Parametric comparison between A* and RRT motion planning algorithms

Ahmed El-Ghannam, Hazem Saad, Mohammed Khairy,Wessam Mohammed
Department of aerospace engineering
University of science and technology, Zewail city of science and technology
Egypt

***Abstract*— Parametric comparison will be done to evaluate the performance of two motion planning algorithms A* and RRT (rapidly-exploring random tree algorithm). Evaluation matrix will be constructed to capture the main advantages and disadvantages of both algorithms. ROS, robot operating system, with gazebo simulator will be used to simulate both algorithms in different maps to simulate the collision free paths obtained.**

Keywords—Motion planning;A*; RRT,sampling-based search.

I. INTRODUCTION

Motion planning, sometimes called path finding, has been one of the most important issues in robotics area, it is basically finding the optimal policy to guide a robot form initial state to a goal state without colliding with obstacles. There are two directions in literature that deal with motion planning problem, the first concentrates on dealing with the environment with no effort produced to deal with robot control imperfection, some examples include visibility graph and potential field methods [1]. The main goal of path finding algorithms is to provide geometric plan rather than assuring best velocities and acceleration [2]. There are different problems in the development of the optimal algorithm, ranging from uncertainty in sensors and actuators, to non-static environment that needs adaptive motion planning. Many developments have been applied to RRT to improve the efficiency of finding a collision-free path in a short time. A method used to bias the algorithm toward finding the goal has been proposed by [3]

II. BRIEF DESCRIPTION OF THE TWO ALGORITHMS

A. *A* algorithm*

A* search is a discrete motion planner algorithmic depends on finding the optimal path based on choosing a cost function f relates the heuristic distance between nodes and goal and the cost of moving from one node to another. The mathematical illustration goes as the following:

$$f(n) = g(n) + h(n)$$

Where f is the cost function, g is the cost of the path from starting node to node n and h is the heuristic between each node and the goal [3].

Choosing the lowest value of f guarantees the choice of optimal path according to this algorithm.

Pseudo code

1) Use a queue to store all the partially expanded paths.
2) Initialize the queue by adding to the queue a zero-length path from the root node to nowhere.

3. Repeat
Examine the first path on the queue.
If it reaches the goal node then success.
Else {continue search}
Remove the first path from the queue.
Expand the last node on this path by one step.
Calculate the cost of these new paths.
Add these new paths to the queue.

B. *RRT algorithm*

- RRT (rapidly-exploring search algorithm) is categorized under sampling-based path planning algorithms. This group of algorithms uses collision detection as the main tool to separate the motion planning from the geometric models by performing C-space sampling and discrete search [4]. It is a single-query algorithm that constructs a large number of collision free paths to choose the best possible path to the goal [5]. RRT algorithm shows to give exponential rate of decay for the probability of failure (Frazzoli et al., 2002). Random samples are generated and then connected to the nearest and a tree grows, path is generated, these random samples can be viewed to control the direction of tree growth. Random samples a (i) are connected to the nearest node q_n in a tree. Stopping criteria occurs when a connection is constructed between the random sample and q_{near} in a topological graph G(V,E) where V vertices and E edges [6].

Pseudo code [7]

RRT_SINGLE (q_0)

1) G.init(q_0).
2) **For** i=1 to k do

3) $q_n \leftarrow$ NEARST(S,a(i)).
4) $q_s \leftarrow$ STOPPING_CONFIGURATION $(q_n, a(i))$.
5) **If** $q_s \neq q_n$
6) G.Add_vertex(q_s).
7) G.Add_edge(q_n, qs).

III. COMPARISON RESULTS

Different maps will be used to get in the comparison. the evaluation parameters will be as the following:

1. Smoothness of the path (number of edges).
2. Computation time of each algorithm (Time complexity).
3. Total distance covered.
4. Storage volume (Space complexity).
5. Size of map

RRT algorithm has been ten run times and the average values have been taken because of the probabilistic nature of the algorithm.

Note: each search step counts for 16 bytes.

