American University of Beirut

Maroun Semaan Faculty of Engineering and Architecture

Mechanical Engineering Department

MECH 691: Convex Optimization

Project Proposal

Fall 2020-2021

Convex Optimization for Wheeled Autonomous Mobile Robots Motion Planning Applications

Theodor Chakhachiro

Table of Contents

List of Figures

1 INTRODUCTION AND MOTIVATION

Wheeled Autonomous Mobile Robots are currently the recipient of most attention in the major robotics developmental research institutes due to their wide range of applications in today's cunning edge technology. These applications include but are not limited to space exploration, underwater navigation, search and rescue missions, intelligent transportation systems, robot racing competitions as well as agriculture and forestry robots. More common applications are industrial, manufacturing and construction robotics, but all these examples rely on a solid motion planning algorithm to ensure the wheeled robot achieves the given task without collision. Motion planning includes both trajectory planning and velocity control; this needs to be done simultaneously during the task without pre-planning as most of the applications involve an unknown environment which was not previously explored. Thus one can begin to realize the potential application of convex optimization, due to the availability of fast state of the art optimization algorithms, in order to optimize the trade-off between these two objectives, that is, if the problem can be convexified.

In this proposal, we will begin by conducting a review of previous research in the field, then we will formulate the proposed problem. Furthermore, we will list the expected deliverables as well as the potential challenges we might face. Finally, we will discuss the possible extensions of this work.

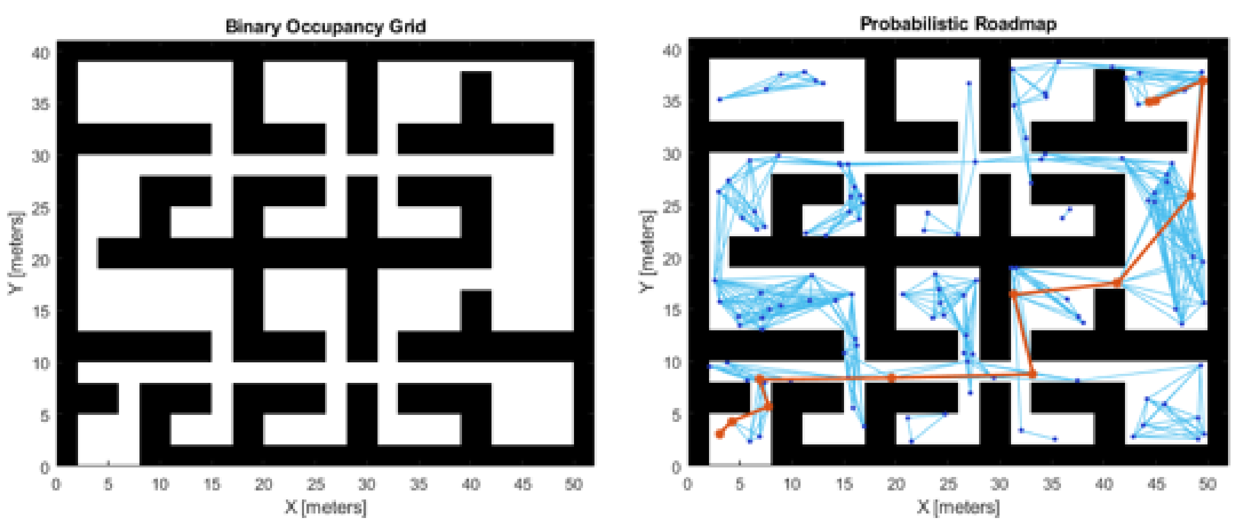

Figure 1: Occupancy grid map trajectory planning [1]

