Toward More Efficient Methods for Path Planning of Mobile Robots: Simplified Non-Convex Constraints

Mohammad Biglarbegian

Abstract— **In this paper the path planning of robots is formulated in the context of an optimization framework. It is shown that generating motion trajectories for mobile robots in real environment is cast into a non-convex problem. Global optimization techniques in the non-convex framework are thus exploited to obtain the desired trajectories. The approach presented in this paper can easily take the robot speed limitations into consideration and hence is more effective than other well-known existing techniques. Simulation results are presented to demonstrate how robot trajectories are obtained. It is found that using the proposed methodology herein results in more realistic solutions for mobile robots path planning. Trajectory generation using optimization with non-convex constraints is an important move forward toward efficient path planning algorithms. Thus, results of this paper are of great significant especially in real experiments.**

I. Introduction

Path planning is an important aspect in mobile robot research. Sampling based techniques such as probabilistic road map (RPM) and Rapidly-exploring Random Trees (RRT) have gained popularity amongst roboticists because of their efficiency in certain applications [1]. Amongst the recently developed sampling-based algorithms, RRT is the most well-known. This algorithm expands the search in all directions and as such its efficiency is also limited because distribution of the points is entirely random without any clear objective towards the goal. Moreover, for complicated kinodynamic constraints the RRT algorithm becomes inefficient [3]. They also suffer from generating non-smooth path [5] and there are attempts to improve this method [4]. Other popular techniques such as potential field, although very efficient, suffer from the local minima problem. In summary, the existing probabilistic approaches require the intelligence and vision required to clear a feasible solution in a timely manner. Implementation of those approaches in dynamic environments is another concern that should also be considered. Alternative methods for path planning are optimization techniques that formulate the trajectory generation as an optimization problem. Doing so will enable taking into account non-holonomic robot constraints. Control techniques such as optimal control and model predictive control (MPC) have been used to generate trajectories for UAVs and robots. Particularly, MPC has recently become an attractive tool for planning and navigation [8]. In general, obstacles are non-convex. Hence, path planning involves with non- convex constraints. However, nonlinear MPC strategies induce non-convexity in their structures and thus are known to suffer from the drawback of the local minima problem [9]-[10]. Most recently, there have been attempts to tackle this issue using other complementary methods such as sampling based MPC [13]. However, achieving a globally optimal solution depends on the sample points. Thus, the computational time required using cluttered environments is expected to be significant. Due to this limitation, it remains an open question if these strategies can be efficient for such environments; especially for systems requiring fast response. Recently, research have been dedicated on the development of efficient algorithms for the deployment of multiple robots. In [12] a solution for Voronoi coverage of multiple robots was presented for environments with non-convex polygons. The proposed solution combines classical as well as path planning algorithms for the motion of the robots. The methodology has been developed for static environments and in the experiments only simple obstacles (triangles) have been used. Most recently, Schwager et al. [11] unified geometric, probabilistic, and potential field approaches for multiple robots and showed that these strategies are related to a same cost function. Accordingly, controllers were designed to converge to the local minimum of the proposed function. The closing conclusion of their paper is that the multi robot deployment in fact requires non-convex optimization. In the literature, researchers have attempted optimal control and model predictive control (MPC) to generate trajectories for UAVs and robots. However, trajectory generation, as will be shown in this section, is a non-convex problem and to find global solutions, it must be solved using global optimization. This will also answers the local minima problem. Recently, there has been interest in using global optimization to tackle the path planning of mobile robots [6]. However, the existing approaches do not consider non-holonomic robot or non-convex environmental constraints. The benefit of finding the trajectories using the proposed framework allows one to formulate other similar problems (with even different and difficult constraints); hence, providing a more general theme for trajectory generation.

The organization of this paper is as follows: Section II reviews the most well-known path planning algorithms. Sections III and IV present our general approach for path planning using optimization, Section V provides simulations, and Section VII concludes the paper.

II. Path Planning: review

Path planning for robot manipulators and mobile robots is one of the key concepts in robotics research. Path planning is defined as finding trajectories that guides the robot from

Mohammad Biglarbegian is with the School of Engineering, University of Guelph, Ontario, Canada mbiglarb@uoguelph.ca

starting point to the goal while avoiding obstacles, see Figure 1. In the literature, numerous algorithms for path planning have been proposed. This section reviews the most popular algorithms used in robotics.