A. First comparison (normal maps)

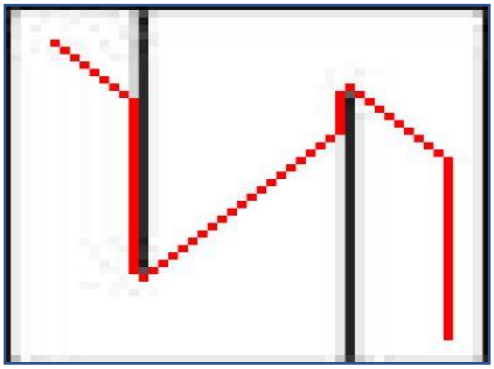


Fig 1. A* search algorithm result

Search steps	1312 times
Time spent	3.02 sec
Total distance	765 unit
Number of edges	6
Size of image in pixel	50X50

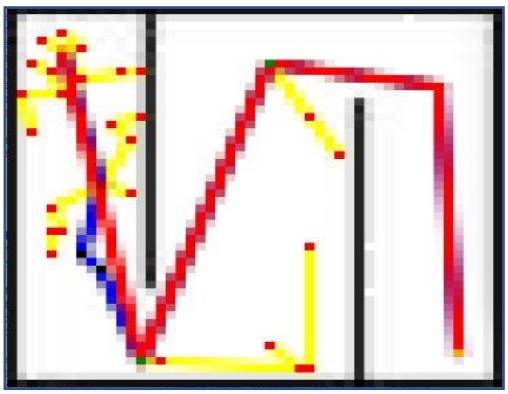


Fig 2. RRT search algorithm result

Search steps	41 times
Time spent	0.365 sec
Total distance	137 unit
Number of edges	5
Size of image in pixel	50X50

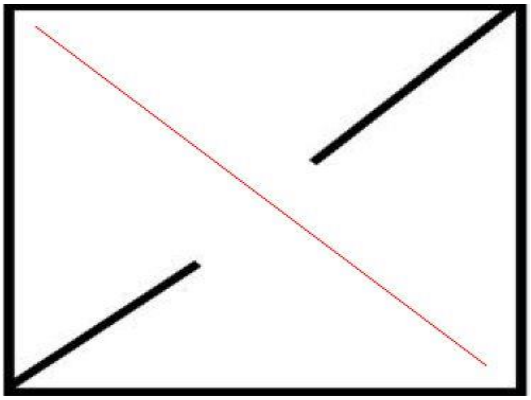


Fig 3. A* search algorithm result

Search steps	216 times
Time spent	0.218 sec
Total distance	302 unit
Number of edges	1
Size of image in pixel	250X250

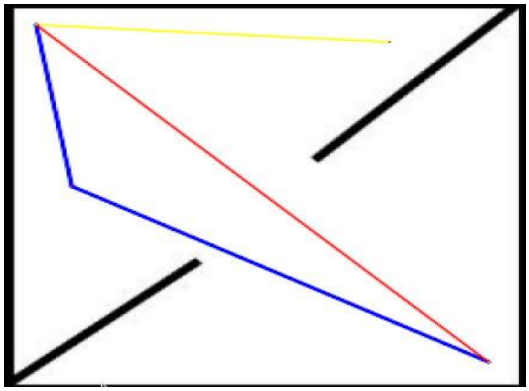


Fig 4. RRT search algorithm result

Search steps	9 times
Time spent	3.02 sec
Total distance	765 unit
Number of edges	6
Size of image in pixel	250X250

B. Second comparison (maze map)

1.

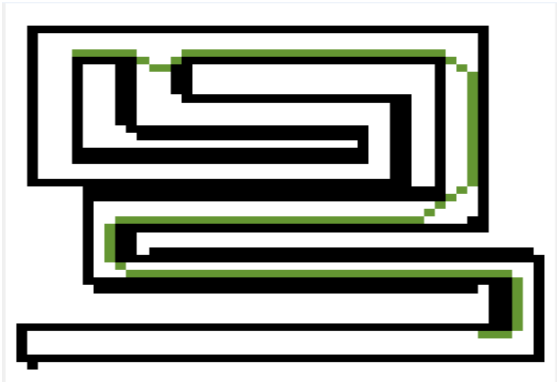


Fig 5. Collision free-path obtained for A* algorithm

Search steps	582 times
Time spent	0.068325 sec
Total distance	135 unit
Number of edges	18
Size of image in pixel	50X50
Simulation time	74 sec

*Search steps: number of searches done at each step

2.

RRT algorithm fails to find an optimal path in this map as the number of iterations exceeds the memory limit.

3.