2 LITERATURE REVIEW

Motion planning has been a constant center of attention due to the importance it carries. A wide range of approaches successfully tackled the problem over the years, providing several solutions, each with advantages and disadvantages. Early preliminary work in this field used a stochastic model to determine an appropriate trajectory for robots with multiple degrees of freedom [3].J. Barraquand et al. developed a Monte-Carlo algorithm to generate a collision free trajectory for the robot using a joint distribution between gradient motion and discretized Brownian motions. Novel approaches apply neural network-based planning due to the computational complexity of the motion planning problem[4]. A. H. Qureshi et al. showed that their MPnet algorithm is more efficient than state of the art technology such as BIT (Batch Informed Trees) and Informed-RRT (Rapidly-exploring Random Trees). Vision based path planning is also heavily studied due to the vast availability of camera equipped mobile robots [5]. B. Hummel et al. presented a new algorithm for real time vision based planning under a holistic probabilistic framework, relying on different sensors to map the uncovered environment. However, this method has its drawbacks due to the limitation of LiDAR and sonar sensing on reflective surfaces. Subsequently, new path planning algorithms are utilizing convex optimization to solve these problems. Mainly, in [6], a heuristic algorithm named Convex Elastic Smoothing was developed to optimize the speed of dynamic vehicles as well as to smooth out the initially generated trajectory. This novel algorithm is robust against unusual trajectories, a key problem for other precedent state of the art algorithms, and can be used in real time due to the very small computational time for solution generation. Moreover, [7] proposed a new method to solve the motion planning problem, by formulating the problem as a time and energy optimization to generate an appropriate trajectory for nonholonomic wheeled mobile robot. Since the optimization problem in [7] is nonlinear, a change of variable is introduced to transform it into a discrete second order cone programming problem. In [8], a trajectory planning algorithm with optimal velocity control was presented by formulating the lane changing problem faced by road vehicles as a convex optimization problem by making some assumptions about the vehicle dynamics. This method resulted in higher speeds compared to its predecessor. Also, [2] developed a collision free algorithm that takes into account dynamic and static environment objects for motion planning, relying on the computational efficiency of the pseudo-code to utilize the maximum speed of the wheeled robot in order to reduce the path length as well as the task completion time. Finally, [9] proposed a novel approach to the motion planing problem, by combining convex optimization with neural networks, thus developing a convex–nonconvex constrained quadratic programming (CCNC-QP)-based dual neural network (DNN) (CNCC-QP-DNN). This work proves its superiority by achieving real-time control and incredible accuracy.

3 PROBLEM STATEMENT AND FORMULATION

3.1 Chosen Paper

Due to the clarity in the problem formulation as well as the availability of problem parameter, we have chosen to replicate the work done in [2]. In the following sections, we will define the problem by modeling and studying the dynamics of a wheeled mobile robot, then we will convexify the problem and address the objective function as well as the constraints it is subject to.

3.2 Problem Statement

Given the starting and ending points of the mobile robot as well as the coordinates and radius of the circular obstacles, the convex optimization algorithm will generate an optimal trajectory to suit the main objective introduced. This includes minimal time of completion, minimizing the total energy used and minimizing the total length of the planned trajectory. However, in the case of dynamic obstacles, we take as input a function defining the trajectory of the obstacles and generate a resulting output path plan.

