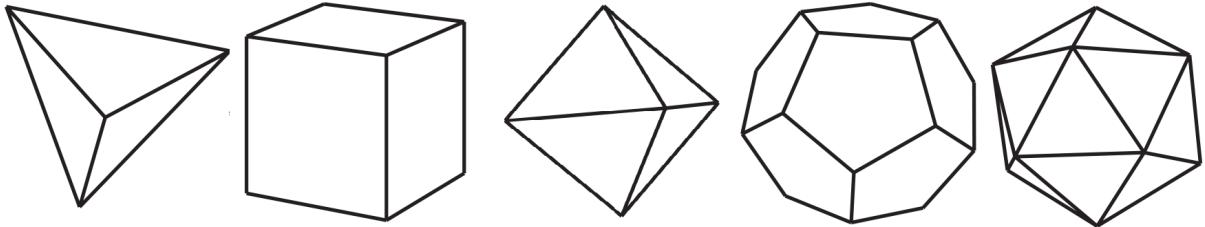


Figure 1: Platonic solids. From left to right: Tetrahedron, Cube, Octahedron, Dodecahedron, and Icosahedron

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

Five types of Platonic Solids:

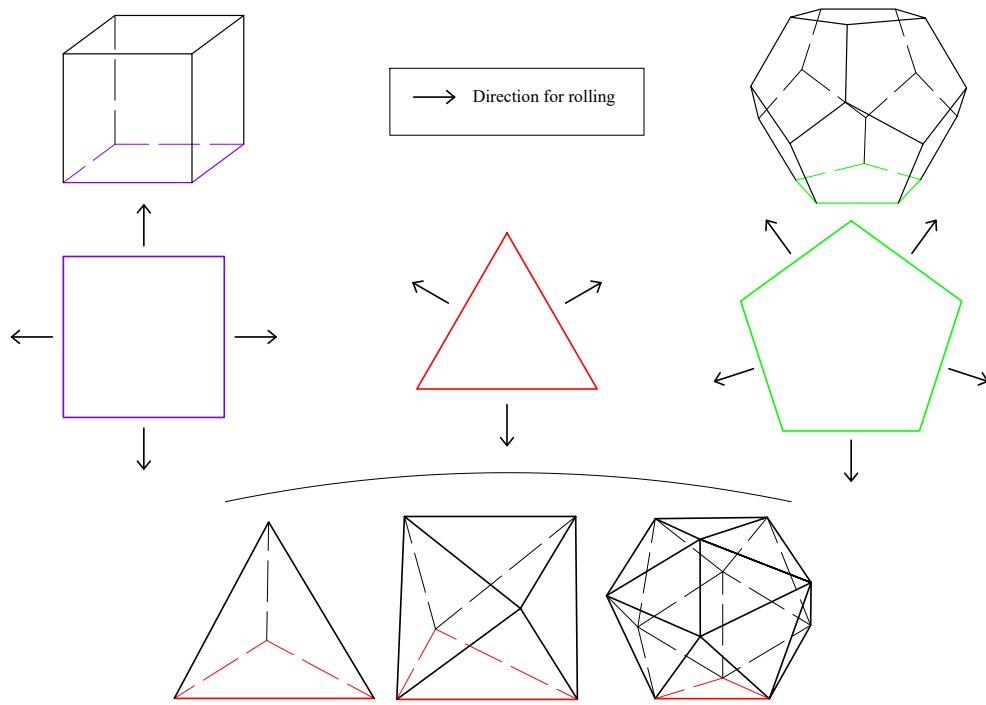


Figure 2: Rolling direction for each types of platonic solids

Version1:

Example:

Algorithm 1 Path planning based rolling contact for Cube model.

```
1: procedure CUBEPATHPLANNING( $S_p, G_p$ )  $\triangleright$  Find the shortest path from start to goal position with  
   different orientation  
2:    $flag \leftarrow false$   
3:    $Path[S_p] \leftarrow S_p$   
4:    $newPoints \leftarrow ROLLING4DIRECTIONS(S_p)$   $\triangleright$  Generate first four updated points  
5:   while  $newPoints \neq G_p$  do  
6:      $updatedPoints \leftarrow TREEEXPLORATION(newPoints)$   $\triangleright$  Update new three right rolling models  
7:      $n \leftarrow size(updatedPoints)$   
8:     for  $i \leftarrow 0, n$  do  
9:       for  $j \leftarrow 1, n$  do  
10:        if  $updatedPoints[i] = updatedPoints[j]$  then  
11:           $remove(updatedPoints[i])$   
12:        end if  
13:      end for  
14:    end for  
15:     $flag \leftarrow CHECKINGTARGETPOINT(updatedPoints)$   $\triangleright$  Compare updated points with goal point  
16:    if  $flag = true$  then  
17:      return "path found"  $\triangleright$  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:    $(newRightPoint, newLeftPoint, newUpPoint, newDownPoint) \leftarrow ROLLINGCONTACT(S_p)$   
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:  
41:   if  $updatedPoints = G_p$  then  $\triangleright$  Consider both position and orientation  
42:      $flag \leftarrow true$   
43:   end if  
44:   return  $flag$   
45: end procedure
```