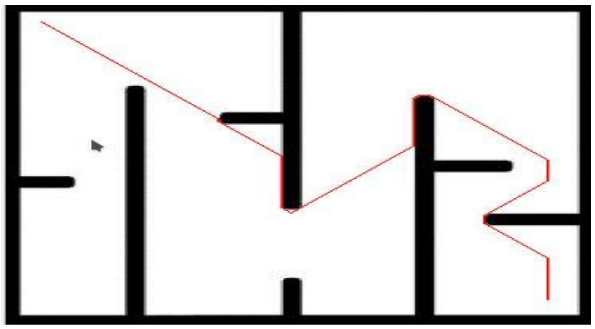


Fig 6. Collision free-path obtained for A* algorithm

Search steps	34859 times
Time spent	6.97 sec
Total distance	507 unit
Number of edges	16
Size of image in pixel	250X250

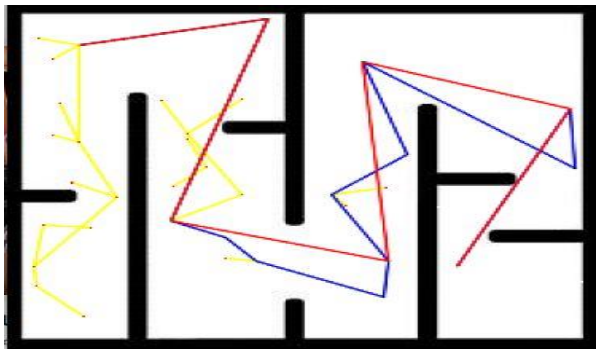


Fig 7. Collision free-path obtained for RRT algorithm

Search steps	59 times
Time spent	19.23 sec
Total distance	690 unit
Number of edges	7
Size of image in pixel	250X250

C. Third comparison (maze map)

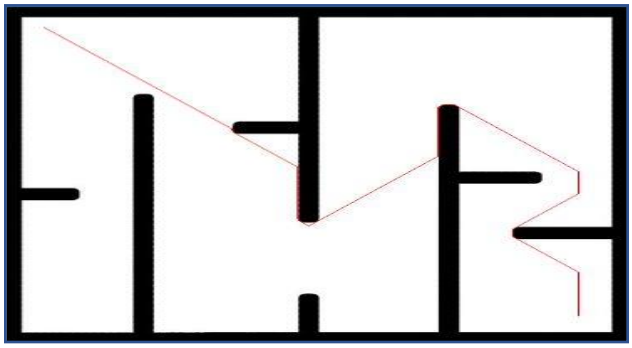


Fig 7. Collision free-path obtained for A* algorithm

Search steps	138083 times
Time spent	221.9 sec
Total distance	1013 unit
Number of edges	20
Size of image in pixel	800X800

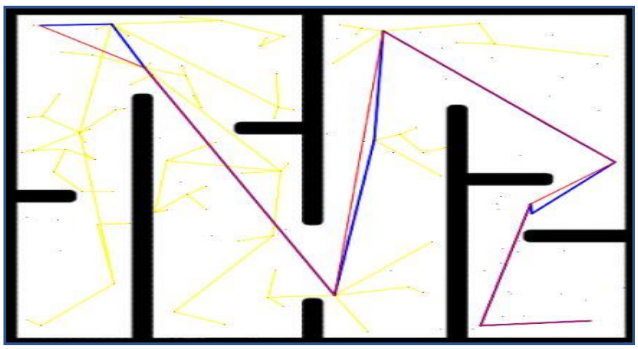


Fig 8. Collision free-path obtained for RRT algorithm

Search steps	128times
Time spent	118.1 sec
Total distance	1469 unit
Number of edges	8
Size of image in pixel	800X800

IV. SIMULATION IN GAZEBO

ROS, robot operating system, [7] has been used to implement a PID controller and connected to Gazebo simulator [8] to simulate the proposed motion planning algorithms. Two different simulation sequence will be shown here.

