Risk-Aware Motion Planning for Multi-Robot Systems

Graduation Thesis

Qi Luo (4994736)

Master of Science Thesis

Delft University of Technology

Department of Cognitive Robotics

Risk-Aware Motion Planning for Multi-Robot Systems

Graduation Thesis

Master of Science Thesis

For the degree of Master of Science in Mechanical Engineering
at Delft University of Technology

Qi Luo (4994736)

September 1, 2021

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Copyright © Department of Cognitive Robotics (CoR)
All rights reserved.

Abstract

Motion planning in cluttered environments is challenging for multi-robot systems, in which each robot needs to avoid obstacles as well as other robots. This thesis presents a distributed risk-aware motion planning method for multi-robot systems in dynamic environments. For each robot navigating in a multi-robot scenario, two major risk elements are considered and formalized: a) the collision risk that is assessed using the defined "deformed distance to the centroid of free space" metric, and b) the congestion risk that is assessed via the designed "potential to goal" metric. These risk elements are incorporated into a distributed model predictive control (MPC) framework for risk-aware multi-robot motion planning, in which the collision and congestion risks of each robot are minimized. Simulation results show that the proposed method can improve the robot's safety regarding clearance to each other and obstacles comparing to the baseline method without risk minimization. Moreover, the trajectory efficiency, i.e., time to reaching goals, is also improved thanks to minimizing the congestion risk. We also validate the proposed method in real experiments with a team of Crazyflie 2.1 quadrotors.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor Prof. Dr. Javier Alonso-Mora for his assistance during the thesis project. Besides, I would like to thank my daily supervisor Hai Zhu for his guidance throughout the research.

Delft, University of Technology
September 1, 2021

Qi Luo (4994736)

Chapter 1

Introduction

1-1 Motivation

In the last few years, the development of hardware technology and artificial intelligence has facilitated commercial applications of robots. Autonomous vehicle fleets and aerial robot phalanxes are popular applications of multi-robot systems, as Fig. 1-1 shows. One of the most critical tasks for multi-robot systems is to find appropriate paths for all robots to complete their tasks, which is the main task of multi-robot motion planning.

(a) A quadrotor group sprays pesticides over a corn field[1]

(b) Autonomous vehicles are controlled by urban fleet management platform[2]

Figure 1-1: Applications of quadrator groups in agriculture and military

[1] https://consortiq.com/using-drones-in-agriculture-industry/

[2] https://fleetnewsdaily.com/stratim-unveils-fleet-management-platform-urban-fleets-autonomous-vehicles/

Planning in unknown environments entails plenty of challenges. The uncertainty of navigation and localization increases errors in the position information, and the predicted trajectories of obstacles sometimes differ too much from actual values. These uncertainties obstruct robots from finding available trajectories, which increases the risk in multi-robot motion planning. The risk in motion planning represents the possibility that the robot will not be able to complete its path. Deadlock may happen in the middle of planning due to unexpected terrain. Robots may crowd in a cluttered space but not choose a free region on the side. By minimizing risk in multi-robot motion planning, robots can plan safer and more efficient trajectories to finish their tasks.

Recent studies focus on how to assess collision risk, and how to minimize collision risk in planning. But other risk metrics also profoundly influence the possibilities of accidents. These metrics can evaluate the planning efficiency, power supplies, communication cost, and many different properties of multi-robot motion planning. Due to the limits of computation, it is sometimes inconvenient to involve these metrics in local motion planning as it is challenging to formulate them as functions. Therefore, it is necessary to develop efficient and reasonable metrics to assess risk in multi-robot motion planning.

This thesis presents metrics that can evaluate collision risk and congestion risk in multi-robot motion planning. The metrics will be considered in an MPC-based local multi-robot motion planner. To show the performance of our metrics, we will also adopt traditional risk metrics in the planner. All the planners will be tested in cluttered environments so that the possibilities of collision and congestion will be higher than in general scenarios.

1-2 Contribution

The main contributions of this thesis are:

- A risk assessment method that can evaluate the collision risk and congestion risk in multi-robot motion planning. The method is based on the risk elements, Deformed Distance to the Centroid of Free Space (Dis2Centr) and Potential to Goal (P2g). Dis2Centr involves both obstacle density and distance to obstacles. And P2G evaluates congestion risk by the relative positions and orientations of obstacles and goals. Both metrics assess risk more reasonably than traditional metrics.

- A risk-aware MPC-based motion planner is developed. Incorporated with corresponding risk elements, the result of the planners shows lower risk compared with classic multi-robot motion planners. Safety and efficiency are both improved with the planner.The planner is tested on a multi-robot testing system, which is based on Crazyflie 2.1.

1-3 Thesis Outline

This thesis starts with Chapter 2, where I will introduce the background knowledge of this thesis, including multi-robot motion planning, model predictive control, and risk assessment methods. In Chapter 3, we will compare the risk elements that can evaluate risk in motion planning. And I will focus on the formulations of these metrics and their applications in planning. In Chapter 4, I will discuss the basic formulation of risk-aware motion planning and how to incorporate our elements into it. Different scenarios are built up to the performance of the planners. In Chapter 5, the proposed methods will be tested on Crazyflie 2.1 quadrotor groups to validate their performance in cluttered environments. Finally, in Chapter 6, I will summarize the results of experiments and present some potential works in the future.

Chapter 2

Preliminary

In this chapter, the theoretical background of the thesis is introduced. The knowledge of multi-robot motion planning will be introduced in Section 2-1. Model predictive control is chosen as the trajectory planner in this thesis. The structure and properties will be discussed in Section 2-2. In the end, the state of the art of risk-aware motion planning will be overviewed to show the assessment of risk in motion planning in Section 2-3.

2-1 Multi-robot Motion Planning

Motion planning is a fundamental topic in robotics, which studies how a robot finds an available path without violating any constraints or limits—furthermore, multi-robot motion planning focus on how a group of robots can complete the motion planning task.

The basic idea of single robot motion planning is to generate an available trajectory τ in configuration space $\mathcal{C}$ from its initial position q_{init} to goal q_{goal} without colliding with obstacles $\mathcal{O} \in \mathcal{W}$ in workspace $\mathcal{W}$ [17]. The configuration space $\mathcal{C}$ is the construction of the physical state spaces of robots. Moreover, By following a continuous path in $\mathcal{C}$, a robot will run in a continuous sequence of motions. Various indicators can evaluate the performance of a robot in motion planning. Commonly, the robots are expected to reach the goal within the shortest time, following the shortest trajectory, or meeting any other requirements.

The classic methods to solve motion planning can be divided into global and local motion planning methods. Combinatorial and incremental methods are classic global motion planning approaches. Local motion planning usually depends on local optimization methods to find the available trajectories. The *local* means that at a specific position, the algorithms generate actions of the robots in one or several next time steps. In every time step, this process would repeat until the robots reach the goal.

This section assumes that the variables in planning, including locations and velocities, are deterministic, which means the uncertainties of the variables are ignored in planning. In Section 2-3, the methods to involve the uncertainties in deterministic motion planning will be introduced.

2-1-1 Collision Avoidance

In local motion planning, the robots need to avoid collision with obstacles and other agents in every time step. The collision avoidance methods can be mainly divided into geometric approaches, optimization-based approaches, and artificial potential field. Also, the space decomposition method is introduced, which is a practical approach to solve the multi-robot collision avoidance problem.

The geometric approach reformulates the geometric structure by considering the velocity and position of the obstacles, which means it can avoid collisions with dynamic obstacles. A good example is velocity obstacle (velocity obstacle (VO)) [9]. The volumes of obstacles are extended in velocity orientation, which represents the region with high collision risk. Based on VO, the reciprocal VO (reciprocal velocity obstacle (RVO)) is proposed to solve the oscillation problem in the original VO, which smooths the path and promotes the safety of the planning trajectory[23]. The distributed formulation of RVO will slow down the calculation exponentially with the increasing number of robots. To solve this, the optimal reciprocal collision-avoidance (optimal reciprocal collision-avoidance (ORCA)) transforms the problem into a low-dimensional linear optimization program, which performs well in complex situations that contain thousands of robots at a time[24]. RVO can only be applied to the same holonomic robots with linear dynamics models, and the generalized RVO approximates the dynamics models, which can be used in multi-modal robot systems.

The optimization-based approach is similar to the geometric method, as geometric information plays an important role. The geometric method usually assumes that the velocity of obstacles will not change in the next few time steps. However, the optimization-based method predicts all possible states of obstacles and verifies the predicted vehicle trajectory does not intersect these obstacles [4]. In that case, this method shares similarities with obstacle trajectory prediction. In [4], trajectories of the robots will be generated considering the possible motion of the obstacles in the future. The robot will optimize a "best" position, where the collision is free, and all other requirements of local motion planning can be fulfilled.

The artificial potential field (artificial potential field (ARF)) assesses the distance between the robots and obstacles[16] to avoid collisions. The free spaces in the workspaces are mapped as a potential field, and the potential energy will decrease with the distance to obstacles. ARF requires accurate information of the surrounding environments, which makes it inaccurate in uncertain environments [10]. The control barrier function also takes into account ARF, which ensures the Lipschitz continuity in the constraint set[1].

Another interesting method of collision avoidance is space decomposition. One typical method is Buffered Voronoi Cells (Buffered Voronoi Cells (BVC)) in local collision avoidance[27]. BVC method separates the space into non-overlapping Voronoi cells and plans the paths in the independent spaces. It highly relies on navigation and localization accuracy, and the number of robots is required to be small enough[28]. Another method divides space into high-resolution discrete grids, finds the optimal path by graph searching, and smooth the path by the dynamic models of the robots[15]. This method can be adopted to giant-sized robot groups with linear dynamics models. Both methods are sensitive to navigation errors, which is not suitable to be combined with risk-awareness methods.

2-1-2 Multi-Robot Motion Planning

As we have studied single robot motion planning, the main topic of this section is how other robots in the workspace influence planning. Assume a group of n robots running in a workspace $\mathcal{W}$ and planning in its k-dimension configuration space $\mathcal{C}^k$. The motion planning turns into a problem in configuration space $\mathcal{C}_n = \sum_{i=1}^{n}(\mathcal{C}_i^k)$. The dimension of the whole problem turns into $Dim(\mathcal{C}_n) = n * k$, which means the dimension increase linearity with the number of the robots, which will lead to high computation costs and makes the problem intractable. A classic way to solve the problem is to separate the problem into n single robot motion planning with $n - 1$ dynamic obstacles. The configuration space for a specific robot can be defined as equation 2-1. The core of the problem is to obtain other robots' states while planning, which means the communication between robots is one of the keys of multi-robot motion planning.

$$\mathcal{C}_i = \mathcal{C}_i \bigcup \left(\bigcup_{j, j \neq i}^{n} \mathcal{C}_j \right) \tag{2-1}$$

Setting priority for robot group is a common solution[2], which orders the robots in the group with different priorities. The highest-order robot will plan first without considering the others. Then the second-order robot will plan based on its configuration and the planning result of the first one. Another option is that all robot planning without considering the states of others independently [11]. Then perform a collision check to find out whether the result is collision-free. If not, deform the configuration space by optimization rules and repeat the process. Both of the methods show promising results in practice. However, as there is no communication between robots in planning, it will take longer when the distances between robots are constrained.

For local motion planning, different communication methods can be adopted. As the planning performs in every step and the result is based on the instantaneous configuration space, the configurations can be updated in time by the planning result of the other robots. The other robots can be regarded as dynamic obstacles that can communicate their states during planning. The difference in communication methods in local motion planning will be introduced in Section 2-2.

2-2 Model Predictive Control

In this section, the general formulation of MPC is introduced. We analyzed the effects of cost functions, equality constraints, and inequality constraints. Also, different communication approaches in MPC are compared.

2-2-1 Introduction

The general idea of Model Predictive Control (model predictive control (MPC)) is to find an optimal input that can minimize objective functions under different constraints within finite time T_N[8]. The result of an MPC problem will be a set of input values at each time step within the time horizon, and only the first input value will be applied. This method can deal with nonlinear, multivariate problems with complex constraints, making it a popular algorithm to solve complex optimization problems.

For a discrete multi-robot system with n MAVs, given the obstacle position $p_o^{1:N}$ and the estimated initial state $\hat{\mathbf{x}}^k, k \in [0, N-1]$, the cost function is designed to optimize the trajectory from the initial position to the goal without collided with obstacle and other robots. The basic formulation of an MPC designed for motion planning is shown as equation 2-2

$$
\begin{aligned}
\min_{\mathbf{x}^{1:N}, \mathbf{u}^0 : \mathbf{v}_{-1}} \quad & \sum_{k=0}^{N-1} J^k \left(\mathbf{x}^k, u^k \right) + J^N \left(\mathbf{x}^N \right) \\
s.t. \quad & \mathbf{x}^0 = \hat{\mathbf{x}}^0 \\
& \mathbf{x}^k = f \left(\mathbf{x}5^{k-1}, \mathbf{u}^{k-1} \right) \\
& O(\mathbf{x}^k) \cap O(\mathbf{p}_o^k) = \emptyset \\
& O(\mathbf{x}^k) \subset \mathcal{C} \\
& \mathbf{u}^{k-1} \in \mathcal{U}, \quad \mathbf{x}^k \in \mathcal{X}, \quad \forall k \in \{1, \ldots, N\}.
\end{aligned}
\tag{2-2}
$$

1. $\sum_{k=0}^{N-1} J^k \left(\mathbf{x}^k, \mathbf{u}^k \right)$ is the stage cost among 0 to $N-1$ time steps. $J^N \left(\mathbf{x}^N \right)$ is the terminal cost at Nth time step. As the aim of a MPC is to minimize the cost function, by using different cost functions, we can control the robot to perform as our expectations.

   The robot should potentially run forward to the goal position. Therefore, it is necessary to minimize the distance between the current location and goal position. An option is to set a terminal cost as the equation,

$$
J^N \left(\mathbf{x}^N \right) = w_{\text{goal}}^N \frac{\left\| \mathbf{p}_g - \mathbf{p}_{ro}^N \right\|}{\left\| \mathbf{p}_g - \mathbf{p}_{ro}^0 \right\|}, \tag{2-3}
$$

   where, $\mathbf{p}_g$ is the goal position and w_{goal}^N is the weight coefficient.

Another popular cost function for motion planning is to minimize the input value, as the equation 2-4 shows. The robot will achieve a smoother performance with help of this cost. By applying this cost, robots are less likely to show sharp and quiver movement in practice [29].

$$\sum_{k=0}^{N-1} J^k\left(\mathbf{u}^k\right) = w_{\text{input}}^k \left\|\mathbf{u}^k\right\|, \quad k = \{1, 2, \ldots, N-1\}, \tag{2-4}$$

where, w_{input}^k is the weight coefficient.

2. $\mathbf{x}^0 = \hat{\mathbf{x}}(0), \quad \hat{\mathbf{x}}^k = \mathbf{f}\left(\hat{\mathbf{x}}^{k-1}, \mathbf{u}^{k-1}\right)$ is the equality constraint that confirms the dynamic continuity of the whole system. All robots in the real world follow their own dynamics model, and a precise dynamic model can describe the robots' motions more realistically, which will achieve better performance. However, a too complicated and highly nonlinear dynamic model may lead to higher computation costs, even infeasible results. Therefore, dynamics models should always match the workload of application scenarios.