4.2 Tree Exploration Algorithm

The node tree exploration for searching algorithm described in Algorithm1 is similar to non-recursive depth-first-search algorithm. The graph search in the Figure3 shows the expansion from the *root* with node *S* to multi-level from *level¹*...*levelⁿ*. 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.

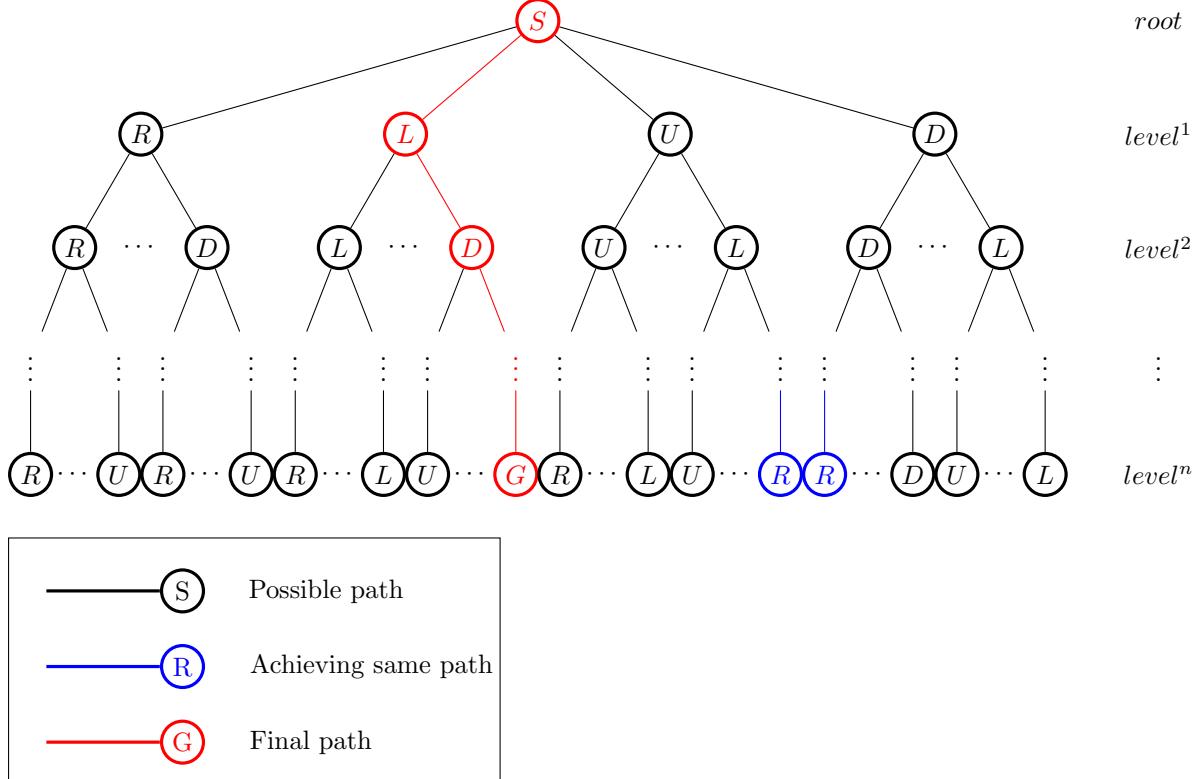


Figure 3: Tree Exploration of Cube Rolling

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.

5 EVALUATION

Simulations: Our algorithm was implemented in MATLAB. Three case studies of path planning are considered for validation: same location and different orientation between initial configuration and goal configuration, long distance between two configuration, and bi-direction path finding.

Cube solid: Writing about cube solid properties

Case study 1: 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. Note how the linear technique has the triangles piling up along the edges of the original face of the octahedron, where the radial technique does a better job of spacing them out.