1. **A* search simulation**

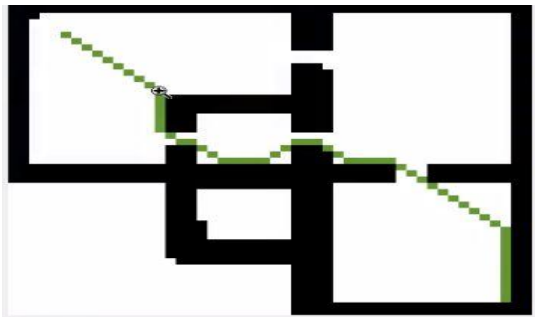
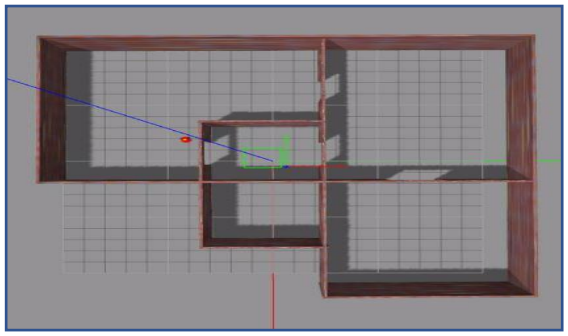


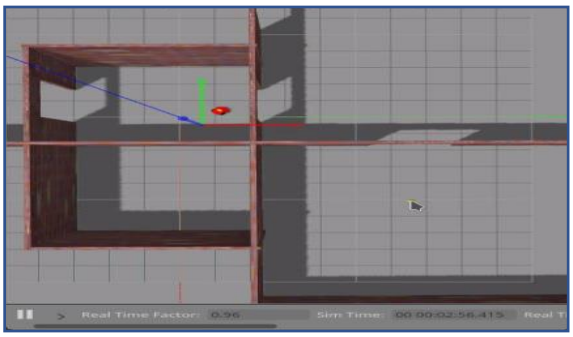
Fig 9. Collison free path for RRT A*



(1)

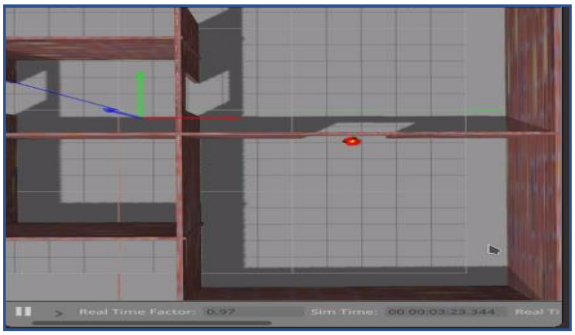
(2)

(3)

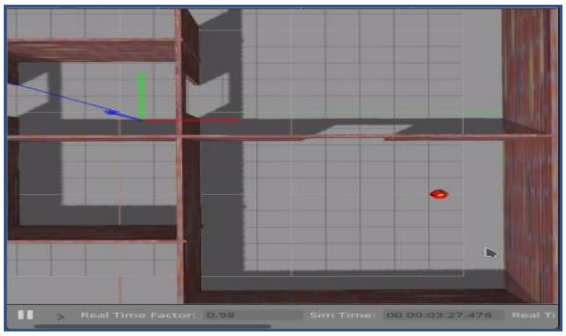


(4)

(5)



(6)



(7)

(8)

2. **RRT algorithm simulation**

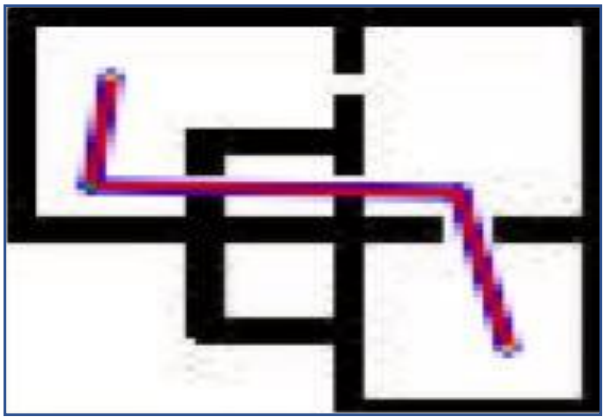
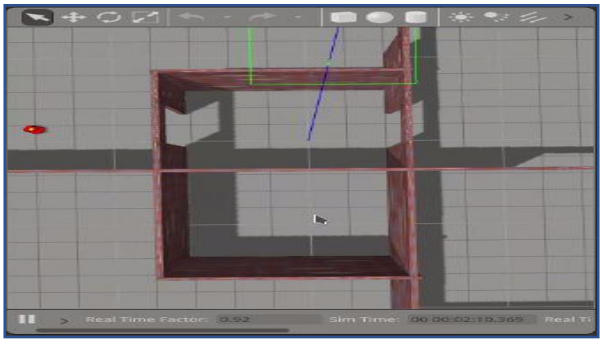
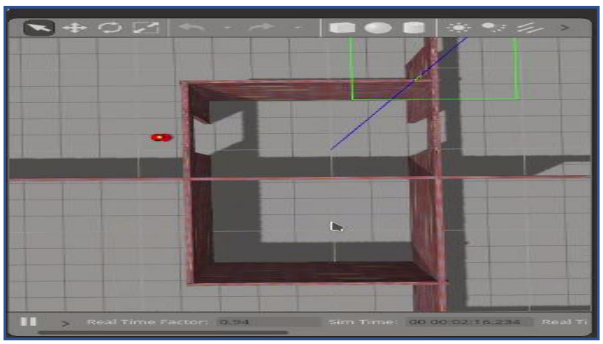