3. $O(\mathbf{x}^k) \cap O(\mathbf{p}_o^k) = \emptyset$ show that there is no overlapping between the occupied spaces of robots and obstacles, which means there is no collision between robots and obstacles and for different types of definition approaches of obstacles, the constraints are following,

   - For spherical robot and obstacle models, the constraint can be:

   $$\left\|\mathbf{p}_r^k - \mathbf{p}_o^k\right\| \geq R_r + R_o, \tag{2-5}$$

   where, R_r and R_o are the radius of the robot and obstacle.

   - For convex obstacles, it is possible to set the half-space between robot and obstacles as collision constraints,

   $$A_o\mathbf{x}^k \leq b_o. \tag{2-6}$$

   Although the linearization simplifies the problem, the feasible spaces in configurations are also compressed.

4. $O(\mathbf{x}^k) \in \mathcal{C}$ is a constraint that confirms that the robot stays in its configuration space.

5. The time horizon $N = T_N/\Delta t$ decides how "far" the robot can predict. By increasing N, the prediction ability of the method can be enhanced. However, a too-long horizon results in unacceptable calculation time for one step, leading to local deadlock.

One of the most significant advantages of MPC is that it is convenient to apply the risk-aware component into the inequality constraints and nonlinear objective functions of MPC. In Section 2-3, the methods that combine MPC with risk-aware methods will be introduced.

2-2-2 Multi-Robot Communication in MPC

As mentioned in Section 2-1, by dividing the multi-robot motion planning into several single robot motion planning with dynamic obstacles, the dimensions of configuration spaces are reduced, which simplifies the problem. However, different from dynamic obstacles, the planning result of other robots is available during planning. In MPC, the planning result is the control input, which is equivalent to a trajectory path. In every stage in MPC, inequality constraints avoid collision with obstacles at the planned position. Therefore, by communicating their planning trajectories, the robots can approximate the future states of the others without prediction. Here we introduce three main approaches, centralized sequential MPC, distributed MPC, and decentralized MPC.

Centralized Sequential MPC

Similar to the global multi-robot motion planning with priority, this method prioritizes each robot in the group and solves MPC sequentially. The robot with the highest priority plans first and sends its planning result to the next one at each step. The next one plans its trajectory to avoid collision with the planning result of the last robots in the current time step and the inferior robots in the previous time step. After the last robot completes the planning, all the robots execute the optimization result simultaneously. After that, inferior robots send their planning trajectory to prior robots to estimate their position next step. Centralized sequential planning transfers the actual execution in the current time step, which will help the robot avoid collision better since the robot's prediction is equal to real value theoretically. Compared to optimizing all trajectories, this method saves computation costs and confirms the planning precision. One disadvantage of sequential planning is that when the number of robots is too large, the robots with lower priority will perform not so "freely" like the senior ones[6]. This problem can be lightened by change increasing the parallel robots and reduce the priority levels. Also, accurate localization of robots and high sampling rates are necessary, so the predictions of other robots will not differ too much from the actual positions.

Distributed MPC

The distributed MPC requires all robots to communicate once a time step to exchange their planning trajectories. Unlike centralized sequential MPC, the robots avoid the trajectories of all the other robots at the previous time step. After all the robot completes the planning, they execute the optimization result at the same time. The computation cost in distributed MPC is the same as centralized one, but the communication cost is much lower when the computation is performed in the planner on robots individually. Also, the optimization problem can be solved parallelly in practice, which is much faster than the sequential one. Although the prediction of other robots may not be the same as actual execution, the result is still reliable when the sampling rate is high enough. One of the shortages of distributed MPC, which is also the shortage of centralized MPC, is that all robots are required to move simultaneously. Due to the communication delay and response time, it is a challenge when the number of robots is high in practice.

Decentralized MPC

The decentralized MPC only requires the robot to communicate their locations but not the total planning result to each other. It is even possible that the robots using perception approaches to locate the others, and the problem is simplified into single robot motion planning with dynamic obstacles. The main focus in decentralized MPC is trajectory prediction. As mentioned in section 2-1, both constant velocity model (CVM) and learning-based prediction are applicable in practice. The performance of decentralized MPC highly depends on the trajectory prediction. When a rough prediction model like CVM is used, the performance will be highly degraded, even though it can prevent collision in some scenarios. The most significant advantage of this method is that the robot can perform optimization and execute the result individually. Moreover, the errors from other robots will not influence the results.

2-3 Risk-Aware Motion Planning

Risk in motion planning, which can be defined as *relative likelihood of the robot not being able to finish the paths* [25], is an important component to evaluate the quality of the planning result. If a trajectory indicates low risk, a robot will possibly run from the initial to the goal by following the path. Risk is a very general element, and the methods to scale risk numerically is the main topic of this section. Currently risk can be parameterized as **planning uncertainty** or **a risk function consisted by states of robots** [26]. Collision probability is a classic method to assess planning uncertainty, especially localization uncertainty. Restricting the collision probability between one robot and one obstacle in motion planning helps the robot avoid collision with obstacles under uncertainty. There are also many other risk functions to evaluate the risk of robots by different specialties of robot performance. By combining these functions with motion planners, the trajectory will be more reasonable, and the likelihood of the robot completing the path will be increased. In this thesis, I mainly focus on the reasoning elements that can assess various risk in motion planning.

2-3-1 Risk Reasoning by Elements

Apart from planning uncertainty, risk functions can evaluate the comprehensive risk of the planning by risk elements. The risk here is more than the collision risk caused by planning, but all metrics can influence the robots finishing the path.

The risk elements in robot motion planning should follow specific requirements. Six axioms are proposed to normalize the design of risk elements [18], monotonicity, translation invariance, positive homogeneity, subadditivity, comonotone additivity, and law invariance[18]. However, not all popular risk elements can fulfill these axioms. Therefore, it is necessary to formulate risk elements into proper functions in practice.

Risk elements reflect the properties of risk from different perspectives. They can be divided into three types elements, instantaneous risk elements, action risk elements, and global risk elements [26]. The main difference between these risk elements is the dependence of the information from planning history. The risk elements belonging to one category are independent, and the relationship between different types of elements are shown as {instantaneous, action} $\subset$ global [26].

Instantaneous risk elements adopt no information from planning history but only the states in the current time step. Instantaneous risk elements are commonly involved in local motion planning, as both of them use the current state information. Classic instantaneous risk elements, like clearance and visibility, mainly evaluate collision risks with surrounding objects in the workspace, which will also be introduced in the following sections. Other instantaneous risk elements focus on the risk caused by the properties of robots, such as motion singularity and deadlock, which evaluate the risk that the robots deadlock in unexpected space.

Action risk elements evaluate the risk in limited planning history, which means how the last finishing robot states will influence the risk of the current state. A typical element is the action length, which evaluates the norm distance of input between neighboring time steps. An extended action length shows that the rapid movement of the robots, which will lead to higher risk in planning. Action risk elements can reflect the properties of robot movements in a limited time window, which are more reasonable compared to instantaneous elements. However, the action risk limits the response flexibility of robots, which makes it hard to avoid collisions in ever-changing environments. Therefore, it is essential to balance the importance between instantaneous and action risk elements in practice.

Global risk elements evaluate the general risk of the entire planning history, which is a summary of instantaneous and action risk elements. Apart from analyzing the previous risk elements over entire history, global elements can also reflect global risk, like communication delay and the remaining power of the energy system. These elements assess the risk caused by the complexities and stabilities of working environments. An unstable working environment will lead to unexpected danger, and the robots will be more likely to meet accidents. As it is complex to combine global risk elements with local motion planning, it is not in the scope of this thesis.

2-3-2 Applications of Risk Elements in Local Motion Planning

Although the risk elements above can evaluate different risks in motion planning, not all of them are suitable for local motion planning, which requires numerical formulations of risk elements. Also, the computation efficiency should be good enough to fulfill the requirement of a high sampling rate. A classic way is to apply risk elements r^k to the cost functions of MPC, as the equation 2-7 shows.

$$
\begin{aligned}
\min \quad & J\left(\mathbf{x}^k, \mathbf{u}^k, \mathbf{r}^k\right) \\
s.t. \quad & \mathbf{x}^0 = \hat{\mathbf{x}}^0, \quad \mathbf{x}^k = f\left(\hat{\mathbf{x}}^{k-1}, \mathbf{u}^{k-1}\right) \\
& O(\mathbf{x}^k) \cap O(\mathbf{p}_o^k) = \emptyset, \quad O(\mathbf{x}^k) \subset \mathcal{C} \\
& \mathbf{u}^{k-1} \in \mathcal{U}, \quad \mathbf{x}^k \in \mathcal{X}, \quad \forall k \in \{1, \ldots, N\}
\end{aligned}
\tag{2-7}
$$

By using MPC, the risk of the result assessed by $\mathbf{r}^k$ can be minimized, by which the robot will run in a safer trajectory. In chapter 3, we proposed several navigation risk elements that are suitable for motion planning. In chapter 4, we discussed the formulations to combine these elements with a classic local MPC motion planner.

Chapter 3

Risk Assessment

In this chapter, I mainly focus on risk assessment methods that evaluate navigation risk in motion planning. As mentioned in the previous sections, risk in motion planning is a relative likelihood of the agent not finishing the path, which is numericalized by risk elements in planning. Using risk elements to evaluate risk numerically is efficient and straightforward in practice. This chapter discusses risk elements that assess the collision and congestion risk, whose pros and cons are analyzed in different scenarios. New elements are also developed to assess risk more reasonably. The Free Space Area and the Distance to the Centroid of Free Space are two elements that can assess the danger of the collision between a single robot and other obstacles in workspaces. Potential to Goal is the element that can evaluate the congestion risk for multi-robot systems. All these elements can be formulated numerically.

The outline of this chapter is the following. In section 3-1, the collision risk elements are introduced. Examples are generated to show the pros and cons of these elements. In section 3-2, the idea of congestion risk are introduced. Potential to Goal is implemented to assess congestion more reasonably, compared with existing numerical elements.

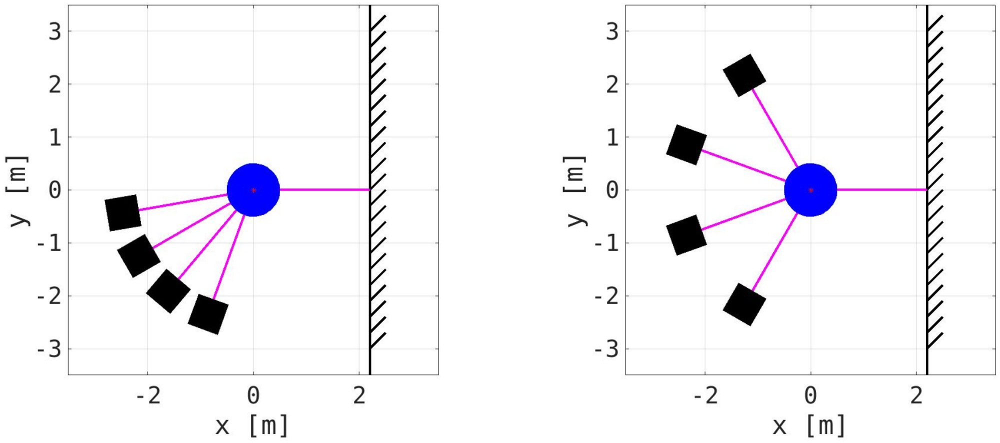
3-1 Navigation Risk Among Static Obstacles

3-1-1 Collision Risk Elements

Collision risk elements are used to evaluate the collision risk between robots and obstacles. The word *"collision"* means a statement of a robot striking against other obstacles in the workspace, and "collision risk" means the risk caused by the relative location between robots and other obstacles which may lead to collisions. In multi-robot motion planning, robots are more likely to work in safer spaces by minimizing collision risk, which means they are less likely to collide with obstacles. The following collision risk elements are all based on the current position information of robots and other obstacles, which contains no history working information. In the scope of this thesis, collision risk elements include the Sum of Distances to all obstacles, the Clearance (the closed distance to obstacles), the Free Space Area, and the Distance to the Centroid of Free Space.

Sum of Distances

Sum of distances $\Sigma_d = \sum_i d_i$ is the most direct element to evaluate the collision risk. The "distance" between two obstacles is the clearance between the closest two points on the obstacles. Σ_d is also named as collision potential field, which is widely used in the studies of motion planning [29][28][12]. The value of the sum of distance evaluates how close a robot is to other obstacles. The limitation of this element is that all position information is added, which ignores the other statement information, like the orientations and density of obstacles. The Fig. 3-1 show that the sums of distance are the same in the left and right scenarios, but in 3-1(b) the robot is situated in a narrower space, which shows higher risk intuitively. Therefore, more elements are required to evaluate risk in this scenario.



(a) The obstacle partly surrounds the robot (b) The obstacle totally surrounds the robot

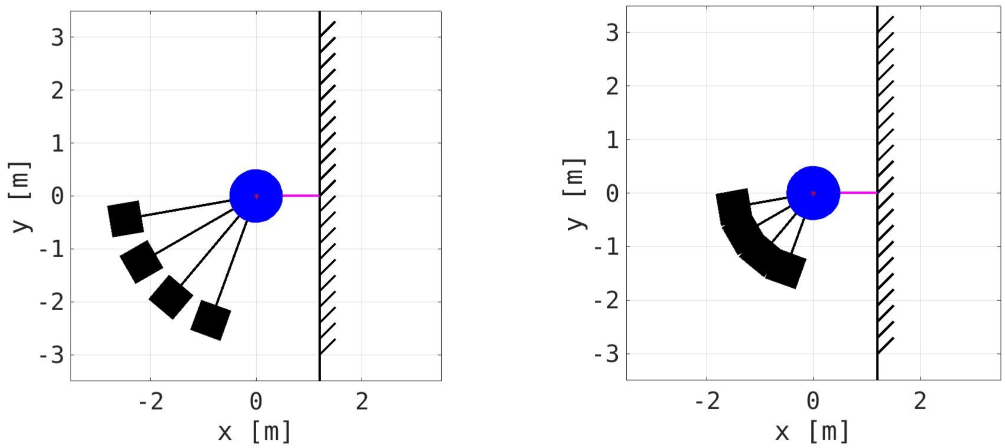
Figure 3-1: Comparison between two scenarios of Sum of Distance. The black squares represent the obstacles and the blue circle represents the robot.

Clearance

Clearance $d_{min} = \min(d_i)$, which is the closed distance to all other obstacles, is also a pretty widely used elements. Unlike the sum of distances, every distance is compared with each other, and only the shortest one makes a difference. This element also shows its limitation in some situations. As the Fig. 3-2 show, the clearances are the same in the two scenarios. However, in left scenarios 3-2(a), the robot is not so tightly surrounded as the right one, showing the lower probability of colliding with surroundings. Sum of Distance and Clearance only contain the scale information between the robot and other obstacles, excluding relative position information among obstacles. In the following, elements that adopt the obstacle densities are introduced to assess collision risk more reasonably.