A. Potential Fields

Potential fields approach is probably the most widely adopted path planning algorithm and it was first proposed by Khatib [14] because of their simplicity have been extensively applied; however, this approach suffers from the drawbacks of local minima. Moreover, to effectively navigate a robot to a goal, the parameters of the potential functions need to be tuned which is not a straightforward task.

B. Probabilistic Road Map (PRM)

Another well-known approach is probabilistic road map (PRM) that has gained significant popularity amongst robotists. Although its simplicity, this approach is plagued by a significant amount of time that is required for collision avoidance checking; hence, making it computationally very expensive. Furthermore, the PRM method is ineffective for some environments such as constrained surfaces. Over the years, various complementary algorithms have been developed to address the limitations of this algorithm. Nevertheless, due to the sampling-based nature of this algorithm which almost blindly searches for a feasible trajectory, it lacks the intelligence that is necessary for trajectory generation. As a result, the applicability of this approach especially in real-time as well as in multi-robots is very limited.

C. Rapidly-exploring Random Trees (RRT)

Most recently, another sampling-based technique, Rapidly-exploring Random Trees (RRT) algorithm, has been developed and shown to be efficient for certain applications. This algorithm randomly samples points from a workspace and creates motion trajectories in various directions that ultimately results in a feasible solution. While this algorithm, unlike PRM, expands the search in all directions, its efficiency is also limited because distribution of the points is entirely random without any clear objective towards the goal. Also, for difficult kinodynamic constraints the applicability of this approach is limited [3].

III. Path Planning using optimization

Our approach in the design of trajectory is based on the optimization of a desired function, which is called objective, i.e., energy, time to travel, etc. In this research, we focus on minimizing the energy. We also assume that, without loss of generality, objects/obstacles are circumscribed by circles. Thus, we can formulate the trajectory generation in the context of constrained optimization problem in which obstacle avoidance our constraints.

Suppose the trajectories of the robot at a time instant, t are designated as $x(t)$ and $y(t)$. The control input for the robot is also given by $\boldsymbol{u} = [u(t), v(t)]$, where $u(t)$ and $v(t)$ are control inputs for the left and right wheels of the robot, respectively. Then the objective is expressed as

$$f_{obj} = u^2(t) + v^2(t) \tag{1}$$

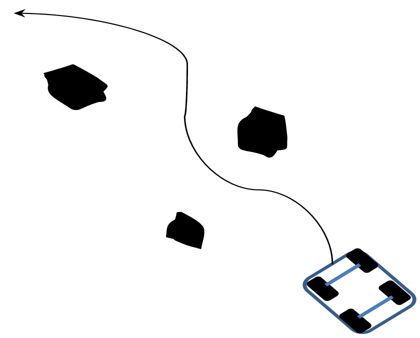

Fig. 1. Mobile robot path planning.

To formulate the constraints, assume obstacle 'i' is circumscribed by a circle given by

$$(x - x_i)^2 + (y - y_i)^2 = R_i^2 \tag{2}$$

where x_i, y_i are the coordinates and R_i is the radius of the circle. Therefore, to avoid this obstacle, the trajectory must satisfy

$$(x - x_i)^2 + (y - y_i)^2 > R_i^2 \tag{3}$$

Now the optimization problem can be formulated as

$$\begin{cases} \text{Minimize} \qquad f_{obj} = u^2(t) + v^2(t) \\ \\ \text{Subject to} \\ \qquad \dot{x} = u \\ \qquad \dot{y} = v \\ \qquad \forall i : (x - x_i)^2 + (y - y_i)^2 > R_i^2 \\ \qquad u_{\min} \leq u \leq u_{max} \\ \qquad v_{\min} \leq v \leq v_{max} \end{cases} \tag{4}$$

where $u_{\min,\max}$ and $v_{\min,\max}$ are the bounds of the control inputs. The first two constraints are related to the dynamics and the third constraints, as discussed, is due to the obstacles.

It is useful to review the definition of a convex function.

Definition 1: [15] Given $0 \leq \theta \leq 1$, a function f is convex if for $\forall x, y \in f$,

$$f(\theta x + (1 - \theta)y) = \theta f(x) + (1 - \theta)f(y) \tag{5}$$

The above definition is useful when path planning is formulated as an optimization problem.