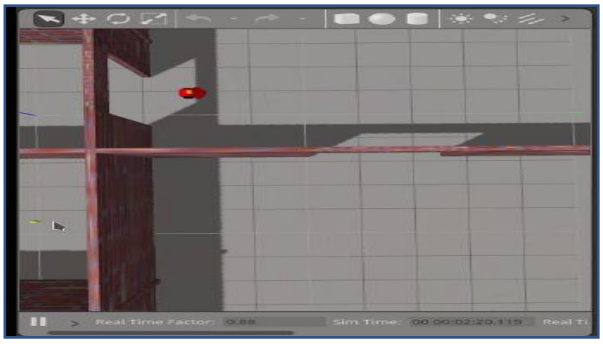
Fig 10. Collison free path for RRT



(1)

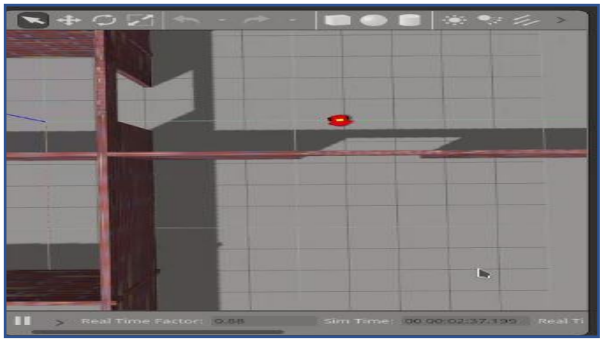


(2)

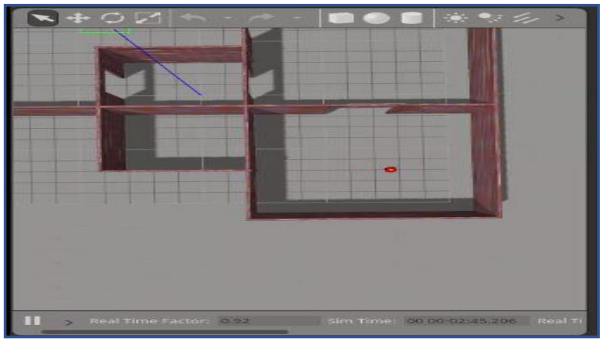


(3)

(4)



(5)



(6)

D. Map of real city

1.

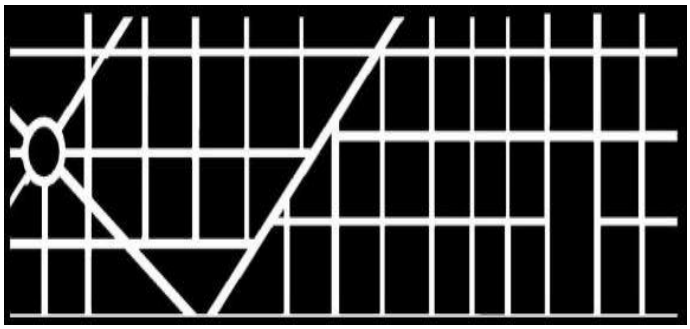


Fig 8. Map of portion of real city

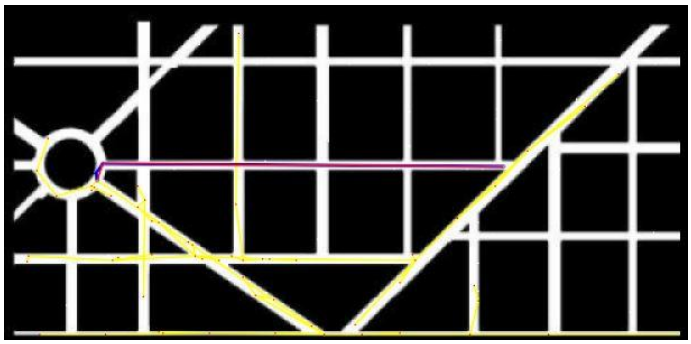


Fig 9. Collision free-path obtained for RRT algorithm

Search steps	424 times
Time spent	372 sec
Total distance	633 unit
Number of edges	6
Size of image in pixel	800X800