Free Space Area

Free space area S_{Free} represents the size of the secure space where a robot can move without collisions. S_{Free} can be formulated as the intersection of the half-space feasible regions, whose edges are vertical to the centerline of the robot and the obstacles. The free space is always convex, and the obstacle density influences the shape of the free space. Position information is adopted in S_{Free} more sufficiently than previous elements. The weakness of the free space area shows when the obstacles tightly surround the robot. As the Fig. 3-3 show, the robot in 3-3(a) is more closed to obstacles, and the collision risk is higher in the left scenario. However, the S_f is the same as the right one in 3-3(b). This example indicates that the free space area can only represent the obstacle density but is short of assessing the distance to obstacles.

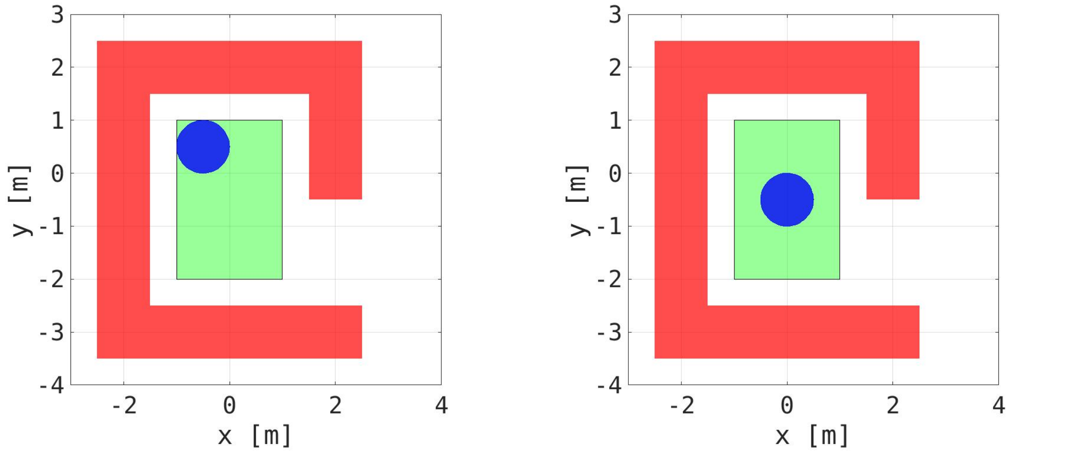


(a) The obstacles are far away from the robot. (b) The obstacles are closed to the robot.

Figure 3-2: Comparison between two scenarios of Clearance. The black squares represent the obstacles and the blue circle represents the robot. The red line represents the clearance.

Distance to the Centroid of the Free Space

Distance to the centroid of the free space (distance to the centroid of the free space (Dis2Centr)) d_{centr} indicates the distance to the geometry centroid of the free space, which is a complement of the free space area, as the Fig. 3-4 shows. For a given free space, the centroid of free space always locates in the middle and keeps the same distance with surrounding obstacles. Consider a 2-demension polygon free space



(a) The robot locates close to obstacles (b) The robot locates at the center of free space

Figure 3-3: Comparison between two scenarios of Free Space Area. The red area represents the obstacles and the blue circle represent the robot. The green region represents the free space.

with a S_{Free} area and a group of vertices $\mathbf{p}_i = (x_i, y_i) \in \mathcal{P}_n$. The centroid of this free region $\mathbf{c}_{\text{centr}} = (c_x, c_y)$ can be computed as:

$$\begin{aligned} c_x &= \frac{1}{6S_{\text{Free}}} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \\ c_y &= \frac{1}{6S_{\text{Free}}} \sum_{i=0}^{n-1} (y_i + y_{i+1})(y_i x_{i+1} - y_{i+1} x_i) \end{aligned} \tag{3-1}$$

Only the boundaries that shape free space influence the value of d_c, which mutes the "invalid" obstacles. It is fascinating that when the robot locates at $\mathbf{c}_{\text{centr}}$, it also is closed to all obstacles equally. It seems reasonable to use the sum of distances to valid obstacles to assess collision risk. However, the number of valid obstacles varies with the robot position in a cluttered environment, making the value of Σ_d discontinuous. The centroid of the free space changes continuously with the robot position, which keeps the numerical value of collision risk smooth in plannings.

3-1-2 Deformed Distance to the centroid of the Free Space

To sum up, the elements based on distance and obstacle density are necessary for assessing collision risk. The absolute distance information evaluates how close the robot is to a location, and the obstacle density evaluates how tightly obstacles surround the robot. Therefore, I propose that the combination of distance to the centroid of the free space and area of free space is sufficient enough to evaluate risk in most situations in multi-robot motion planning, as the equation 3-2 shows.

$$h_{centr} = \frac{(\mathbf{p}_{\text{r}} - \mathbf{p}_{\text{centr}})^T (\mathbf{p}_{\text{r}} - \mathbf{p}_{\text{centr}})}{S_{\text{Free}}} = \frac{d_{\text{centr}}^2}{S_{\text{Free}}} \tag{3-2}$$

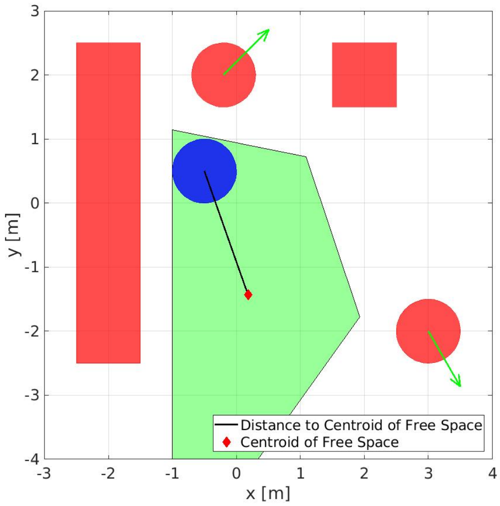


Figure 3-4: The scenario of Distance to the Centroid of the Free Space. The red area and the green arrows represent the obstacles and their velocities, the blue circle represent the robot. The green region and the black line represent the free space and the distance to the centroid of the free space.

In the previous subsection, the intersection of the half-space feasible regions represents the free space. The dynamic constraints restrict the movement of a robot, which means the obstacles that locate too far away from robots make no difference to the "freedom" of a robot. The distances to the workspace's boundary hardly influence the planning of the robot. Therefore, I propose the Valid Free Region to reshape the free space, limiting the biggest free area of robots. The intersection of the origin free space and the valid free region formulates the reshaped free space, eliminating the obstacles or boundaries in the distance.

The most reasonable valid free region should be a circle area around the robot, as the distances to the boundary of the valid free region are equal. However, it is challenging to compute the area of a space that is shaped by both straights and curves using geometric methods. Therefore, in this thesis, the valid free region is represented by a rectangle region of length a_{centr} and width b_{centr}, as the Fig. 3-5 shows. The orientation of the valid free region should be the same as the orientation of the robot velocity. When the value of a_{centr} and b_{centr} are different, the orientation of the rectangle will always keep the same as the robot movement, which equals the distance to the boundary of the rectangle in the forward direction. The region shaped by the blue dotted line is the biggest valid region, the yellow polygon is the original free space, and the green intersection area of these two spaces is the reshaped free space.

Compared with other classic collision elements, the deformed distance to centroid shows better performance will assess risk in a cluttered environment, like a narrow corridor. A simple experiment can prove this advantage, as the Fig. 3-6 shows. Two MPC local motion planners are designed as the baseline in section 2-2. The only difference between these two planners is that d_{centr} is involved in the cost function in

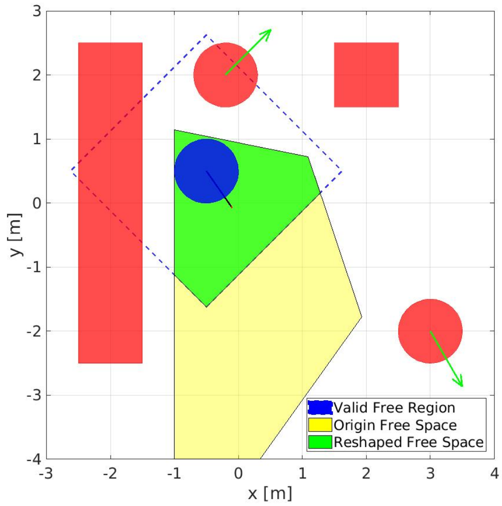


Figure 3-5: Comparison between the origin free space and the reshaped free space. The green region, the reshaped free space, is the intersection of the blue-spot valid free region and the yellow origin free space

one of the planners, and the other adopts the potential field. The result shows that the planner with d_{centr} prefers to pass around the obstacle group but not go through it, as the free space in the middle of the obstacle group is much smaller than the outside. Also, the closer the robot is approaching obstacles, the higher d_{centr} will be. This simple experiment shows that the deformed distance to the centroid can assess collision risk reasonably. The experiments will carry on in chapter 4 to test its performance in more complex environments.

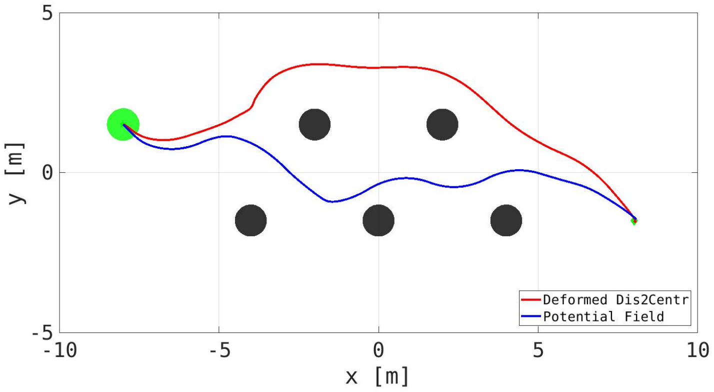


Figure 3-6: The planning result by the planner involving deformed distance to centroid and potential field. The green circle represent the robot and its initial position.

3-2 Navigation Risk Among Moving Agents

3-2-1 Congestion Risk

Congestion is a situation in which there is too much traffic, and movement is strenuous. For a multi-robot system, when the robots crowd in a narrow space, the robots may block the paths of each other, even deadlock in the middle. The congestion risk is the risk that the robots cannot finish the path but crowd in the unexpected space. Since robots' task is to find a collision-free trajectory from initial to goal positions, the goal position plays an essential role in navigation. Also, for robots working in a 2-dimensional workspace, it is crucial to involve the robot velocity to solve congestion problems [5]. Therefore, it is more complex to assess congestion risk in practice than collision risk.

Currently, a cellular automaton model has been developed to evaluate the congestion. The workspace is divided into grids, and the congestion risk in each grid is computed discretely [21]. The numerical value of congestion on each grid is derivable, but the computation cost is pretty high. This method is more appreciate to be applied in global motion planning methods, and the congestion potential field is a better choice when using local motion planning methods. The basic congestion potential field is based on the position information of robots. The behavior potential field is a promotion to evaluate the congestion risk in dynamic environments, as the equation 3-3 shows [14]. The explanation of equation 3-3 is introduced in [14]. The Diagram 3-7 shows the geometric relationship of a robot group in a workspace. The behavior potential field captures the relative positions and directions and absolute velocities of moving agents.

$$R\left(\mathbf{p}_i, \theta_{ij}, \mathbf{p}_j, \mathbf{v}_j\right) = \sum_{i=1}^{n} \alpha_h \frac{\exp\left[\kappa \cos\left(\theta_{ij}\right)\right]}{2\pi I_0(\kappa)} \times \exp\left[-\frac{\left(\mathbf{p}_i - \mathbf{p}_j\right)^T \left(\mathbf{p}_i - \mathbf{p}_j\right)}{2\sigma_h}\right] \left\|\mathbf{v}_j\right\|. \tag{3-3}$$

Risk Level Set (Risk Level Set (RLS)) is another choice to evaluate collision risk [20]. Similar to behavior potential risk, *RLS* captures the relative positions between the robot and other agents. The only difference is the absolute velocities of moving obstacles, as the equation 3-4 shows.

$$R(\mathbf{p}_i, \mathbf{p}_j, \mathbf{v}_j) = \sum_{i=1}^{n} \frac{\exp\left(-\left(\mathbf{p}_i - \mathbf{p}_j\right)^T \Omega \left(\mathbf{p}_i - \mathbf{p}_j\right)\right)}{1 + \exp\left(-\alpha \mathbf{v}_j^T \left(\mathbf{p}_i - \mathbf{p}_j\right)\right)} \tag{3-4}$$

The basic constructions of both methods are a distance peak function multiplied by a twisting function. The distance peak function represents the collision risk discussed in section 3-1. The twisting function skews the original risk functions by obstacle velocities. Both twisting functions capture the relative direction between relative position and absolute velocities of moving agents. However, both methods sacrifice the collision risk to meet the requirement of congestion risk. Therefore, I propose a new congestion risk element: Potential to Goal.

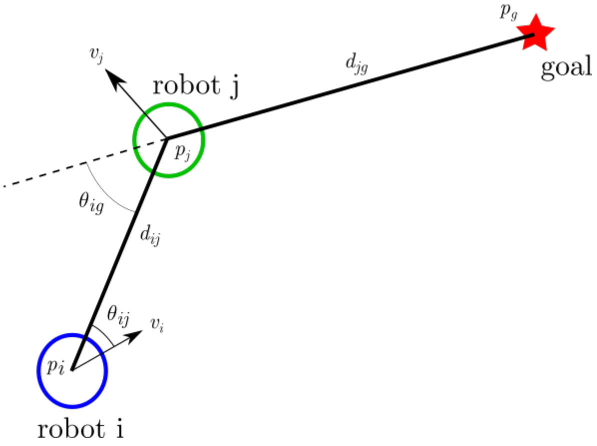


Figure 3-7: Geometric diagram of a objective robot, its goal position, and a moving robot in workspace

3-2-2 Potential to Goal

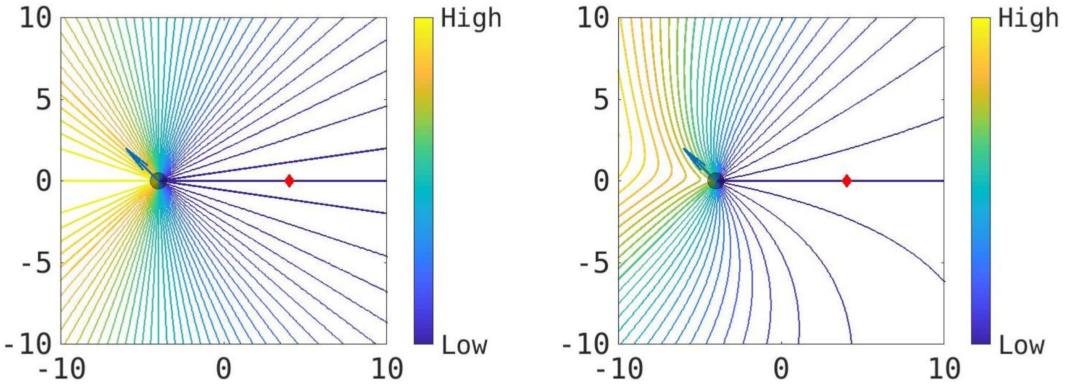
Unlike previous congestion risk elements, *Potential to Goal* (Potential to Goal (P2G)) twists the navigation cost by the velocities of moving agents. The congestion risk formulated by *Potential to Goal* is shown as equation 3-5,

$$R(\mathbf{p}_i, \theta_{ig}, \mathbf{p}_j, \mathbf{p}_g, \mathbf{v}_j) = \left(\sum_{j=1}^{m} \left[\frac{1}{2m} \frac{cos\left(\theta_{ig}\right) + 1}{1 + \exp(-\beta \mathbf{v}_j^T * (\mathbf{p}_i - \mathbf{p}_j))} \right] + 1 \right) \|\mathbf{p}_i - \mathbf{p}_g\| \tag{3-5}$$

where, m is the number of other moving agents in the workspace, p_i, $p_j$and p_g the positions of the objective robot, moving agents and goal, v_j is the velocity of moving agents, θ_{ig} is the angle between robot to agent and agent to positions and β is the parameter controlling the smoothness of the potential.