If the robots are non-holonomic, then extra constraints must be added. Hence, the optimization problem can be expressed as

$$\begin{cases} \text{Minimize} \qquad f_{obj} = u^2(t) + v^2(t) \\ \\ \text{Subject to} \\ \qquad \dot{x} = u \\ \qquad \dot{y} = v \\ \qquad \forall i : (x - x_i)^2 + (y - y_i)^2 > R_i^2 \\ \qquad u_{\min} \leq u \leq u_{max} \\ \qquad v_{\min} \leq v \leq v_{max} \\ \qquad \dot{x}cos\theta - \dot{y}sin\theta = 0 \end{cases} \tag{6}$$

where θ is the angular position of the robot. The last constraint appears because of the non-holonomic constraints.

If an optimization problem is convex, it is guaranteed to achieve global solutions. There exists several numerical techniques for solving convex optimization problems. It is easy to see that (4) is a non-convex optimization problem, due to the inequality required to avoid obstacles, i.e., (3). In real environments, objects are non-convex, and consequently convex optimization techniques cannot be exploited. Therefore, this problem must be solved as a global nonlinear optimization problem. Global optimization techniques such as simulated annealing, branch and bound, or genetic algorithm can be also readily adopted. Matlab nonlinear optimization toolbox was used for solving this problem.

IV. Non-circular objects

This section discusses the case where the objects are not circular.

A. Ellipse

For an ellipse, the constraint is expressed as

$$\frac{(x-x_i)^2}{a^2}+\frac{(y-y_i)^2}{b^2}>1 \tag{7}$$

B. Circumscribed circle

If the objects can be circumscribed with a circle, circumscribed-circle, then the above methodology can be easily adopted for those cases as well; the radius of each circumscribed-circle is replaced with R_i in (6).

For triangles, there exists an analytical expression for the circumcircle. The circumcircle equation for a triangle with vertices x_i, y_i $(i = 1, 2)$ is given by [16]

$$\begin{vmatrix} x^2+y^2 & x & y & 1 \\ x_1^2+y_1^2 & x_1 & y_1 & 1 \\ x_2^2+y_2^2 & x_2 & y_2 & 1 \\ x_3^2+y_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0 \tag{8}$$

Using the circumcircle as constraint, even though will result in a desired solution it excludes the regions between the exterior of a triangle and its circumscribed. To include those regions, one may consider adding them as extra constraints in formulating the optimization problem. To include the

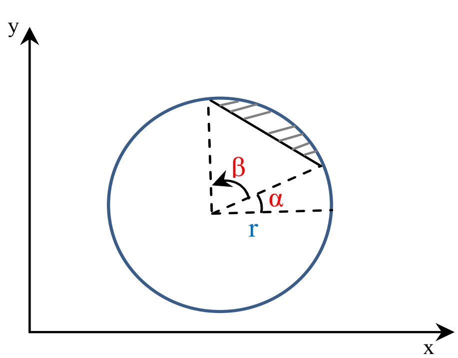

Fig. 2. A triangular obstacle with its circumscribed circles.

segments (shown with the hash in Fig. 2), the following additional constraints should be considered:

$$\begin{aligned} (x-x_0)^2+(y-y_0)^2 &\le r_{cc} \\ x_0 - r\cos(\alpha+\beta) \le x &\le x_0 + r\cos\alpha \\ y_0 + r\sin\alpha \le y &\le y_0 + r\sin(\alpha+\beta) \end{aligned} \tag{9}$$

where r_{cc} is the radius of the circumcircle. The first constraint given in (9) is convex and hence easier to solve. If the

For a regular polygon with side a, it is known that the radius of a circumscribed circle is given by

$$R = \frac{1}{2}\csc\left(\frac{180}{n}\right) \tag{10}$$

where n is the side number of the polygon. Therefore, it is straightforward to solve the problem for this case as well.