A* algorithm fails to find optimal path in this map due to the high storage volume

V. RESULTS INTERPRETATION

In this section, we present an overall of the performance comparison between RRT algorithm and A* Algorithm on two types of maps [9]:

- ***Open-space type maps***
- ***Maze-type maps***

From the above results, we can reach a rough conclusion

- A* is much more efficient (time wise) in case of maze type maps however RRT takes much time constructing the tree and sometimes fails to find solution
- RRT is to some extend better in open space maps while A* takes more time discretizing the whole domain.
- Regarding the map size, we've compared the same map with two different resolutions (500 X 500) & (50 X 50). the results showed:
  - In case of 50 X 50 maps, A* is much better in all cases whether maze type or open type.
  - In case of 500 X 500, A* is better in maze type maps but RRT is better in open space maps.

These results seem to be logical since in case of 50 X 50 maps, A* will not find a problem searching the whole domain. but this is not the case for a much larger maps

- In terms of path length and number of edges, A* has shorter path length than RRT in most of the cases.
- In terms of consistency, A* showed to give the same results always. But this is not the case for RRT as if we run 10 instances of RRT it results 10 different results that's why we represented it in the form of box blot.

50 X 50 maps

		Time	Path length	Number of edges	Consistency
Open-space Maps	A *	✔	✔	✔	✔
	RRT	✘	✘	✘	✘
Maze-Type maps	A *	✔	✔	✔	✔
	RRT	✘	✘	✘	✘

500 X 500 maps

		Time	Path length	Number of edges	Consistency
Open-space Maps	A *	✘	✔	✔	✔
	RRT	✔	✘	✘	✘
Maze-Type maps	A *	✔	✔	✔	✔
	RRT	✘	✘	✘	✘

VI. CONCLUSION

In this project, parametric comparison between two motion planning algorithms RRT and "A star", A*, has been conducted. Evaluation criteria has been used to distinguish both algorithms in different frameworks. The results have been verified by implementing actual controller in ROS on a mobile robot in Gazebo simulation environment. The results are promising and matched to what has been proposed in literature. Both algorithms have pros and cons depending on the map type and discretization criteria.

REFERENCES

[1] C. Belta, V. Isler and G. Pappas, "Discrete abstractions for robot motion planning and control in polygonal environments", *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 864-874, 2005.

[2] K. Tang SH and K. Zulkifli N, "A Review on Motion Planning and Obstacle Avoidance Approaches in Dynamic Environments", *Advances in Robotics & Automation*, vol. 04, no. 02, 2015. I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[3] Yingying K., Xiong C., Jianda H. (2012) Bidirectional Variable Probability RRT Algorithm for Robotic Path Planning . In: Wang X., Wang F., Zhong S. (eds) Electrical, Information Engineering and Mechatronics 2011. Lecture Notes in Electrical Engineering, vol 138. Springer, London

[4] D. Yershov and S. LaValle, "Simplicial Dijkstra and A∗Algorithms: From Graphs to Continuous Spaces", *Advanced Robotics*, vol. 26, no. 17, pp. 2065-2085, 2012. R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[6] L. Knispel, R. Matousek, "A Performance Comparison of Rapidly-exploring Random Tree and Dijkstra's Algorithm for Holonomic Robot Path Planning" Recent Advances in Applied and Theoretical Mathematics.

[7] "Documentation - ROS Wiki", *Wiki.ros.org*, 2017. [Online]. Available: http://wiki.ros.org. [Accessed: 09- Jan- 2017].

[8] "Gazebo", *Gazebosim.org*, 2017. [Online]. Available: http://gazebosim.org. [Accessed: 09- Jan- 2017].

[9] L. Knispel, R. Matousek, "A Performance Comparison of Rapidly-exploring Random Tree and Dijkstra's Algorithm for Holonomic Robot Path Planning" Recent Advances in Applied and Theoretical Mathematics.