The twisting parameter of potential to goal, as the equation 3-6 shows, consists of two main parts, the intention of motion $I_{\theta_{ig}}$ and skewing parameter T_{vel}.

$$\begin{aligned}
P2G(\mathbf{p}_i, \theta_{ig}, \mathbf{p}_j, \mathbf{p}_g, \mathbf{v}_j) &= \sum_{j=1}^{m} \left[\frac{1}{2m} I_{\theta_{ig}} T_{\text{vel}} \right] \\
&= \sum_{j=1}^{m} \left[\frac{1}{2m} \frac{cos\left(\theta_{ig}\right) + 1}{1 + \exp(-\beta \mathbf{v}_j^T * (\mathbf{p}_i - \mathbf{p}_j))} \right] \\
I_{\theta_{ig}} &= cos\left(\theta_{ig}\right) + 1 \\
T_{\text{vel}} &= \frac{1}{1 + \exp(-\beta \mathbf{v}_j^T * (\mathbf{p}_i - \mathbf{p}_j))} \\
cos\left(\theta_{ig}\right) &= \frac{\left(p_j - p_i\right)^T \left(p_g - p_j\right)}{\|p_j - p_i\| \, \|p_g - p_j\|} \\
I_{\theta_{ig}} T_{\text{vel}} &= \frac{cos\left(\theta_{ig}\right) + 1}{1 + \exp(-\beta \mathbf{v}_j^T * (\mathbf{p}_i - \mathbf{p}_j))}
\end{aligned} \tag{3-6}$$



(a) $I_{\theta_{ig}}$

(b) $P2G$

Figure 3-8: Contours of one example of $I_{\theta_{ig}}$ and $P2G$. The line with lighter color represents higher risk positions.

The intention of motion $I_{\theta_{ig}}$ represents the relative congestion potential between objective robot, moving agents, and goal. We proposed that when θ_{ig} is equal to 0, the objective robot i shows the highest risk to congest with moving agent j. When the robot i has passed by j, the congestion risk is closed to 0. Similar to *RLS*, skewing parameter T_{vel} is applied to deform the intention of the motion to the velocity direction of moving agents. P2G is higher along the direction of the velocities of moving agents than the original intention of motion. The Fig. 3-8(a) and 3-8(b) illustrate the contours of *Potential to Goal* before and after skewing. The moving agent locates at (-4,0), and the robot's goal is (4,0). The velocity of the robot is (-1,1), as the arrow in Fig. 3-8 shows. The lowest risk region keeps the same after skewing. However, the highest risk region is skewed to the direction of velocity $\mathbf{v}_j$.

The congestion risk is defined as the risk of robots congesting together but not heading to their goals. Therefore, it is more reasonable to combine them with the distance to the goal position. The Fig. 3-9 shows the contour of congestion risk formulated by *Potential to Goal*. As the Fig. shows, when a robot i is running from A to goal position, the congestion risk is higher than origin navigation risk around initials. After passing by the robot j, the congestion risk hardly influences the navigation cost. The robot will run directly to the goal position.

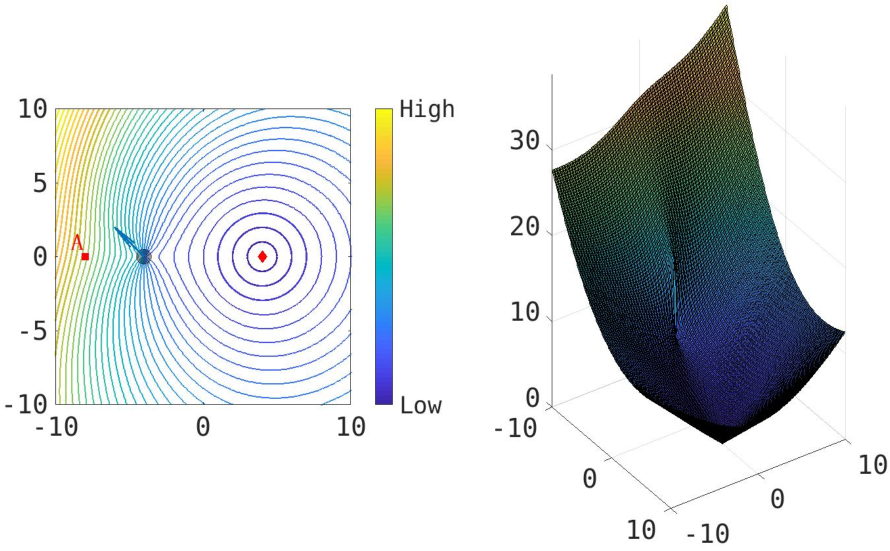


Figure 3-9: The Contour and the value of congestion risk formulated by Potential to Goal

Chapter 4

Risk-Aware Multi-Robot Motion Planning

In this chapter, the proposed risk-aware multi-robot motion planning is the main topic, which can navigate the robot from the initial to the goal position while minimizing the risk of trajectory. In this scope, we focus more on the navigation risk in Chapter 3. Collision risk among static obstacles and congestion risk among moving obstacles are the central parts of our study. To study the performance of collision risk elements, we will test a single robot in cluttered environments with static obstacles. As for the performance of congestion risk elements, we mainly focus on how a group of robots swaps their positions in environments without static obstacles.

The core of the work is the combination between risk elements and the classic multi-robot motion planner. Therefore, we introduce distributed multi-robot model predictive control as the baseline of multi-robot motion planning. By modifying objective functions of the baseline with risk elements, the planner can evaluate risks from different perspectives. As discussed in section 2-2, the distributed multi-robot MPC will be solved parallelly by the same computer, using the planned trajectory from the last time step as the prediction of current states of the robot.

The outline of this chapter is shown as follows. In section 4-1, the formulation of the risk-aware motion planning and the combination of different risk elements will be introduced. In section 4-2, the basic scenarios and settings of my planner will be described. In section 4-3, I will introduce the test scenarios of collision risk elements and compare their performance by collision number, clearance, and average velocity. In section 4-4, I will introduce the test scenarios of congestion risk elements, which shows how to formulate a risk-aware motion planner to balance the efficiency and safety of planning trajectories. In section 4-5, I focus on the influence of the uncertainty of robots' localization.

4-1 Problem Formulation

This section presents the risk-aware multi-robot motion planning that we used in the following experiments. All components of the MPC planner are introduced in this section. This MPC planner can solve both single and multi-robot robot motion planning problems, which will be adopted in Sections 4-3 and 4-3 to check the performance of collision and congestion risk elements.

4-1-1 Baseline

The main topic of this scope is using multi-robot motion planning to plan the expected trajectory to finish the task. The basic formulation of MPC has been introduced in section 2-2. A group of n robots working in a bounded workspace $\mathcal{W} \subseteq \mathbb{R}^2$. Adopting a 2-dimensional workspace but not a 3-dimensional one is because the collision and congestion problems are more complicated in 2-dimensional environments, as robots cannot move vertically to avoid collisions or deadlocks in cluttered environments. Therefore, the performance of risk elements can be shown more clearly in a 2-D workspace.

Collision Constraints

Each robot $i \in \mathcal{R} = \{1, 2, ..., n\}$ is represented by a circle with a radius r, as the real geometric model of robots is complicated, which will reduce the computation cost of collision checking at the same time. Each static obstacle $o \in \mathcal{O} = \{1, 2, ..., m\}$ is represented by ellipses or rectangle with width $2a$, height $2b$ and orientation γ. The shapes of static obstacles in cluttered environments are usually irregular and hard to be indicated mathematically. The approximation can simplify the collision checking but keep the basic geometry information of obstacles as well. Dynamic obstacles are not in the scope of this article, as other agents can be regarded as dynamic obstacles in risk-aware multi-robot motion planning, and no more dynamic obstacles are required in tests.

One of important constraints mentioned in section 2-2 is the collision free constraints. Even though robots and obstacles are approximated as primitive objects, it is still not simple to check collision between a circle and a ellipse, or a circle and a rectangle, as the distance to a ellipse can not be expressed as a closed form formulation [22] and the distance to a rectangle is not a linear function. The classic way to check the collision is adopting quadratic constraints, which is shown as,

$$\begin{gathered} \left\|\mathbf{p}_i^k - \mathbf{p}_j^k\right\|_\Omega = (\mathbf{p}_i - \mathbf{p}_o)^T \, \Omega \, (\mathbf{p}_i - \mathbf{p}_o) > 1, \quad i \in \mathcal{R}, \quad o \in \mathcal{O}, \\ \Omega = \mathrm{diag}(\frac{1}{(a+r)^2}, \frac{1}{(b+r)^2}). \end{gathered} \tag{4-1}$$

In order to normalize the collision constraints between different obstacles, we assume that every robot should work in its free space, which is shaped by all sur-

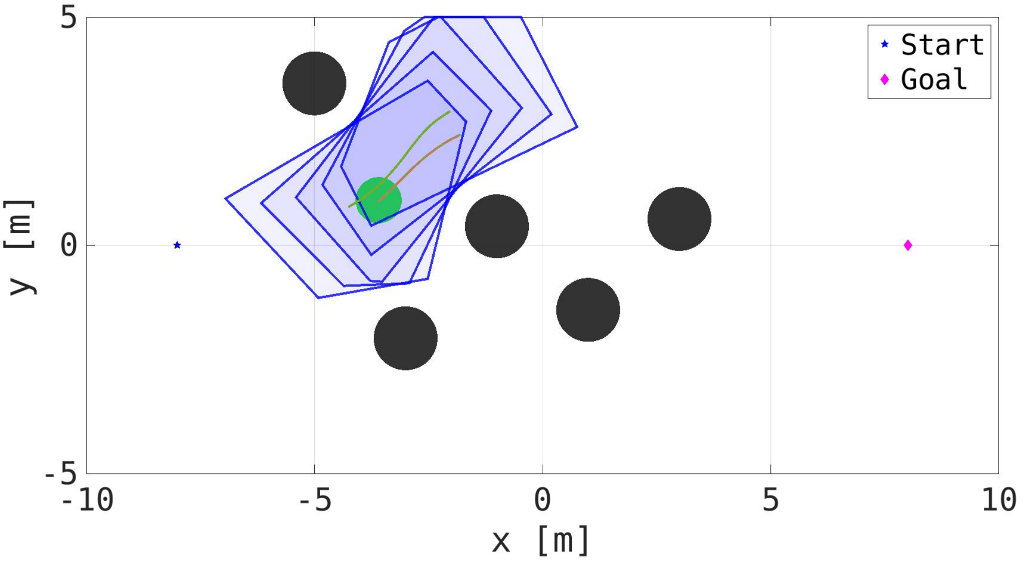


Figure 4-1: Linearized Collision Constraints of the MPC. The blue ploygons represents the free space in different time stages. The red curve is the planning result and the green curve is the path of the centroids of free space.

rounding static obstacles and moving agents. The distributed MPC adopts the planning trajectories in the last time step as the position of moving agents. Therefore, the collision constraints turn into a set of linear constraints. The linearization of quadratic constraints between a circle and ellipse can be expressed as a linear half-space $A_{io}\mathbf{p}_i \leq b_{io}, i \in \mathcal{R}, o \in \mathcal{O}$, whose boundary is tangent to the obstacle ellipse. The formulation of A and b can be expressed as,

$$\begin{aligned} &A_{io} = {A_{io}^{k}}^{T} \Omega M(\gamma), \quad b_{io} = b_{io}^{k}, \\ &A_{io}^{k} = \frac{\mathbf{p}_i^{tr} - \mathbf{p}_o^{tr}}{\|\mathbf{p}_i^{tr} - \mathbf{p}_o^{tr}\|}, \quad b_{io}^{k} = {A_{io}^{k}}^{T} \left(\mathbf{p}_o^{tr} - A_{io}^{k}\right), \\ &\mathbf{p}_i^{tr} = \Omega M(\gamma)\hat{\mathbf{p}}_i, \quad \mathbf{p}_j^{tr} = \Omega M(\gamma)\hat{\mathbf{p}}_j, \\ &M(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix}, \end{aligned} \tag{4-2}$$

where, $\mathbf{p}_i^{tr}$ and $\mathbf{p}_o^{tr}$ are the normalized position of robots and obstacles and $M(\gamma)$ is the rotation matrix by the orientation of the obstacle. As for rectangle obstacles, we suppose that obstacles are surrounded by closed ellipses with width $2\sqrt{2}a$, height $2\sqrt{2}b$. For other agents in the workspace, the constraints can also be transformed into $A_{ij}\mathbf{p}_i \leq b_{ij}, i \in \mathcal{R}, j \neq i \in \mathcal{R}$. And the conditions that the robot stays in its free space can be expressed as,

$$A_i\mathbf{p}_i \leq b_i, \quad A_i = [A_{io}, A_{ij}], \quad b_i = [b_{io}, b_{ij}]. \tag{4-3}$$

Dynamics Equality

The dynamics model of each robot i can be formulated as a discrete-time equation,

$$\mathbf{x}_i^{k+1} = \begin{bmatrix} \mathbf{p}_i^{k+1} \\ \mathbf{v}_i^{k+1} \end{bmatrix} = \begin{bmatrix} I & \Delta t \cdot I \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{p}_i^{k} \\ \mathbf{v}_i^{k} \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta t \cdot I \end{bmatrix} \mathbf{u}_i^k, \quad \mathbf{x}_i^0 = \hat{\mathbf{x}}_i^0, \tag{4-4}$$

where, $\mathbf{x}_i^k \in \mathbb{X} \subset \mathbb{R}^{n_x}$ is the states of the robots, including the positions and velocities, $\mathbf{u}_i^k \in \mathbb{U} \subset \mathbb{R}^{n_u}$ is the inputs of the system, $\mathbf{x}_i^0$ is the initial state of robots, I is the identity matrix and Δt is the sampling time. All the robots in the experiments are assumed to have the same dynamics model.