C. Polygons

For a general concave polygon the circumscribed circle may not always exist. However, one simple approach is to divide the polygon into several triangles or regular polygon and consider the corresponding circumcircles. This approach

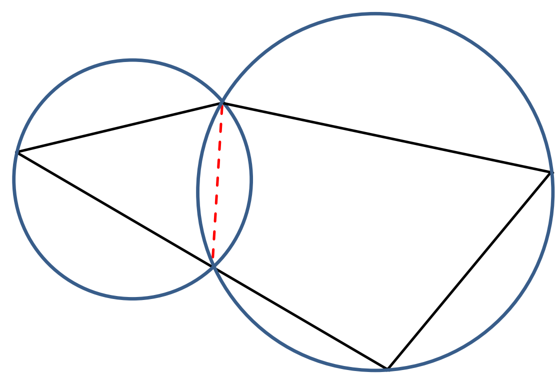

Fig. 3. A concave polygon decomposed into triangle/polygon and their corresponding circumscribed circles.

excludes the segments of the circumscribed circles that can be viable points of a robot path. However, the simplicity of this approach makes it very attractive and applicable to concave polygons with complex geometries.

V. Simulations

This section presents the simulation results of path planning for several objects. Two sets of simulations are performed and results are reported. The first set of simulations present the path planning results in the absence of velocity constraints. The second part presents the path planning results considering velocity as constraints.

A. Path planning without velocity constraints

This part illustrates the simulation results of path planning for several objects. We begin with a single circular object. Simulations are run in two-dimensional environments. Given the robot starting point at $x_0 = 0, y_0 = 0$, the optimization algorithm should find the robot path to navigate it to the goal while avoiding a circular object located at$x = 0.7, y = 0.2$ with a radius of 0.13. Figure 4 demonstrates the output path of the algorithm which effectively shows the strength of this approach.

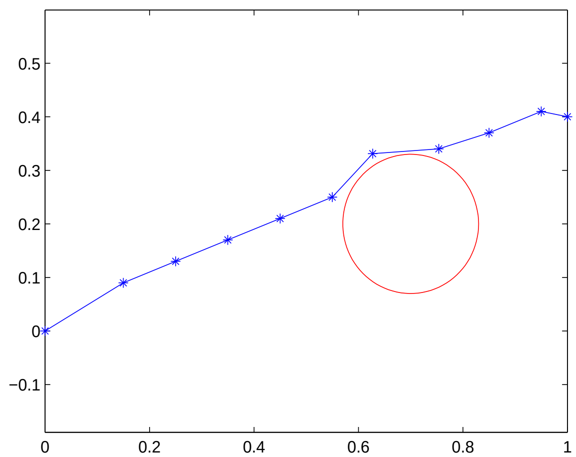

Fig. 4. One circular object.

The next experiment is to have two circular objects in the environment. Figure 5 shows the generated path for this case and demonstrate the effectiveness of this approach.

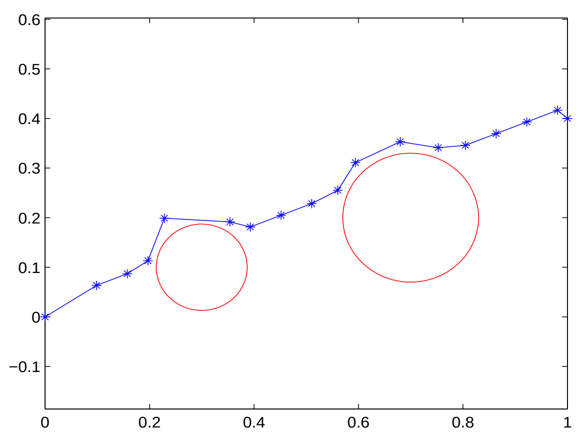

Fig. 5. Two circular objects.

When objects are polygons, based on the proposed method, it is easy to find the path. To illustrate this, an environment with three non-regular polygons are considered. Figure 6 shows the path generated for the robot which can effectively navigate the robot by solving the optimization algorithm.

B. Path planning with velocity constraints

The second part is dedicated to simulation results when the velocity of the robot is constrained. The $\max$ and $\min$ velocities for the robot in x and y directions are as

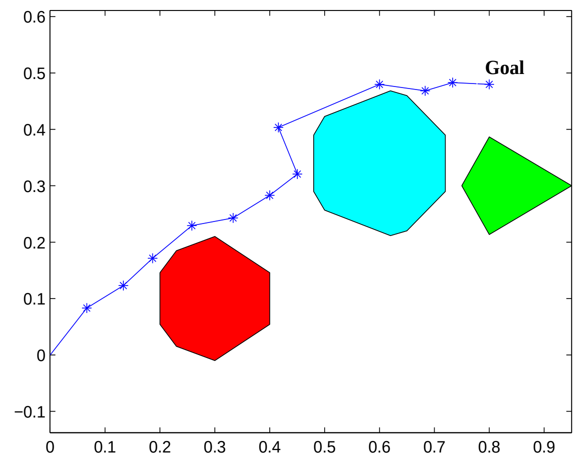

Fig. 6. Polygon objects.

follows: $Vx_{\min} = 0.07m/s, Vx_{\max} = 0.21m/s, Vy_{\min} = -0.01m/s, Vy_{\max} = 1m/s$. Figure 7 gives the generated path for the robot considering the above constraints.