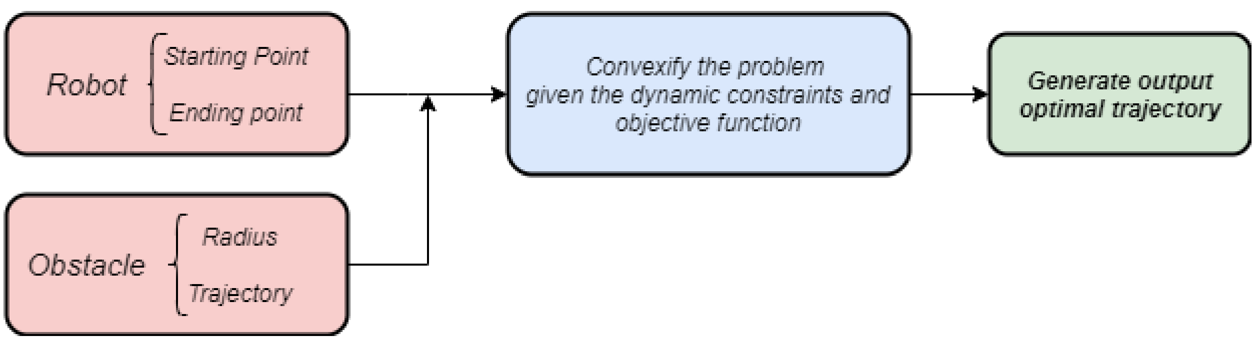

Figure 2: Problem Flowchart

3.3 Problem Formulation

In this section, we will define the dynamic model of a two-wheels mobile robot, then we will formulate the convex optimization problem.

3.3.1 Dynamic Model

In [2], the robot used is an E-puck, which is a mobile robot with two wheels, thus the problem can be scaled to other robots with similar dynamics, taking into account their maximum velocity and specifications.

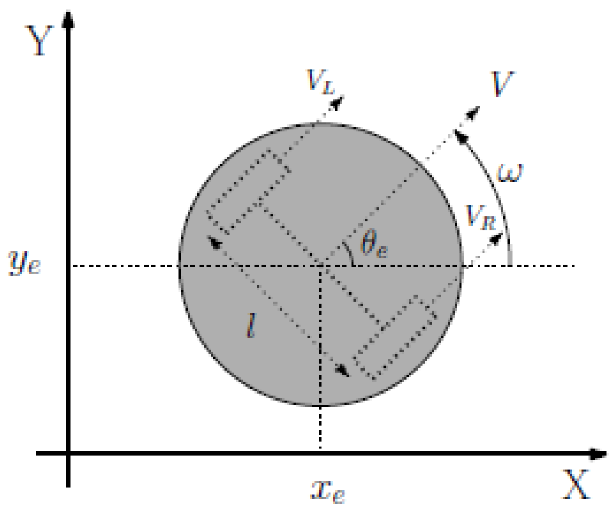

Figure 3: Global and Robot Frames

From figure 3, the coordinates of the robot in the global frame can be expressed as:

$$X = [x_e, y_e, \theta_e]^T \tag{1}$$

Taking the derivative with respect to time:

$$\dot{X} = [\dot{x}_e, \dot{y}_e, \dot{\theta}_e]^T \tag{2}$$

Given the right and left wheel velocities v_R and v_L as well as the distance between the wheels l, we can model the kinematics of the mobile robot:

$$\dot{X}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \frac{(v_R+v_L)\cos\theta_e}{2} \\ \frac{(v_R+v_L)\sin\theta_e}{2} \\ \frac{(v_R-v_L)}{l} \end{bmatrix} \tag{3}$$

3.3.2 Convexification

(a) **Objective Function**

As stated earlier, the objective function depends on the task to be carried out, mainly it could be either or a combination of minimizing the energy consumption, minimizing the trajectory length and minimizing the time taken to complete the task. Thus an appropriate objective function i a weighted sum of these tasks. In discrete programming, the trajectory can be discretized into k time steps, thus, minimizing the overall path length in done by minimizing the sum of the distance between each consecutive location of the robot in the global frame. Since we are working in 2D, our set of possible coordinates is $\mathbb{R}^2$, and the first part of our objective function is minimizing the sum of the $\ell - 2$ norm of the difference between each two consecutive robot positions:

$$\min \sum_{k=1}^{h} ||z_e(k+1) - z_e(k)||_2^2 \tag{4}$$

Where $z_e(k) \in \mathbb{R}^2$ is the 2D coordinate of the robot at time step k and h is the control horizon i.e. the number of manipulated variable moves to be optimized at control interval k in the Model Predictive Control design [1].