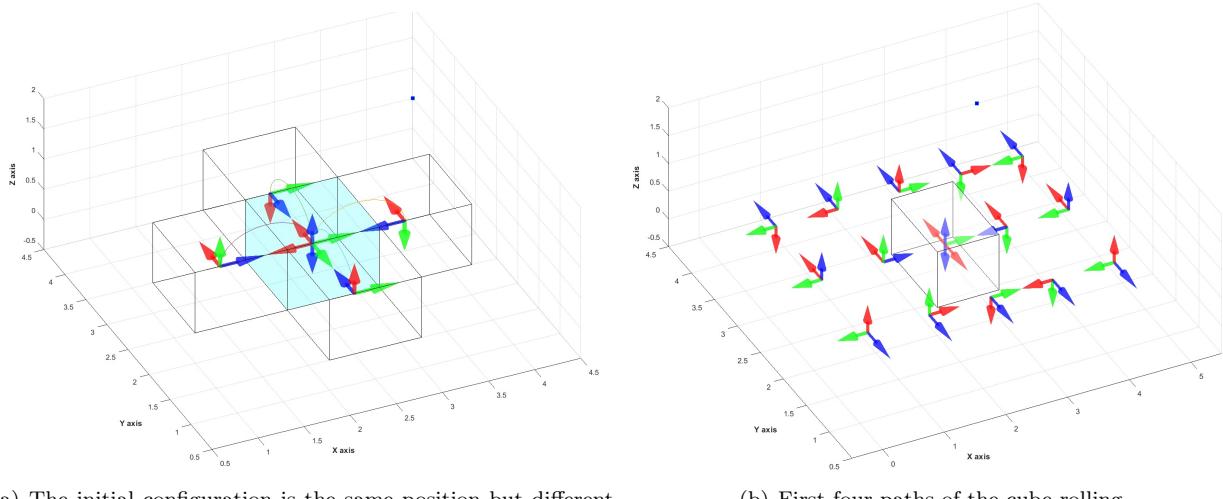


Figure 4: Blah Blah

Result:

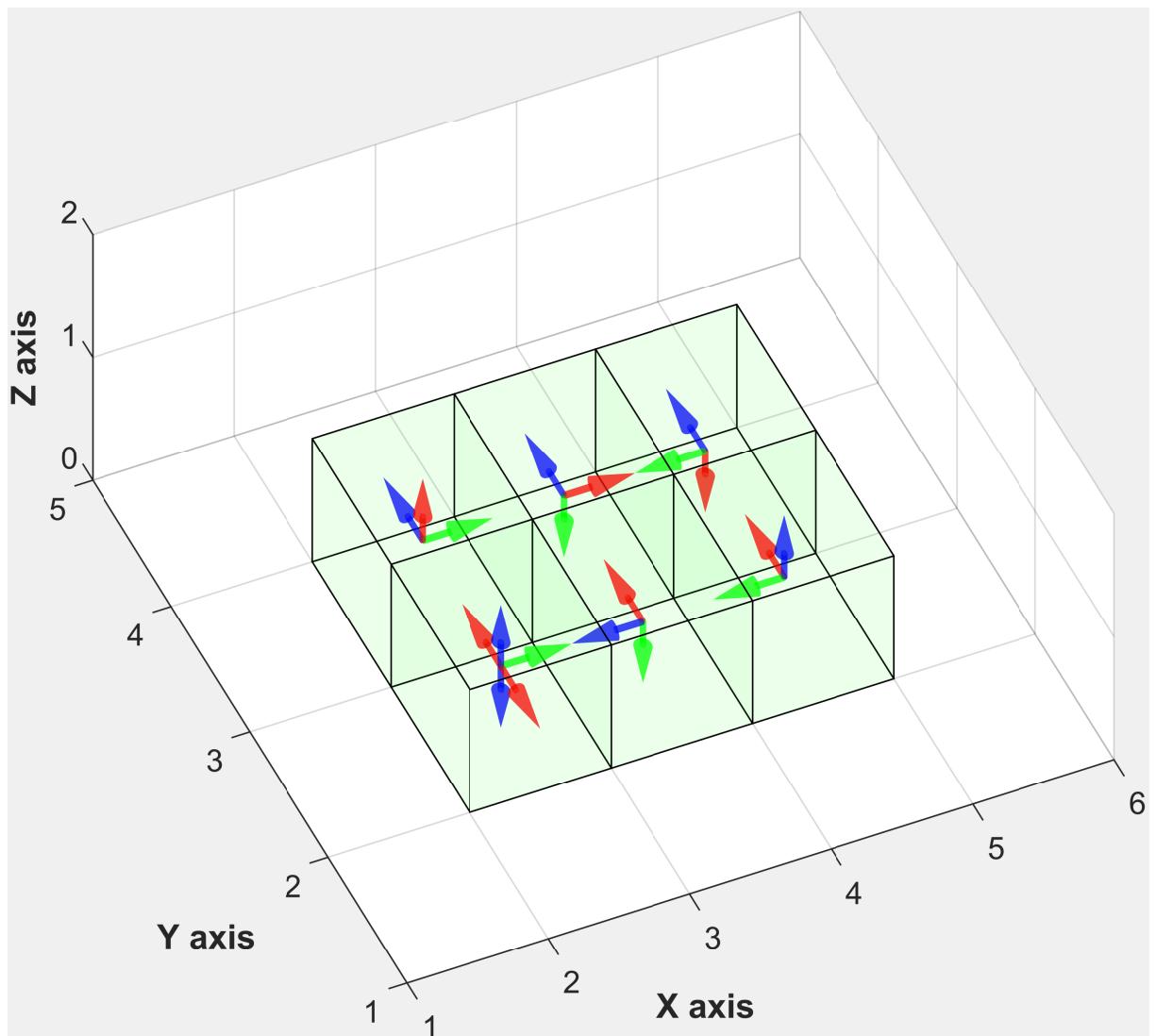


Figure 5: Shortest path of cube rolling

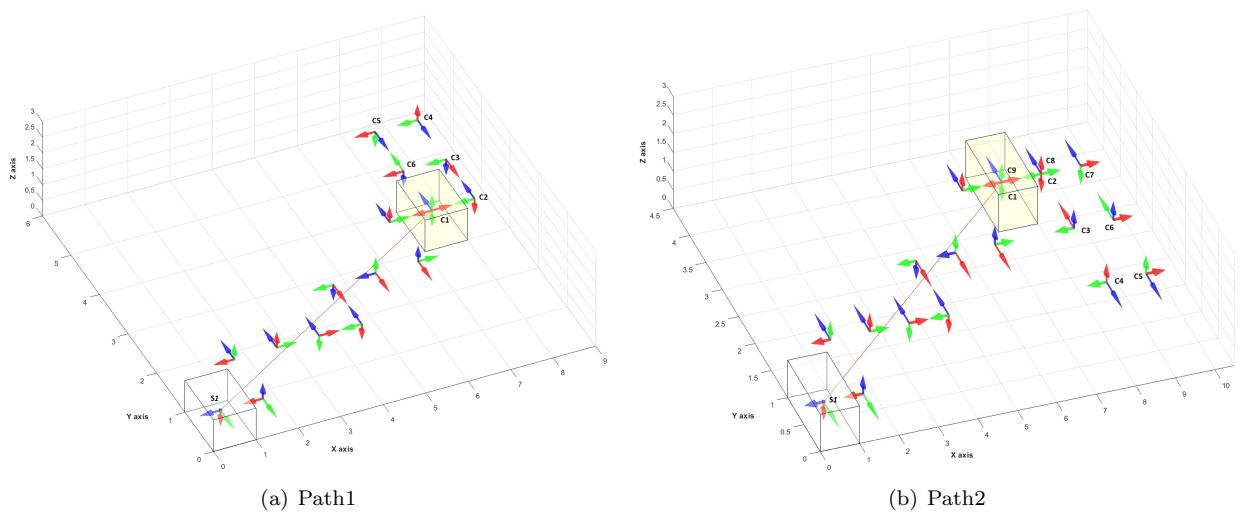


Figure 6: Blah Blah 3

Tetrahedron solid: Writing about cube solid properties

Case study 1: 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. Note how the linear technique has the triangles piling up along the edges of the original face of the octahedron, where the radial technique does a better job of spacing them out.