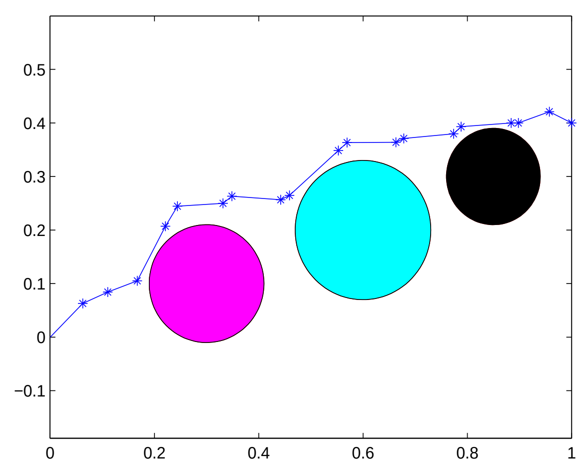

Fig. 7. Path planning with velocity constraints.

Figure 8 compares the generated paths without and with velocity constraints. As can be seen from the graph, the velocity bounds impact the path generated by the algorithm. In real-time experimentation for certain environments, the robot speed are needed to be limited. Consequently, generating path for a robot requires accounting for the speed limitations. Therefore, the results of this section particularly is more realistic and of great use for practical implementations. This also showcases the potential of this approach, because it can render path planning by incorporating the bounds of the robot's speed. Other path planning methods such as potential field, RPM, and RRT all lack addressing velocity constraints in generating a path, unless those original algorithms are modified.

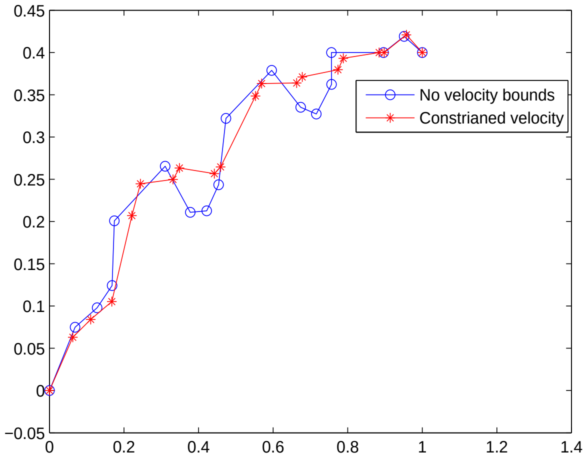

Fig. 8. Path planning with/without velocity constraints.

VI. Discussion

To further efficiently generate motion trajectories, it is desirable to reduce the computational time required to run the proposed algorithm. A very effective method is to partition the working space into several sections and run the algorithm in parallel. Doing so will reduce the computational complexity of the problem significantly. Furthermore, it will reduce the time needed to generate trajectories which is crucial for dynamic environments. To accomplish this 'local' start-goal are define in each subdivision and the path planning algorithm for each domain is executed. The start and goal of the subdivisions must be connected to each other to render a feasible solution at the end. Figure 9 illustrates one such partition and reflects the trajectories produces in each domain.

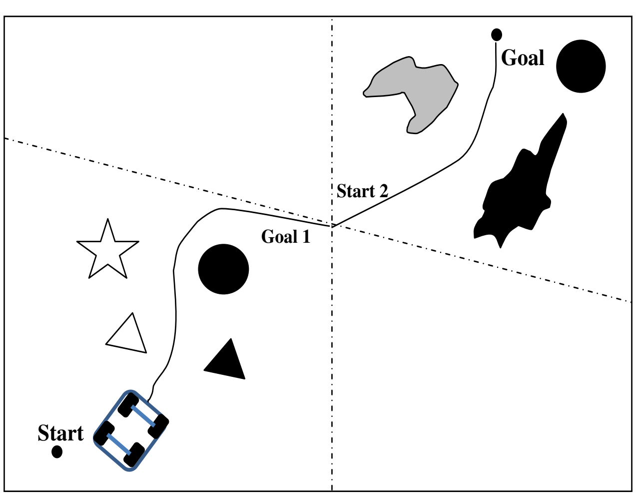

Fig. 9. Path planning with/without velocity constraints.