In the same manner, to minimize the total time taken $T = h * \Delta t$, where Δt is the period, we minimize the sum of the time taken at each step k, labeled as $dt(k)$:

$$\min \sum_{k=1}^{h} dt(k) \tag{5}$$

Finally, since we are using an MPC model, a third component in the objective needs to be added to make sure the final point in the trajectory $z_e(h)$ is close to the ending goal z_g given as input, this is done by minimizing the distance between the two vectors which is the $\ell - 2$ norm in this case:

$$\min ||z_e(h) - z_g||_2^2 \tag{6}$$

(b) **Constraints**

One major part of motion planning is collision avoidance, thus, to ensure the problem is convex, we need to approximate each obstacle, whether convex or non-convex, as a polygon which is the intersection of half-spaces which in terms, is convex. Let set C denote the set of all points ξ that lie inside the polygon, defined as:

$$C = \{\xi | A\xi < b\} \tag{7}$$

Where $A \in \mathbb{R}^{s \times k}$ and $b \in \mathbb{R}^s$. Thus a point ξ lies outside the polygon if one of the Linear Matrix Inequalities is not satisfied, formulated as:

$$A\xi \geq b + (v - 1)M \tag{8}$$

$$\sum_{i=1}^{s} v_i \geq 1 \tag{9}$$

Where M is a constant and $v \in \mathbb{R}^s$ is a binary vector such that $v_i \in \{0, 1\}$. This additional constraint will ensure that at least one point ξ will lie outside the polygon defined by the set C.

Finally, we require that the instantaneous velocity at any time step k, between two consecutive points seperated by a period Δt, is less than the maximal velocity of the robot v_{max}:

$$|\frac{z_e(k+1) - z_e(k)}{\Delta t}| \leq v_{max}, \qquad k = 1, ..., (h-1) \tag{10}$$

(c) **Final Formulation**

The problem can be formulated as a weighted sum of the objective functions subject to the stated constraints:

$$\min \qquad w_1 \times T + w_2 \times \sum_{k=1}^{h}(||z_e(k+1) - z_e(k)||_2^2) + w_3 \times ||z_e(h) - z_g||_2^2$$

$$\begin{aligned} \textit{subject to} \qquad & A_C(k)z_e(k) \geq b_C(k) + (v(k) - 1)M \\ & \sum_{i=1}^{s} v_i \geq 1 \\ & z_e(1) = z_s \\ & |\frac{z_e(k+1) - z_e(k)}{\Delta t}| \leq v_{max}, \qquad \forall k = 1, ..., (h-1) \end{aligned}$$

4 CURRENT PROGRESS AND EXPECTED DELIVERABLES

Up to this day, we completed the installation of the various toolboxes we will use to test our optimization pseudo-code and we began coding the algorithm in figure 4 proposed by [2].

Initialize: h, $v_{\max}$, s, M, r_e, $r_{\mathcal{O}}$, $x_{\mathcal{O}}$, $y_{\mathcal{O}}$, ε;
% Find the circumscribing polygon parameters for obstacles
Eqs. (5.16-5.20);
while $dz_{\text{final}} \geq \varepsilon$ **do**
  % Solve the optimization problem by Gurobi
  $(\mathbf{z}_e, \mathbf{z}_f, dt, \mathbf{dx}_e, \mathbf{dy}_e) = \texttt{Path\_Optimizer}(h, \mathbf{z}_s, \mathbf{z}_g, v_{\max}, \mathbf{m}_{\mathcal{O}}, \mathbf{b}_{\mathcal{O}})$;
  if $\mathbf{dx}_e(2) \geq 0$ **then** *% Find the angel of movement*
    $\theta_e = \arctan(\frac{\mathbf{dy}_e(2)}{\mathbf{dx}_e(2)})$;
  else
    $\theta_e = \arctan(\frac{\mathbf{dy}_e(2)}{\mathbf{dx}_e(2)}) + \pi$;
  end if
  $v = \mathbf{dy}_e(2)/(\sin(\theta_e).dt)$; *% Update the linear velocity*
  $\omega = (\theta_e - \theta_{e_0})/dt$; *% Update the angular velocity*
  % Update the position of the robot using kinematic model
  $x_e = x_e + v.\cos(\theta_e).dt$;
  $y_e = y_e + v.\sin(\theta_e).dt$;
  $\theta_{e_0} = \theta_{e_0} + \omega.dt$;
  $\mathbf{z}_s = [x_e, y_e]$;
  $dz_{\text{final}} = ||\mathbf{z}_s - \mathbf{z}_g||_2$; *% Find the distance to the goal point*
  $T_{\text{total}} = T_{\text{total}} + dt$; *% Update the total time*
end while
Go to the goal position and stop;