Objective Function

The objective of the multi-robot motion planning is to find a control input $\mathbf{u}_i^k$ for each robot i at discrete time step k. By following the control input, the robots can find an available trajectory to get closer to the goal $\mathbf{g}_i^k$ position while following the dynamic model and avoiding collision with other objects in the workspace in the next time horizon τ.

The baseline of the objective function can be shown as,

$$\min_{\mathbf{x}_i^{1:N}, \mathbf{u}_i^{0:N-1}} \sum_{k=0}^{N-1} J_i^k \left(\mathbf{x}_i^k \right) + J_i^N \left(\mathbf{x}_i^N, \mathbf{g}_i \right), \tag{4-5}$$

where, J_i^k is the stage cost of robot i at time k, J_N^k is the terminal cost of robot i at time N and $N = \tau / \Delta t$ is the number of time steps of MPC, which is time horizon.

The stage cost is a collision potential cost function, which guides the robot to keep its distance from other agents and static obstacles, as the equation 4-6 shows,

$$\begin{aligned}
J_i^k \left(\mathbf{x}_i^k \right) &= \sum w_c \left(C_{il}^k + C_{ie}^k \right), \\
C_{il}^k &= \begin{cases} \sum_{l=1}^{m+n-1} \left(1 - \left\| \mathbf{p}_i^k - \mathbf{p}_l^k \right\|_{\Omega_{\text{ex}}} \right), & \left\| \mathbf{p}_i^k - \mathbf{p}_l^k \right\|_{\Omega_{\text{ex}}} < 1, \\ 0, & \left\| \mathbf{p}_i^k - \mathbf{p}_l^k \right\|_{\Omega_{\text{ex}}} \geq 1, \end{cases} \\
\Omega_{ex} &= \operatorname{diag} \left(\frac{1}{(\lambda_l a + \lambda_r r)^2}, \frac{1}{(\lambda_l b + \lambda_r r)^2} \right), \\
C_{ie}^k, &= \begin{cases} \sum_{e=1}^{4} \left(1 - \frac{\mathbf{d}_{ie}^{k\,T} \mathbf{d}_{ie}^k}{d_{et}^2} \right) & \left\| \mathbf{d}_{ie}^k \right\| < d_{et}, \\ 0, & \left\| \mathbf{d}_{ie}^k \right\| \geq d_{et}, \end{cases} \\
& i \in \mathcal{R}, \quad l \in (\mathcal{R} \bigcup O \setminus i), \quad e \in \mathcal{E},
\end{aligned} \tag{4-6}$$

where, C_{il}^k is the sum of collision potential cost of all objects in the workspace, which include all static obstacles and other moving agents. C_{ie}^k is the collision potential cost of workspace boundary $\mathcal{E}$. w_c is the weight coefficient of the collision cost, Σ_{ex} is the extended distance metric, λ_l and λ_r is the extended parameter of obstacles and robots. Only if the robot locates in the extended region $\{\lambda_l a + \lambda_r r, \lambda_l b + \lambda_r r\}$ around obstacles, the cost will make a difference to the planning result. The value of λ_l and λ_r decide the area of extended regions, which are always larger than 1. $\mathbf{d}_{ie}^k$ is the vector to each boundary of the workspace. d_{et} is the threshold distance of collision potential cost to the boundaries. when $\left\|\mathbf{d}_{ie}^k\right\|$ is smaller than d_{et}, the collision potential will be computed, otherwise it the collision potential cost is 0.

The terminal cost is the goal navigation cost, which evaluates how a robot is close to the goal position, as the equation shows,

$$J_i^N = w_g \frac{\left(\mathbf{p}_i^N - \mathbf{g}_i\right)^T \left(\mathbf{p}_i^N - \mathbf{g}_i\right)}{\left\|\mathbf{p}_{i_p}^N - \mathbf{g}_i\right\|}, \tag{4-7}$$

where, w_g is the weight coefficient of the goal navigation cost, $\mathbf{d}_{i_p}^N$ is the terminal position of the planning trajectory in the last time step. Moreover, for the first time step, $\mathbf{d}_{i_p}^N$ is equal to the initial position. Dividing the distance to the goal by the previous distance normalizes the value of cost around 1, which is more convenient when designing the weight coefficient.

Baseline Formulation

The completed formulation of multi-robot MPC is shown as 4-8. The baseline of multi-robot MPC only considers the simple collision risk, which is not enough to minimize various risks in planning. In the next part, we will introduce how to adopt risk assessment into the baseline.

$$\begin{aligned}
&\min_{\mathbf{x}_i^{1:N}} \sum_{k=1}^{N-1} J_i^k\left(\mathbf{x}_i^k\right) + J_i^N\left(\mathbf{x}_i^N, \mathbf{g}_i\right) \\
&s.t. \quad \mathbf{x}_i^0 = \hat{\mathbf{x}}_i^0, \\
&\qquad \mathbf{x}_i^k = f\left(\mathbf{x}_i^{k-1}, \mathbf{u}_i^{k-1}\right), \\
&\qquad A_i \mathbf{p}_i \leq b_i, \quad A_i = [A_{io}, A_{ij}]^T, \quad b_i = [b_{io}, b_{ij}]^T, \quad \mathbf{x}_i^k = [\mathbf{p}_i^k, \mathbf{v}_i^k]^T, \\
&\qquad \mathbf{u}^{k-1} \in \mathcal{U}, \quad \mathbf{x}^k \in \mathcal{X}, \quad \forall k \in \{1, \ldots, N\}, \\
&\qquad \forall i \in \mathcal{R} = \{1, 2, ..., n\}, \forall j \neq i \in \mathcal{R}, \forall o \in \mathcal{O} = \{1, 2, ..., m\}.
\end{aligned} \tag{4-8}$$

4-1-2 Risk-Aware Multi-Robot Motion Planning

Compared to classic motion planning, risk-aware motion planning will involve risk elements in the MPC formulation. As we discussed in section 2-3, chance constraints and risk function are two assessments of risk in motion planning. However, the constraints can only restrict the risk under a pre-defined threshold. Involving risk functions into objective functions in MPC can minimize risk while planning, ensuring that the gradient of objective functions is always in the direction of the lower risk. Therefore, we designed two groups of experiments to test the performance of the risk elements in MPC.

Collision Risk

The baseline of multi-robot motion planning has already contained a simple collision risk cost function C_{il}^k. This function is the promotion of sum of distance Σ_d that we discussed in section 3-1. Compared with the origin, C_{il}^k considers the obstacle density by adopting extended regions around obstacles. To compare the performance of the risk elements **Deformed Distance to the centroid of Free Space**, we introduce our collision risk function, as the equation 4-9 shows,

$$J_i^k = \begin{cases} \frac{(\mathbf{p}_{\mathrm{r}} - \mathbf{p}_{\mathrm{centr}})^T(\mathbf{p}_{\mathrm{r}} - \mathbf{p}_{\mathrm{centr}})}{S_{\mathrm{Free}}}, & \mathbf{p}_{\mathrm{center}} \notin \mathbb{F}, \\ \frac{(\mathbf{p}_{\mathrm{r}} - \mathbf{p}_{\mathrm{goal}})^T(\mathbf{p}_{\mathrm{r}} - \mathbf{p}_{\mathrm{goal}})}{S_{\mathrm{Free}}}, & \mathbf{p}_{\mathrm{center}} \in \mathbb{F}, \end{cases} \tag{4-9}$$

where, $\mathbb{F}$ is the reformed free space region that we defined in Section 3-1. When the goal position locates in the free space, the potential to get closer to the deformed centroid and get closer to the goal may conflict with each other, resulting in deadlock in the middle of two positions. Thus, the distance to the goal will replace the distance to the centroid, as the equation 4-9 shows.

A challenge of this method is to compute the area of free space because solving the area of the free space is a linear searching problem. Therefore, It is assumed that free space between neighboring time steps will keep similar under a high sampling rate. The area of free space in the last time step could represent the current area approximately. The area and the centroid of free space need to be computed once a stage before optimization, and the area will not be optimized in MPC. In this way, the objective function becomes quadratic, and the time-consuming in planning can be highly reduced.

Another collision element that we discussed in Section 3-1 is the *Clearance*. Similar to collision potential cost, the clearance can be formulated as 4-10 in MPC. The maximum distance to the obstacles or the boundaries of the workspace represents the clearance value, and only the closed distance will make a difference in planning.

$$\begin{aligned} J_i^k\left(\mathbf{x}_i^k\right) &= w_c \max\left(M_{il}^k, M_{ie}^k\right), \_i \in \mathcal{R}, \_l \in (\mathcal{R} \bigcup O \setminus i), \_e \in \mathcal{E}, \\ M_{il}^k &= \begin{cases} 1 - \left\|\mathbf{p}_i^k - \mathbf{p}_l^k\right\|_{\Omega_{\mathrm{ex}}}, & \left\|\mathbf{p}_i^k - \mathbf{p}_l^k\right\|_{\Omega_{\mathrm{ex}}} < 1, \\ 0, & \left\|\mathbf{p}_i^k - \mathbf{p}_l^k\right\|_{\Omega_{\mathrm{ex}}} \geq 1, \end{cases} \\ M_{ie}^k &= \begin{cases} 1 - \frac{{\mathbf{d}_{ie}^k}^T \mathbf{d}_{ie}^k}{{\mathbf{d}_{et}}^T \mathbf{d}_{et}}, & \left\|\mathbf{d}_{ie}^k\right\| < d_{et}, \\ 0, & \left\|\mathbf{d}_{ie}^k\right\| \geq d_{et}, \end{cases} \end{aligned} \tag{4-10}$$

where, M_{il}^k and M_{ie}^k are the deformed distance of a single object l or a boundary e. One shortage of clearance in MPC is the $\max$ function. The gradients of the objective functions in an optimization problem decide the searching direction. However, it is impossible to compute the gradient of the $\max$ function mathematically. Therefore, the optimizer is difficult to converge to the optimal value when two distances are close.

Congestion Risk

In the scope of this article, two collision risk elements, Potential to Goal and Risk Level Set, are adopted into MPC to show the influence of collision risk elements. The formulation of potential to goal in MPC is similar to $R(p_i, \theta_{ig}, p_j, p_g, v_j)$ in section 3-2. The difference is that the parameter of distance to goal is transformed into the terminal cost of the baseline, which is shown as equation 4-11,

$$\begin{aligned} J_i^N =& w_g \left(P2G + 1\right) \frac{\left(\mathbf{p}_i^N - \mathbf{g}_i\right)^T \left(\mathbf{p}_i^N - \mathbf{g}_i\right)}{\left\|\mathbf{p}_{i_p}^N - \mathbf{g}_i\right\|}, \\ P2G =& \sum_{j=1}^{m} \left[\frac{1}{2m} \frac{cos\left(\theta_{ig}^N\right) + 1}{1 + \exp(-\beta \mathbf{v}_j^{N^T} * (\mathbf{p}_i^N - \mathbf{p}_j^N))} \right], \\ cos\left(\theta_{ig}^N\right) =& \frac{\left(\mathbf{p}_j^N - \mathbf{p}_i^N\right)^T \left(\mathbf{p}_g - \mathbf{p}_j^N\right)}{\left\|\mathbf{p}_j^N - \mathbf{p}_i^N\right\| \left\|\mathbf{p}_g - \mathbf{p}_j^N\right\|}. \end{aligned} \tag{4-11}$$

The robots will prefer to finish the paths in a shorter time without congesting together, which increases the average velocities at the same time. Another type of congestion risk application is *Risk Level Set* that we discussed in Section 3-2, as the equation 4-12 shows,

$$J_i^k = \sum_{i=1}^{n} \frac{\exp\left(-\left(\mathbf{p}_i^k - \mathbf{p}_j^k\right)^T \left(\mathbf{p}_i^k - \mathbf{p}_j^k\right)\right)}{1 + \exp\left(-\alpha \mathbf{v}_j^{k^T} \left(\mathbf{p}_i^k - \mathbf{p}_j^k\right)\right)}. \tag{4-12}$$

Risk Level Set is formulated as a twisted collision potential field and a combination of congestion and collision risk elements. The skewing parameter is similar to *Potential to Goal*. By skewing the peaks of potential fields to the directions of obstacle velocities, the objective function gradient becomes lower on the risky region, and the robots will choose to pass by the obstacles from the opposite directions of their speeds.

These two applications of congestion risk in MPC reflect the formulations of congestion risk elements in terminal navigation cost and stage collision cost. In the following experiments, we will test the influence of the two applications in cluttered environments.

4-2 Simulation Setup

To test the influences of different applications of risk in multi-robot motion planning, we built up a simulation platform in Matlab 2017b in a computer with an Intel(R) Core(TM) i5-7360U CPU @ 2.30GHz CPU. The structure of the MPC is based on the work of [29], which relies on Forces Pro to [7] generate NMPC code. The objective functions and inequality collision constraints are refactored to meet the requirements of the MPC in section 4-1.

In both groups of simulations, the sampling time is $\Delta t = 0.1s$, and the time horizon of the MPC is $N = 20$. The radius of the enclosing circle of the robots is $0.5m$. The initial guess for each time step is based on the previous planning result. When the result is invalid, like reaching the maximum iteration, a constant velocity model will be adopted to predict the robot trajectories. The maximum iteration is defined as 800, which is high enough to find optimal value in practice. The initial velocities are all set as 0. The uncertainties of robot positions and velocities are considered in the experiments, representing the localization and sensor error in practice. The testing scenarios will be introduced in section 4-3 and 4-4. Each scenario consists of 50 samples.

4-3 Performance of Collision Risk Elements

Three scenarios, circle obstacle group, narrow corridor, and random corridor, are generated to show the performance of collision risk elements in motion planning. Robots are tested in closed workspaces in all scenarios and will start and end at the same initial and goal positions.

- **Circle Obstacle Group**: As Fig. 4-2 show, the robots are tested in a 20×20m workspace. The initial position is at (-8, 8), and the goal is at (8, -8). The static obstacles locate on the vertices and the center of a pentagon. The sizes and orientations of obstacles and the size of the pentagon are generated randomly, which confirms enough free space between obstacles so that the robots may choose to go around the group but not pass through.
- **Narrow Corridor**: As Fig. 4-3 show, the robots are tested in a 20×10m rectangle workspace. The initial position is at (-8, 0), and the goal is at (8, 0). Four static obstacles locate on the side of the corridor, and the sizes and orientations of obstacles are generated randomly. The shape of the obstacle is more extreme than the previous obstacle group, which will lead to narrow passages in large part of scenarios. However, the passage is wide enough space for the robot to pass through. As the shapes of the passages differ a lot between different scenarios, the robots may deadlock in unexpected space.
- **Random Corridor**: As Fig. 4-4 show, similar to narrow corridor, the robots are tested in a 20×10m rectangle workspace. The initial position is at (-8, 0), and the goal is at (8, 0). Five static circle obstacles with a radius of 0.5m

are placed in the corridor area randomly. The distances between obstacles are mostly more comprehensive than the other scenarios. Therefore, the collision risk is much smaller.