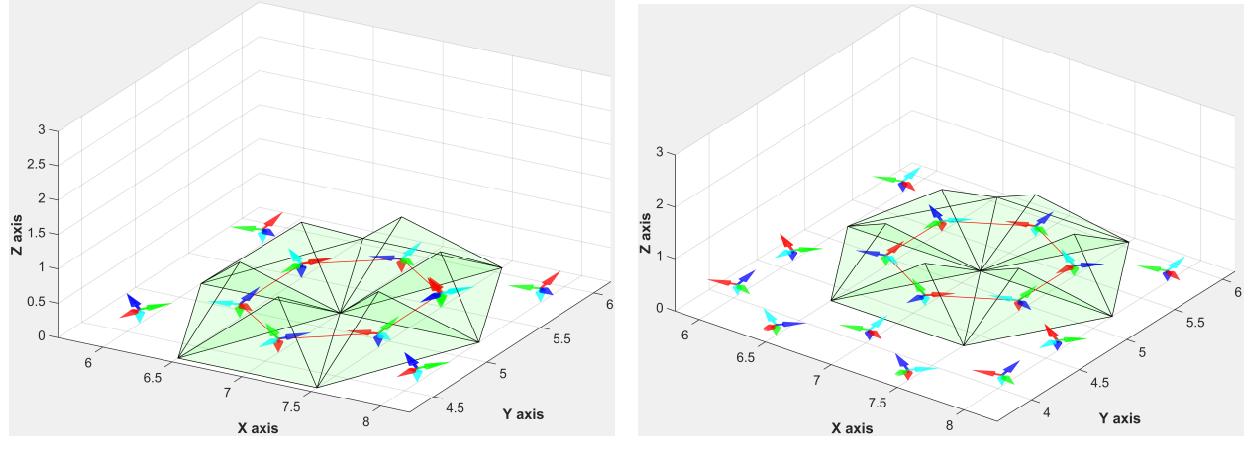
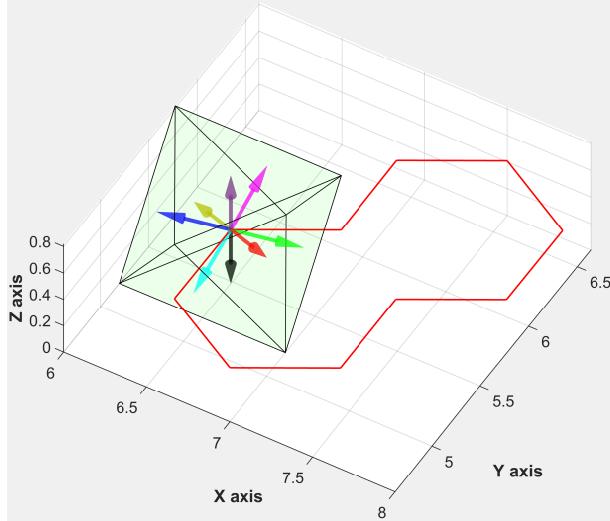
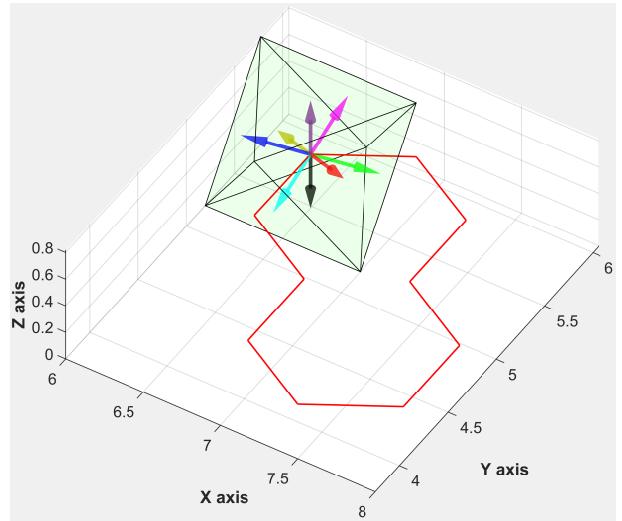


Figure 7: Blah Blah Tetra

Octahedron solid: 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. Note how the linear technique has the triangles piling up along the edges of the original face of the octahedron, where the radial technique does a better job of spacing them out.



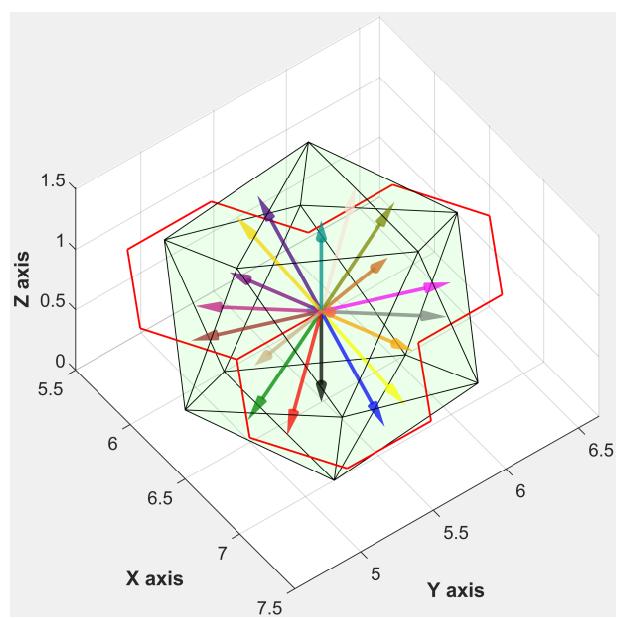
(a) The first shortest path of Octahedron path rolling