Figure 4: Optimization Algorithm [2]

The expected deliverables are listed below, for which we will be uing Matlab CVX [10] and YALMIP, with solvers such as SDPT3, SEDUMI and MOSEK [11]:

(a) Sensitivity analysis on the weights

(b) Sensitivity analysis on the constraint parameters

(c) Simulation with static and dynamic obstacles

(d) Adding/Removing constraints

(e) BONUS: Gazebo simulation for real-time visualization of a moving robot in various situations, such a racing competition, an agricultural ploughing and a basic point to goal pathing

5 POTENTIAL CHALLENGES

Some problems might arise due to time constraints. Moreover, due to the low specification of the PC used, the real-time implementation in gazebo might not be suitable because of the increase in time taken at each step by the algorithm which might result in a different landmark association in the gazebo simulator. Additionally, we might face some problems when implementing both dynamic and static obstacles since this simulation is not addressed by [2].

6 POSSIBLE EXTENSIONS

Convex optimization in motion planning applications has received a lot of attention lately, due to the progress in current state of the art computational algorithms. As stated in [9], the combination of both convex optimization and neural networks can indeed improve the efficiency of current algorithms and even achieve unprecedented results which might revolutionize the real-time motion planning done by robots to explore new environments. Advances in this field will most likely lead to more dependency on wheeled robots as they will become more reliable and even utilized in a wider range of applications.

7 REFERENCES

[1] Mathworks, "Matlab learning," 2020. [Online]. Available: https://www.mathworks.com/help/

[2] M. Zarei, R. S. Novin, and M. T. Masouleh, "Experimental study on optimal motion planning of wheeled mobile robot using convex optimization and receding horizon concept," in *2016 4th International Conference on Robotics and Mechatronics (ICROM)*, 2016, pp. 386–391.

[3] J. Barraquand and J. . Latombe, "A monte-carlo algorithm for path planning with many degrees of freedom," in *Proceedings., IEEE International Conference on Robotics and Automation*, 1990, pp. 1712–1717 vol.3.

[4] A. H. Qureshi, A. Simeonov, M. J. Bency, and M. C. Yip, "Motion planning networks," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2118–2124.

[5] B. Hummel, S. Kammel, Thao Dang, C. Duchow, and C. Stiller, "Vision-based path-planning in unstructured environments," in *2006 IEEE Intelligent Vehicles Symposium*, 2006, pp. 176–181.

[6] Zhijie Zhu, E. Schmerling, and M. Pavone, "A convex optimization approach to smooth trajectories for motion planning with car-like robots," in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 835–842.

[7] W. Serralheiro, N. Maruyama, and F. Saggin, "Self-tuning time-energy optimization for the trajectory planning of a wheeled mobile robot," *Journal of Intelligent Robotic Systems*, vol. 95, 09 2018.

[8] G. P. Bevan, H. Gollee, and J. O'reilly, "Trajectory generation for road vehicle obstacle avoidance using convex optimization," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 224, no. 4, pp. 455–473, 2010.

[9] Z. Zhang, S. Chen, and S. Li, "Compatible convex–nonconvex constrained qp-based dual neural networks for motion planning of redundant robot manipulators," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1250–1258, 2019.

[10] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[11] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.2.*, 2019. [Online]. Available: http://docs.mosek.com/9.2/toolbox/index.html