The basic structure of the there MPC controllers is the baseline. All MPC controllers contain the essential terminal navigation cost. By combing different collision risk elements, the robots will show different trajectories to avoid collision in the workspace. The weights of objective functions are similar, ensuring the cost values are close to each other. Also, the weights are set reasonably so that the robot can finish the path in most scenarios.

- **Baseline+PF (PF)**: The baseline of multi-robot MPC. It is the basic formulation of collision risk in motion planning
- **Baseline+Clearance (Clear)**: *Clearance*, shown as equation 4-10, replaces the collision pential field of the baseline. Only the closed distance to obstacles will influence the planning result
- **Baseline+Dis2Centr (Centr)**: *Distance to the Centroid of Free Space*, shown as equation 4-9, replaces the collision pential field of the baseline.

Although the weight of objective functions is adjusted to make sure the robot can pass through most passages, it is still possible that the robots get deadlock. Therefore, the number of deadlocks and the number of collisions are recorded. The clearance, which is the minimal distance to the obstacles, is the most important metric to evaluate collision risk. The average trajectory lengths evaluate the efficiency of the robots finishing the task. The summary of the simulation results are shown as table 4-1. The Fig. 4-2 to 4-4 shows the trajectories of the robots from each type of scenarios by each controller.

As the simulation result shows, the controller *Dis2Centr* performs best compared with others. By adopting *Dis2Centr*, The robots keep appropriate distances to obstacles, which shows the lowest collision risk in all scenarios. Moreover, in the scenarios narrow corridor and random corridor, the number of collisions is much smaller when the MPC adopts *Dis2Centr*. However, *Dis2Centr* will lead to more deadlock as the centroid of the free space lays opposite to the goal in some scenarios. Varying the size of the feasible region can release the problem. Also, the trajectories of *Dis2Centr* are longer than the others in the random corridor scenario, as the robots prefer to go around the crowded obstacles but not pass through the narrow passages, as the Fig. 4-4 shows. The performance of *Clear* is similar to *PF* in cluttered environments. However, the number of iterations of *Clear* is much higher than the other controllers, especially in a less risky environment. As we discussed in section 4-1, the stage cost function of *Clear* is non-differentiable. When the robots are located in the middle of two obstacles, the optimizer may spend higher iterations, even over the maximum, to converge to the optimal value. Too many iterations will lead to a longer planning time. Therefore, *Clear* is only suitable for low sampling rate scenarios.

Scenarios	**Motion Planner**	**Num. of Colli.**	**Num. of Deadlock**	**Clearance (m)**			**Traj. Length (m)**
				max	min	avg	
Circle Obs. Group	Baseline+PF	0	0	0.70	0.06	0.49	23.49
	Baseline+Clearance	0	0	0.66	0.09	0.49	23.63
	Baseline+Dis2Centr	**0**	**4**	**0.86**	**0.18**	**0.57**	**24.08**
Narrow Corridor	Baseline+PF	2	3	1.33	0.09	0.71	16.54
	Baseline+Clearance	2	2	1.38	0.08	0.73	16.62
	Baseline+Dis2Centr	**0**	5	**1.94**	**0.03**	**0.98**	**17.21**
Random Corridor	Baseline+PF	1	0	1.07	0.04	0.61	17.12
	Baseline+Clearance	1	0	1.08	0.05	0.61	17.11
	Baseline+Dis2Centr	**0**	**2**	**1.45**	**0.23**	**0.69**	**17.87**

Table 4-1: Comparison of performance of 3 MPC motion planners (Baseline+PF, Baseline+Clearance, Baseline+Dis2Centr) in 3 different scenarios (circle obstacle group, narrow corridor, and random corridor). For each scenario, 50 different test samples have been generated

4-4 Performance of Congestion Risk Elements

Four different types of scenarios are generated to show the performance of congestion risk elements in motion planning, symmetric swapping, asymmetric swapping, pair-wise swapping, and random. Six robots are tested in a 12×12m space, whose initial and goal positions differ among test samples.

- **Symmetric Swapping**: As Fig. 4-5 show, The robots locate on the vertices of a regular hexagon at the beginning and swap their positions with the ones on the opposite side of the origin. In different test samples, the side lengths of regular hexagons are equal to 4m, and the regular hexagons rotate random angles around the origin. As all the robots will pass around the origin to their goals, they will possibly congest and collide with each other.

- **Asymmetric Swapping**: As Fig. 4-6 show, the robots locate on the vertices of a random irregular hexagon at the beginning, and the opposite robots are symmetric about the origin, which will swap their positions. Compared to symmetric swapping, the distances to the origin are different initially, which will lead to more challenging congestion problems.

- **Pair-wise Swapping**: As Fig. 4-7 show, the six robots are separated into three groups and swap their positions. The positions of the three groups are generated randomly, and the paths of robots may interact with each other. Thus, the test is different from three individual pair-wise swappings without interaction. The robots are less likely to congest in this scenario than asymmetric swapping as all six robots are not symmetric about the same position.

- **Random**: As Fig. 4-8 show, the goal and initial positions of the robots are generated randomly. The congestion is unlikely to happen in most scenarios.

The basic structure of the five MPC controllers is the baseline. By adopting the risk elements in objective functions, the robots will show different congestion and collision avoidance performances. The following is the explanation of the five motion planner in Section 4-1. The weights of objective functions are equal to each other, and the whole cost values are close enough. Therefore the influence of weights can be minimized in planning.

- **Baseline-PF**: The baseline of multi-robot MPC without collision potential cost. Although the collision potential cost mainly shows the collision risk, it can release the congestion risk in practice.
- **Baseline**: The baseline of multi-robot MPC. The collision potential field and original terminal navigation cost are adopted. Here we propose potential risk but not distance to the centroid of free space as the collision risk assessment, as RLS also involves collision potential field to control collision risk.
- **Baseline+RLS**: The baseline of multi-robot MPC. Moreover, the collision potential cost is replaced by RLS. It shows the performance of different congestion risk assessments.
- **P2G**: Only the potential are adopted as terminal objective functions. Only congestion risk is evaluated in the test.
- **P2G+PF**: The potential to goal replace the objective functions in the baseline, and the collision potential field keeps the same. Both collision and congestion risks are assessed in this controller.

The number of simulations that are not complete the test is recorded as a numerical metric to evaluate the performance of robots. The trajectory time shows the performance of the controllers controlling congestion risk. The average speed of the robots is also analyzed to reflect the congestion situations of a simulation. To numericalize the control performance of collision risk, I also compute the clearance, which is the minimal distance between any two robots, in a simulation. The summary of the simulation results are shown as table 4-2. The Fig. 4-5 to 4-8 shows the trajectories of the robots from each type of scenarios by each controller.

The result of table 4-2 shows that the combination of potential to goal and collision potential field *P2G+PF* performs best, especially in the cluttered scenarios. *P2G+PF* balances the control of congestion and collision risk, which shorten the trajectory time without collisions. High average speed and short average time consumption represent excellent performance in controlling congestion risk. In the symmetric and asymmetric swapping scenarios, Our controller *P2G+PF* shows the highest average velocities and lowest trajectory time. Although the time consuming of *baseline+RLS* is also acceptable, the minimal distance is much less than *P2G+PF* and also *baseline*. Collisions are avoided completely in all test samples when using *Baseline* and *P2G+PF*, while *baseline+RLS* produces 4 and 7 collisions. It shows that RLS scarifies the collision risk to release the congestion, leading to high collision risk in cluttered environments. In less risky scenarios, pair-wise swapping

Scenarios	Motion Planner	Num. of Colli.	Num. of Deadlock	Avg. vel. (m/s)	Traj. time (s)			Clearance (m)
					max	min	avg	
Symmetric Swapping	Baseline-PF	24	0	0.55	31.40	11.80	17.41	0.05
	Baseline	0	7	0.66	15.90	13.90	15.00	0.30
	Baseline+RLS	4	0	0.61	36.60	11.20	13.34	0.08
	P2G	22	0	0.63	21.60	11.20	15.09	0.10
	P2G+PF	**0**	**3**	**0.72**	**14.80**	**11.20**	**12.55**	**0.46**
Asymmetric Swapping	Baseline-PF	30	2	0.59	39.70	11.50	15.90	0.04
	Baseline	0	2	0.67	24.60	11.50	14.47	0.37
	Baseline+RLS	7	1	0.70	27.40	10.50	13.16	0.08
	P2G	3	1	0.66	27.30	11.00	13.86	0.13
	P2G+PF	**1**	**0**	**0.72**	**14.90**	**10.80**	**12.63**	**0.13**
Pair-wise Swapping	Baseline-PF	14	0	0.60	13.10	7.40	9.93	0.04
	Baseline	0	1	0.59	17.00	8.10	11.04	0.23
	Baseline+RLS	6	0	0.60	13.00	7.90	10.20	0.06
	P2G	1	0	0.59	28.30	7.80	10.70	0.12
	P2G+PF	**0**	**0**	**0.60**	**11.00**	**7.90**	**10.94**	**0.17**
random	Baseline-PF	14	1	0.56	16.90	8.30	10.29	0.05
	Baseline	0	0	0.55	15.80	8.40	11.08	0.26
	Baseline+RLS	2	0	0.57	16.00	8.60	10.40	0.11
	P2G	0	0	0.55	18.80	8.50	10.96	0.21
	P2G+PF	**0**	**0**	**0.56**	**11.80**	**9.30**	**10.53**	**0.23**

Table 4-2: Comparison of performance of 5 MPC motion planners (Baseline-PF, Baseline, Baseline+RLS, P2G and P2G+PF) in 4 different scenarios (symmetric swapping, asymmetric swapping, pair-wise swapping and random). For each scenario, 50 different test samples have been generated

and random scenarios, collisions and deadlock rarely happen when using *P2G*. The controller *P2G+PF* and *baseline* spend more time to complete the test, as minimizing the collision risk of the objective functions will result in longer trajectories and higher time-consuming. This result is acceptable because a different trajectory is required to keep robots in the distance, leading to a longer time.

4-5 Uncertainty of Robots in Risk-Aware Motion Planning

In the previous two tests, the distributions of robot positions and velocities are $\tau_p = 0.06$ and $\tau_v = 0.01$, which represent the navigation uncertainties in experiments. As the estimated positions and velocities deviate from their real states, the planning result may lead to collision when the robots are closed to the obstacles.

In this section, I focus on the influence of navigation uncertainties on the planners. The testing scenarios of collision and congestion risk elements are the same as the random corridor and the asymmetric swapping. The distributions of robot positions and velocities are defined as,

- $\tau_p = 0.1$ $\tau_v = 0.015$,
- $\tau_p = 0.06$ $\tau_v = 0.01$,
- $\tau_p = 0.03$ $\tau_v = 0.005$,

which represent the high to low navigation uncertainties. The robots are tested in 50 different samples. The summary of the simulation results are shown as the tables 4-3 and 4-4.

In the experiments of collision risk elements, the navigation uncertainties increase the collision numbers. In all three groups, the average clearance of *Centr* is higher than the others, which proves that the planner with *Centr* can assess collision risk better. The numbers of deadlocks keep almost the same. The obstacles are static in the experiments, and only one robot has uncertainties, which will make a slight difference to the congestion risk. The trajectory lengths increase when the uncertainties become serious in three groups, showing that the errors aggravate the efficiencies of planning results. In conclusion, the planners with *Centr* can evaluate the collision risk well under uncertainties. Increasing uncertainties will hardly influence the collision risk assessment in MPC.

Scenarios	**Motion Planner**	**Num. of Colli.**	**Num. of Deadlock**	**Clearance (m)**			**Traj. Length (m)**
				max	min	avg	
$\tau_p = 0.1$ $\tau_v = 0.015$	Baseline+PF	2	0	1.10	0.08	0.59	17.14
	Baseline+Clearance	1	0	1.11	0.04	0.59	17.12
	Baseline+Dis2Centr	**0**	**1**	**1.26**	**0.05**	**0.67**	**18.30**
$\tau_p = 0.06$ $\tau_v = 0.01$	Baseline+PF	1	0	1.07	0.04	0.61	17.12
	Baseline+Clearance	1	0	1.08	0.05	0.61	17.11
	Baseline+Dis2Centr	**0**	**2**	**1.45**	**0.23**	**0.69**	**17.87**
$\tau_p = 0.03$ $\tau_v = 0.005$	Baseline+PF	0	0	1.06	0.01	0.63	17.08
	Baseline+Clearance	0	0	1.08	0.01	0.64	17.07
	Baseline+Dis2Centr	**0**	**2**	**1.46**	**0.28**	**0.73**	**17.57**

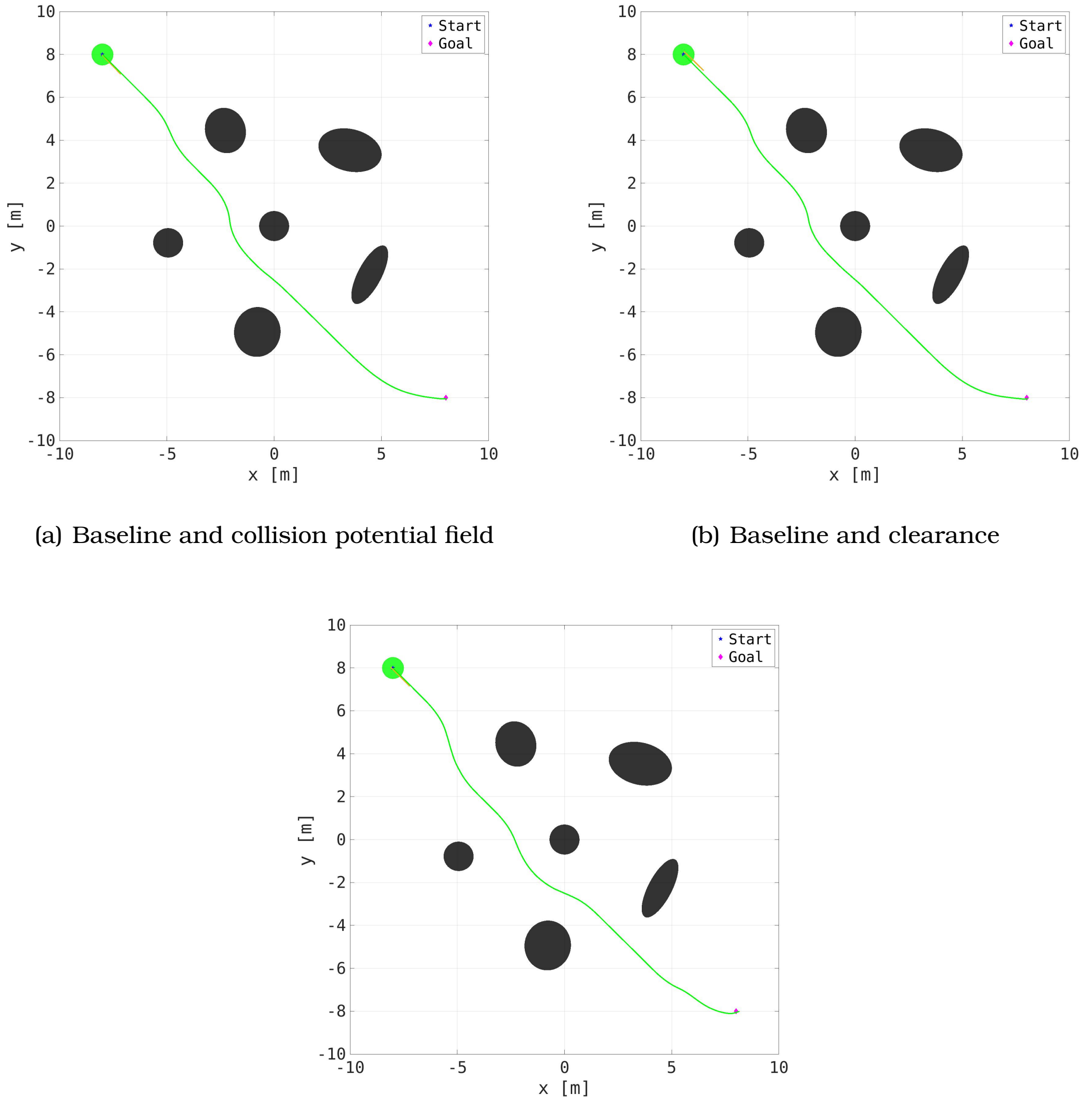
Table 4-3: Comparison of performance of 3 MPC motion planners (Baseline+PF, Baseline+Clearance, Baseline+Dis2Centr) in the random corridor scenario. The robots are under high, middle, and low navigation uncertainties. For each group, 50 different test samples have been generated