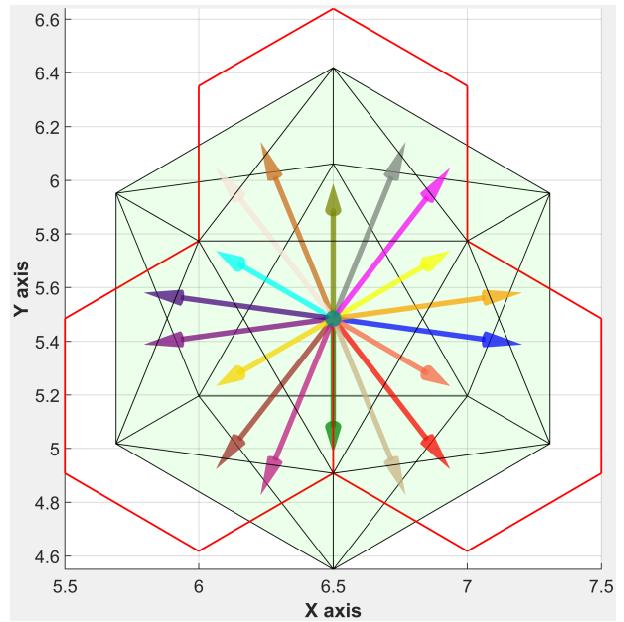
(b) The second shortest path of Octahedron path rolling

Figure 8: Blah Blah Octa

Icosahedron solid: Writing about cube solid properties
 This is the rolling path of icosahedron type 9.



(a) The first shortest path of Icosahedron path rolling



(b) The top view of path planning for Icosahedron

Figure 9: Blah Blah Octa

Dodecahedron:

Properties: An dodecahedron has 12 faces and 20 vertices of which generates a pentagon as shown in Figure 10. 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 10 indicates the lengths of each vertices from d_1 to d_4 and the angles α_1 to α_4 which correspond to the five sides of a pentagon.

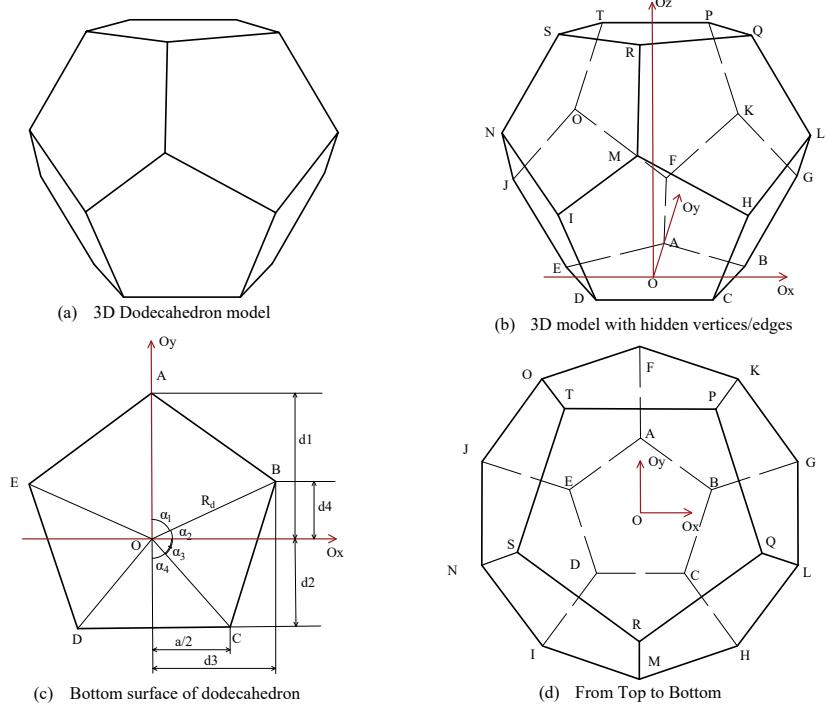


Figure 10: Dodecahedron's vertices.

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 $[A_x A_y A_z]$. 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 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 10(c), these labelled dimensions can be calculated as $d_1 = R_d = \frac{a}{2\sin\alpha_4}$ with R_d is the circumradius of dodecahedron, $d_2 = d_1 \cos \alpha_4$, $d_3 = d_1 \cos \alpha_2$, and $d_4 = d_1 \sin \alpha_2$. 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 $2d_3$. Using the distance properties and squaring the results give:

$$\begin{aligned} AF^2 &= a^2 = (A_x - F_x)^2 + (A_y - F_y)^2 + (A_z - F_z)^2 \\ BF^2 &= (2d_3)^2 = (B_x - F_x)^2 + (B_y - F_y)^2 + (B_z - F_z)^2 \end{aligned} \quad (1)$$