VII. Conclusions

In this paper, the path planning of non-holonomic mobile robots for non-convex obstacles are considered. The path planning was formulated in the form of a constrained optimization problem, which as a result must be solved in the context of global optimization. Simulations for some simple non-convex obstacles demonstrated that the proposed strategy is a viable solution for path planning with several non-convex constraints that could also include robot velocity bounds (if exist). The proposed method also does not suffer from the drawback of local minima. As future work, the proposed technique will be extended to more complex geometries. As well, different global optimization techniques will be used to compare their computational complexity.

References

[1] D. Dolgov and S. Thrun and M. Mantemerlo and J. Diebel, Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments, *Int. J. Robts. Resrch.*, vol. 29, no. 5, 2004, pp.485-501.

[2] S. Karaman S. and E. Frazzoli, Sampling-based algorithms for optimal motion planning, *Int. J. Robts. Resch.*, vol. 30, no. 7, pp. 485-501, 2011.

[3] A. Shkolnik A. and M. Walter and R. Tedrake, Reachability-Guided Sampling for Planning Under Differential Constraints, *IEEE Int. Conf. Robts. & Auto.*, Kobe Japan, May 2009, pp. 2859-2865.

[4] Roadmap-Based Path Planning, *IEEE Mag. Robts. & Auto.*, vol. 15, pp. 58-66, 2008.

[5] Recent Development in Motion Planning, *Lect. notes. Comp. Science, Springer-Verlag*, vol. 2331, no.1 , pp. 3-13, 2002.

[6] Y-Q. Miao and A. Khamis and F. Karray and M. Kamel, Global Optimal Path Planning for Mobile Robots Based on Hybrid Approach with High Diversity and Memorization, *Springer lect. notes. AIS.*, 2011.

[7] J-W. Lee and D-H. Leey and J-J. Lee, Global Path Planning Using Improved Ant Colony Optimization Algorithm through Bilateral Cooperative Exploration, *Proc. IEEE DEST*, Daejeon, Korea, May-June 2011.

[8] T. M. Howard and C. J. Green and A. Kelly, Receding Horizon Model-Predictive Control for Mobile Robot Navigation of Intricate Paths, *Springer-Verlag: J. Field and Service Robts.*, pp. 1396-1401, 2010.

[9] R. V. Cowlagi and P. Tsiotras, Hierarchical Motion Planning with Kinodynamic Feasibility Guarantees: Local Trajectory Planning via Model Predictive Control, *Proc. IEEE Conf. Dec. & Ctrl.*, pp. 1396-1401, 2010.

[10] N. Kantas and J. M. Maciejowski and A. Lecchini-Visintini, Sequential Monte Carlo for Model Predictive Control, *Lec. Notes. Ctrl and Inf. Sci.*, 2009.

[11] D. Rus D. and J. J. Slotine, Unifying Geometric, Probabilistic, and Potential Field Approaches to Multi-Robot Deployment, . *Int. J. Robts. Resch.*, vol. 30, no. 3, pp. 371-383, 2011.

[12] A. Breitenmoser and M. Schwager and J-C. Metzger and R. Siegwart R. and D. Rus, Voronoi Coverage of Non-Convex Environments with a Group of Networked Robots, *Int. J. Robts. Resch.*, IEEE ICRA, Alaska, May-June 2010.

[13] C. V. Caldwell and D. D. Dunlap and Jr. E. G. Collins, Motion Planning for an Autonomous Underwater Vehicle via Sampling Based Model Predictive Control, *IEEE OCEANS*, Seattle, Sep. 2010.

[14] O. Khatib, Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *Int. J. Robts. Res.*, vol. 5, no. 1, pp. 90-98, 1989.

[15] S. Boyd and L. Vanderberghe, Convex Optimization, *Cambridge Univ Press*, 2004.

[16] D. Pedoe, Circles: A Mathematical View, *Washington, DC: Math. Assoc. Amer.*, 1995.