The results of the experiments of congestion risk elements are interesting. Similar to the collision risk elements, the number of collisions increases with the uncertainties. However, the number of deadlocks decreases with them. The symmetry of the robot positions causes most deadlocks in the experiments, and the uncertainties increase the randomness of the planning results, which weakens the symmetry. The velocities almost keep the same, but the trajectory time increase with the uncertainties. The clearance shows the same performance in the experiments of collision risk elements, which shows that the collision risk is also enhanced. The other planners have revealed their weaknesses when the uncertainties increase. The number

Uncertainty	**Motion Planner**	**Num. of Colli.**	**Num. of Deadlock**	**Avg. vel. (m/s)**	**Traj. time (s)**			**Clearance (m)**
					max	min	avg	
$\tau_p = 0.1$ $\tau_v = 0.015$	Baseline-PF	32	3	0.61	31.80	12.00	15.24	0.03
	Baseline	0	1	0.67	28.00	11.40	14.54	0.34
	Baseline+RLS	8	0	0.69	18.50	11.40	13.35	0.04
	P2G	9	0	0.65	21.90	11.50	14.20	0.11
	P2G+PF	**4**	**0**	**0.71**	**14.80**	**10.70**	**12.88**	**0.11**
$\tau_p = 0.06$ $\tau_v = 0.01$	Baseline-PF	30	2	0.59	39.70	11.50	15.90	0.04
	Baseline	0	2	0.67	24.60	11.50	14.47	0.37
	Baseline+RLS	7	1	0.70	27.40	10.50	13.16	0.08
	P2G	3	1	0.66	27.30	11.00	13.86	0.13
	P2G+PF	**1**	**0**	**0.72**	**14.90**	**10.80**	**12.63**	**0.13**
$\tau_p = 0.03$ $\tau_v = 0.005$	Baseline-PF	14	17	0.5	27.60	11.10	15.16	0.05
	Baseline	0	8	0.68	25.80	11.30	14.02	0.39
	Baseline+RLS	7	4	0.71	23.10	10.30	12.60	0.12
	P2G	2	5	0.66	29.60	10.90	14.13	0.12
	P2G+PF	**0**	**3**	**0.72**	**14.90**	**10.40**	**12.56**	**0.13**

Table 4-4: Comparison of performance of 5 MPC motion planners (Baseline-PF, Baseline, Baseline+RLS, P2G and P2G+PF) in the asymmetric swapping scenario. The robots are under high, middle, and low navigation uncertainties. For each group, 50 different test samples have been generated

of collisions is too high when the robot is controlled by the planner *RLS* under high uncertainties. The *Baseline* meets difficulty to overcome deadlock under low uncertainties. The result shows that the planner with *P2G+PF* can evaluate both collision and congestion risk in the multi-robot motion planning under uncertainties. The uncertainties in practice are more complex than the settings of experiments. Thus, the robots can finish the paths under lower risk with the planner with *P2G+PF*.

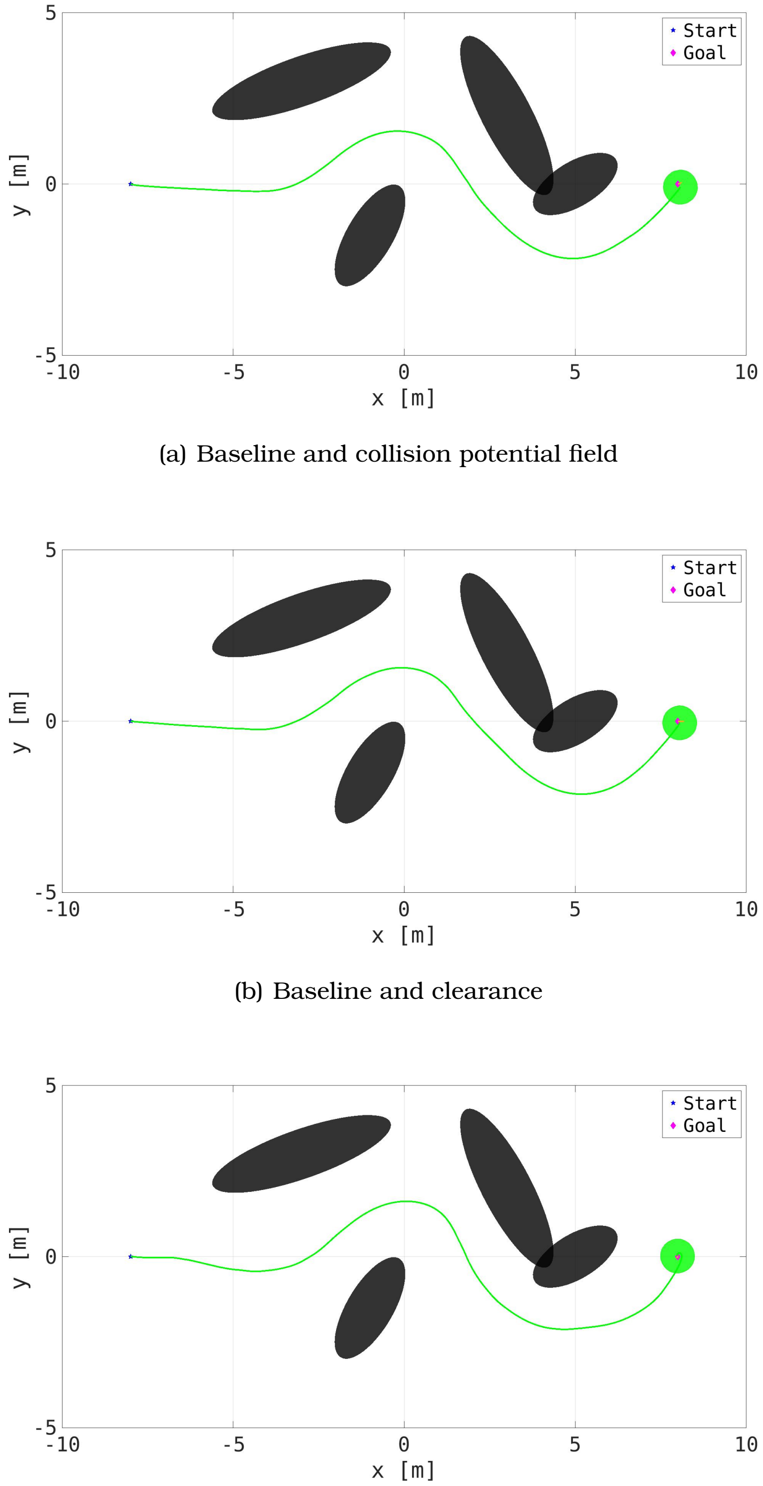


(a) Baseline and collision potential field

(b) Baseline and clearance

(c) Baseline and distance to centroid of free space

Figure 4-2: Planning results for one of the the circle obstacle group scenario. The blue star and the red diamond represent the initial and goal positions

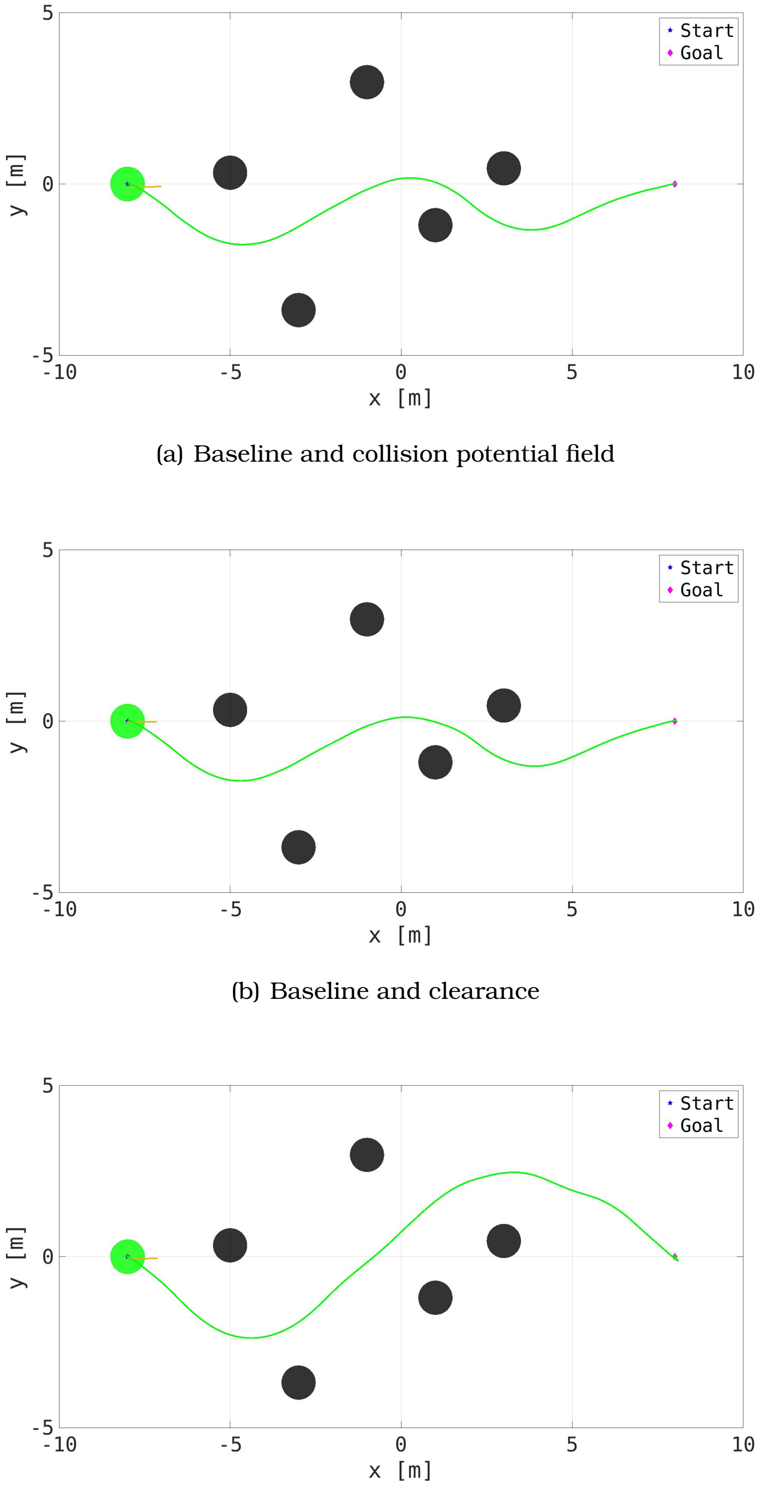


(a) Baseline and collision potential field

(b) Baseline and clearance

(c) Baseline and distance to centroid of free space

Figure 4-3: Planning result for one of the narrow corridor scenario. The blue star and the red diamond represent the initial and goal positions

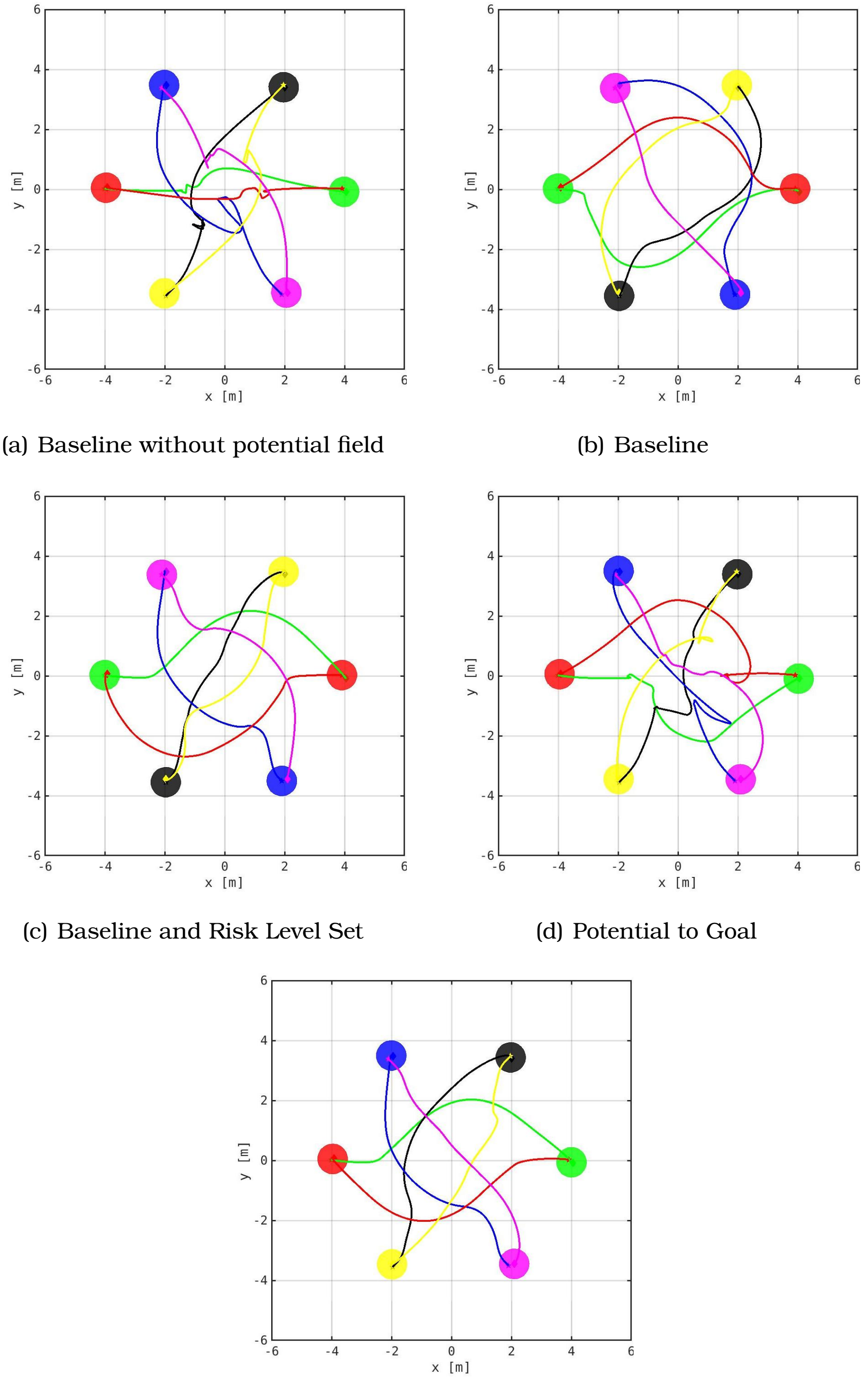


(a) Baseline and collision potential field

(b) Baseline and clearance

(c) Baseline and distance to centroid of free space

Figure 4-4: Planning result for one of the random corridor scenario. The blue star and the red diamond represent the initial and goal positions