Figure 10(d) shows that $A_x = F_x = 0$, $A_y = d_1$, $B_y = d_4$, $B_x = d_3$, $A_z = B_z$. Define $d_5 = A_z - F_z$, the relations of these equations are:

$$\begin{aligned} a^2 &= (F_y - d_1)^2 + d_5^2 \\ (2d_3)^2 &= (F_y - d_4)^2 + d_5^2 + d_3^2 \end{aligned} \quad (2)$$

Solving F_y and d_5 gives:

$$\begin{aligned} F_y &= \frac{a^2 - (2d_3)^2 - (d_1^2 - d_3^2 - d_4^2)}{2(d_4 - d_1)} \\ d_5 &= \frac{1}{\sqrt{2}} \sqrt{a^2 + (2d_3)^2 - (F_y - d_1)^2 - (F_y - d_4)^2 - d_3^2} \end{aligned} \quad (3)$$

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.

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

Questions: Q4. Why should the community care?

Should do: - Overview of Q1, Q2, and Q3; plus

- What does the community still not know?

Examples: - We have introduced a method of

- Most of our effort has focused on The results of our method often contain We believe that there is significant room for improvement by applying ABC methods to the XYZ problem.

- What do we not do?

In this study, we established a method for ... Although we focused on discrete path planning of platonic solid - regular convex polyhedra in known environment, as illustrated using ABC model and EFG example, the developed method/algorithm can be easily implemented to the complex convex polyhedra such as elipsoil ??. The contributions of this study can be summarized as follows:

SECTION IX. CONCLUDING REMARKS

In this paper, we have introduced Bayesian optimization from a modeling perspective. Beginning with the beta-Bernoulli and linear models, and extending them to nonparametric models, we recover a wide range of approaches to Bayesian optimization that have been introduced in the literature. There has been a great deal of work that has focused heavily on designing acquisition functions; however, we have taken the perspective that the importance of this plays a secondary role to the choice of the underlying surrogate model.

In addition to outlining different modeling choices, we have considered many of the design decisions that are used to build Bayesian optimization systems. We further highlighted relevant theory as well as practical considerations that are used when applying these techniques to real-world problems. We provided a history of Bayesian optimization and related fields and surveyed some of the many successful applications of these methods. We finally discussed extensions of the basic framework to new problem domains, which often require new kinds of surrogate models.

Although the underpinnings of Bayesian optimization are quite old, the field itself is undergoing a resurgence, aided by new problems, models, theory, and software implementations. In this paper, we have attempted to summarize the current state of Bayesian optimization methods; however, it is clear that the field itself has only scratched the surface and that there will surely be many new problems, discoveries, and insights in the future.

[Full Text](#)[Abstract](#)[Authors](#)[Figures](#)[References](#)[Citations](#)[Keywords](#)[Footnotes](#)[Back to Top](#)

Figure 11: First four paths of the cube rolling

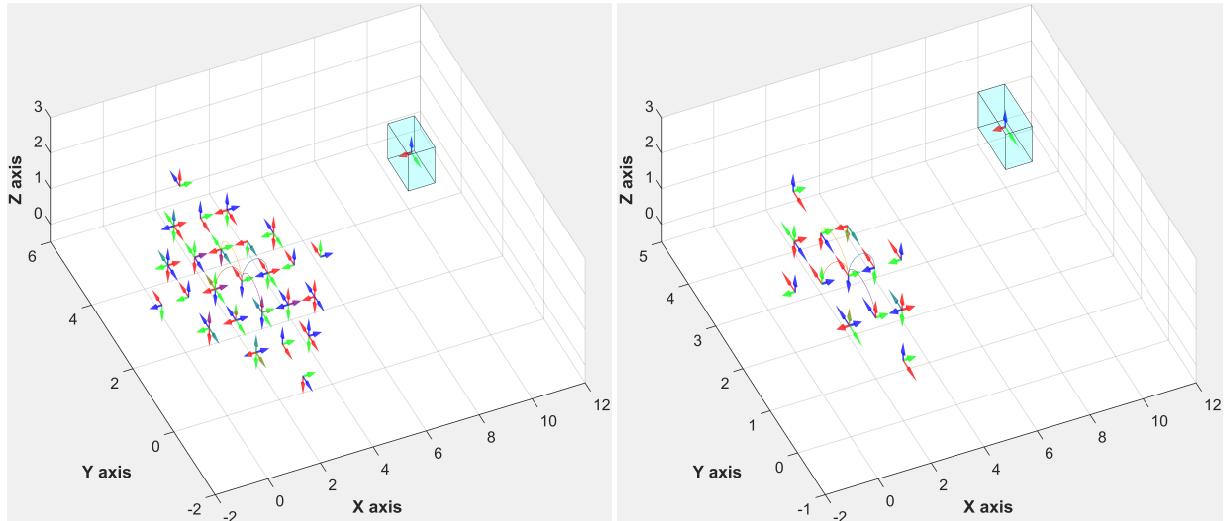
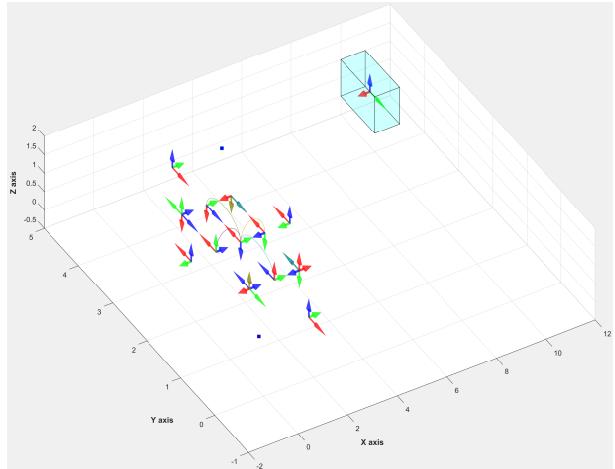
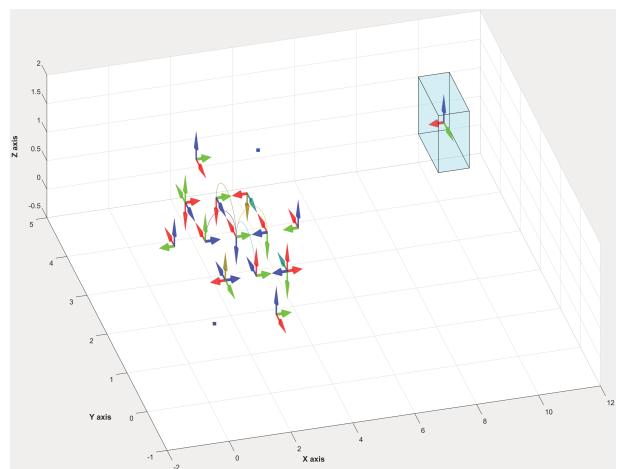


Figure 12: Test



(a) Cube1



(b) First four paths of the cube rolling

Figure 13: Blah Blah

Refer to the Data Management Plan in Appendix A.
[1]

7 Reviews from Prof. Jonathan Paxman for PhD Candidacy Proposal

- Include a discussion of the motivation and advantages for rolling contact for in-hand manipulation
- Reduce the length of the discussion on modelling the kinematics of rolling motion
- Add a brief review of path planning for two general objects under nonholonomic constraints
- Simply the aims: remove specific techniques and algorithms, and describe the broad aim of the project general terms, and in one or two sentences. Ensure that specific objectives are framed so that the aim can be achieved.
- Include a section which describes how a discretised model will be produced such that the discrete planning algorithms described can be applied. How is this discrete model to be obtained from the continuous-time models discussed?
- If optimal planning is discussed, ensure you are specific about in what sense the solution is optimal. In some cases, optimality is not required, only a satisfactory or satisfying solution in the sense of a cost function being below some bound. In such cases, sampling-based solutions (such as RRT) are appropriate.
- Please also review the writing for grammatical correctness (seek some assistance on this if needed).
- Note Robot Operating System (not Software) in Table 1.

[2], [3],[4],[5], [6].

References

1. Barraquand, J. & Latombe, J.-C. Robot Motion Planning: A Distributed Representation Approach. *The International Journal of Robotics Research* **10**, 628–649. ISSN: 0278-3649. <https://doi.org/10.1177/027836499101000604> (1991).
2. Benitez, A. & Vallejo, D. *New Technique to Improve Probabilistic Roadmap Methods* in *Advances in Artificial Intelligence – IBERAMIA 2004* (eds Lemaître, C., Reyes, C. A. & González, J. A.) (Springer Berlin Heidelberg), 514–523. ISBN: 978-3-540-30498-2.
3. Kavraki, L. & Latombe, J. *Randomized preprocessing of configuration space for path planning: articulated robots* in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)* **3** (1994), 1764–1771 vol.3.
4. 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.
5. Holland, J. H. *Adaptation in Natural and Artificial Systems* (In University of Michigan Press, Ann Arbor., 1975).
6. Alouges, F., Chitour, Y. & Long, R. A Motion-Planning Algorithm for the Rolling-Body Problem. *IEEE Transactions on Robotics* **26**, 827–836. ISSN: 1552-3098 (2010).

A APPENDIX