(a) Baseline without potential field

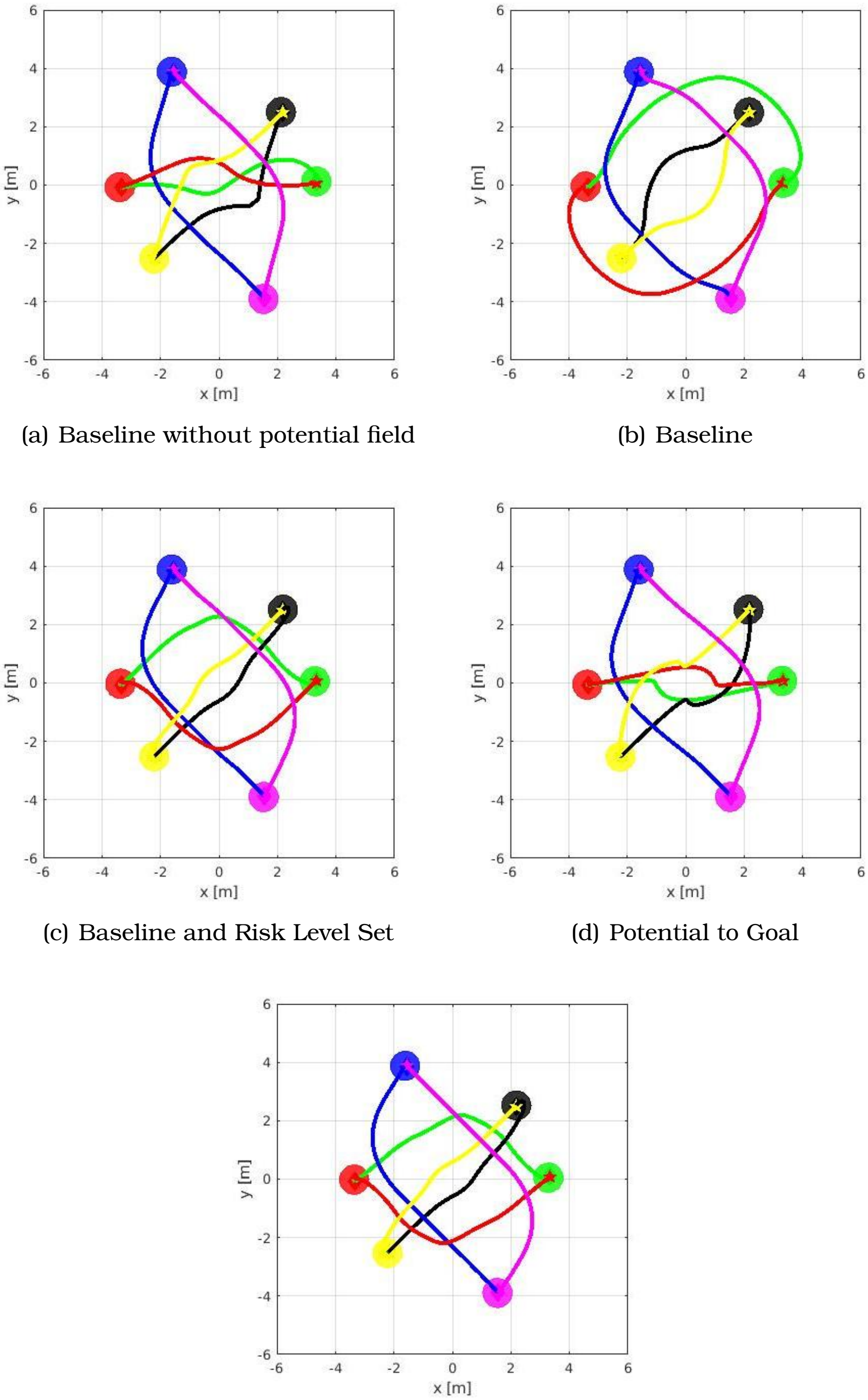
(b) Baseline

(c) Baseline and Risk Level Set

(d) Potential to Goal

(e) Baseline and Potential to Goal

Figure 4-5: Planning result for one of the symmetric swapping scenarios. The solid circles represent the goal positions of the robots and the trajectories with corresponding colors represent the planning results.



(a) Baseline without potential field

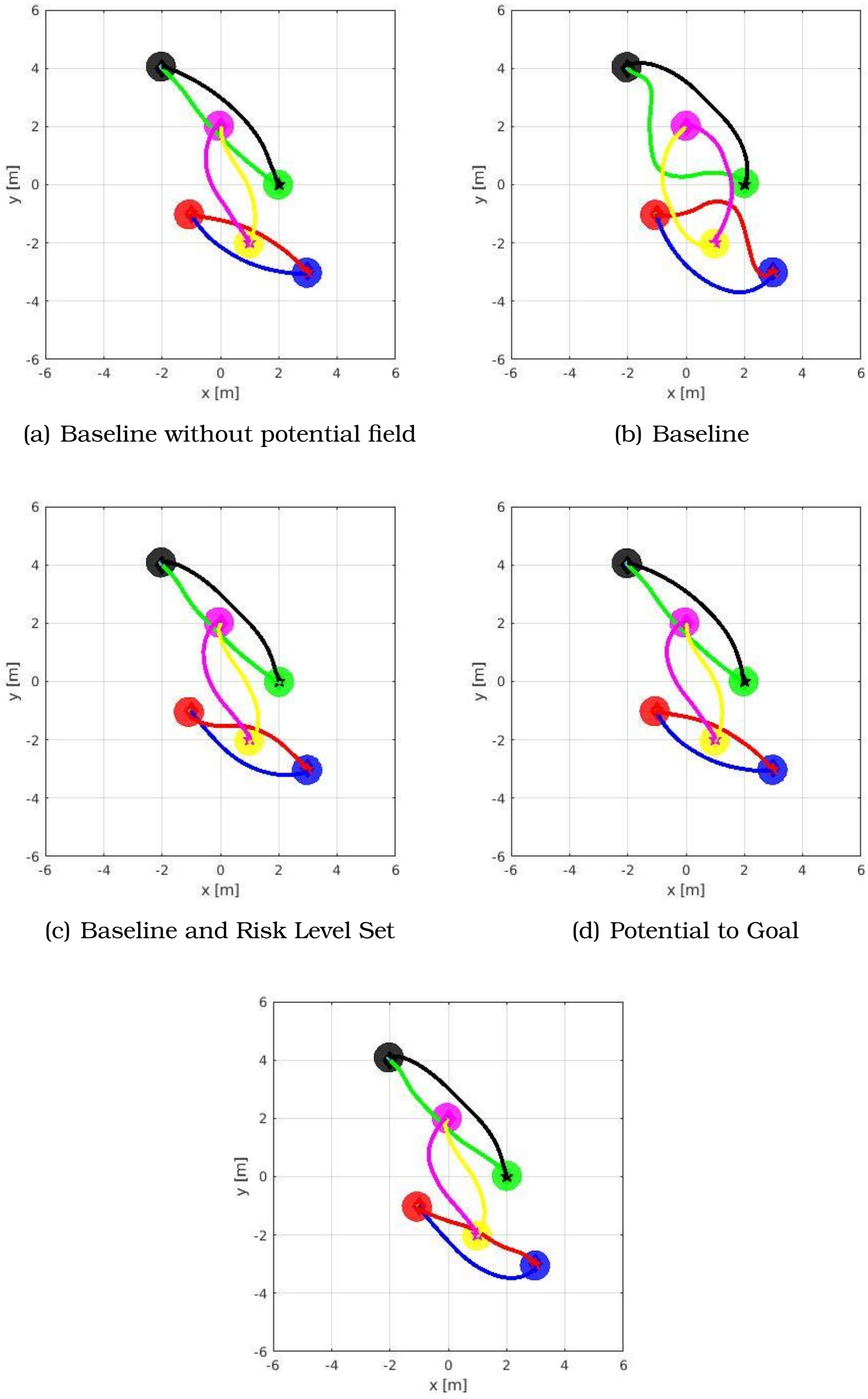
(b) Baseline

(c) Baseline and Risk Level Set

(d) Potential to Goal

(e) Baseline and Potential to Goal

Figure 4-6: Planning result for one of the asymmetric swapping scenarios. The solid circles represent the goal positions of the robots and the trajectories with corresponding colors represent the planning results.



(a) Baseline without potential field

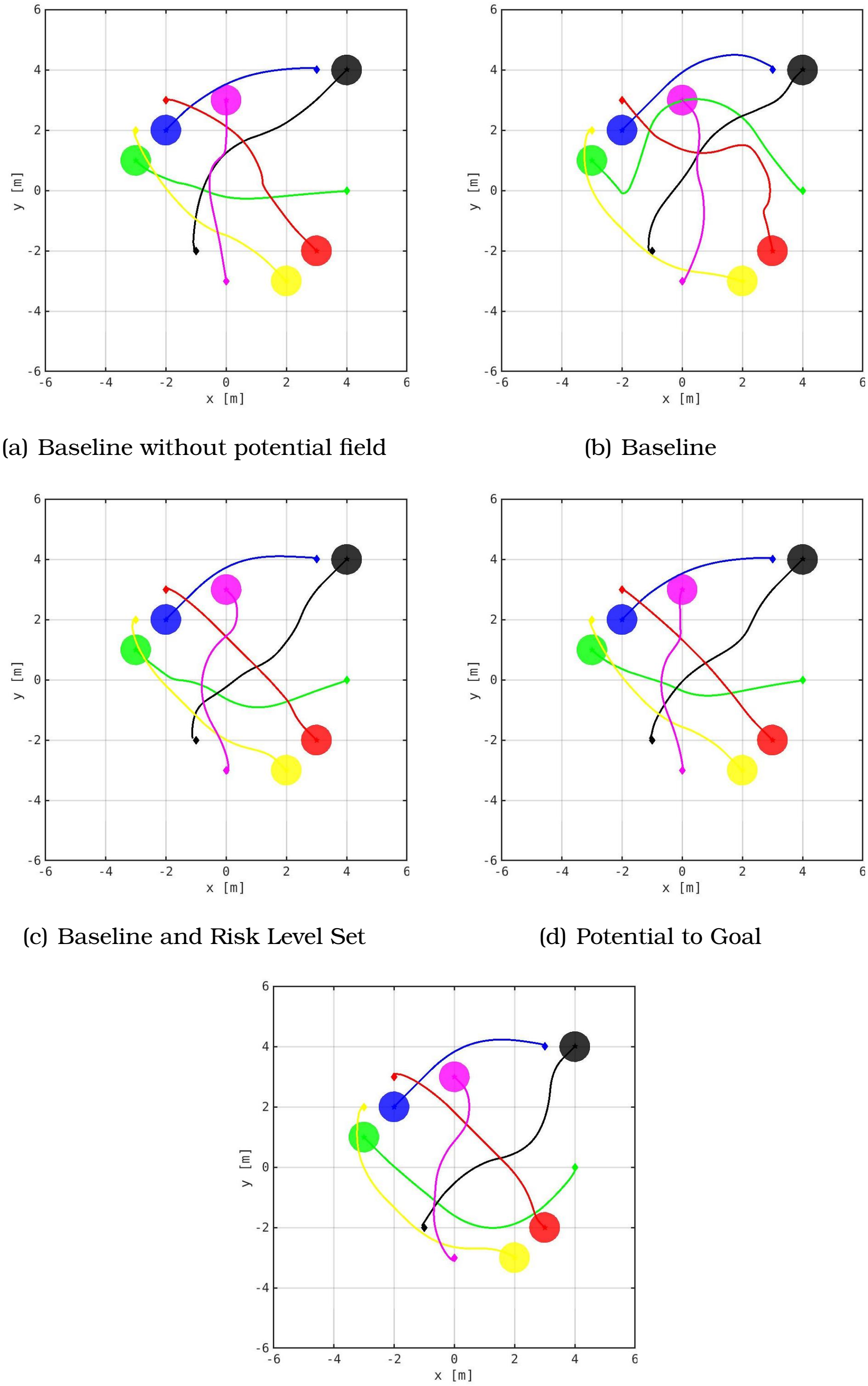
(b) Baseline

(c) Baseline and Risk Level Set

(d) Potential to Goal

(e) Baseline and Potential to Goal

Figure 4-7: Planning result for one of the pair-wise swapping scenarios. The solid circles represent the goal positions of the robots and the trajectories with corresponding colors represent the planning results.



(a) Baseline without potential field

(b) Baseline

(c) Baseline and Risk Level Set

(d) Potential to Goal

(e) Baseline and Potential to Goal

Figure 4-8: Planning result for one of the random scenarios. The solid circles represent the goal positions of the robots and the trajectories with corresponding colors represent the planning results.

Chapter 5

Experimental Validation

In this chapter, the proposed MPC planners are investigated through experiments. In chapter 4, the structure of the MPC planner is introduced to show the application of risk elements in motion planners. The performance of the proposed MPC planners in simulations shows that the robot can complete the task faster and smoother compared with other planners. The robustness of the controller under high uncertainty is also acceptable. To validate the performance of the controller, I will test my controller on quadrotor UAVs. In section 5-1, the experiment environments are set up, including the equipment and control structure. In section 5-3, the experiment results are discussed to show the performance of our controllers.

5-1 Experiment Setup

The experiment setup consists of several components, as the Fig. 5-1 shows. In the experiments, up to three quadrotors (4) are used, whose positions are measured by a motion capture system (1). The system consists of 14 Optitrack Prime 17W cameras and is controlled by the software Motive on a Windows computer (2). The system sends the positions and orientations of the robots to the state estimator, which will compute the estimated positions and velocity of robots. The estimated states are regarded as the inputs of the distributed local multi-robot motion planners in section 4-1. The local motion planner is executed in Matlab using Forces Pros (3). The state estimator and the local motion planners are implemented on a Linux laptop. The velocity commends solved by the planner are sent to the quadrotors wirelessly. The internal controllers of the robots will execute the commands. All the components communicate through the Robot Operating System (ROS), and an Xbox360 handle is used to control the takeoff and landing of the quadrotors.

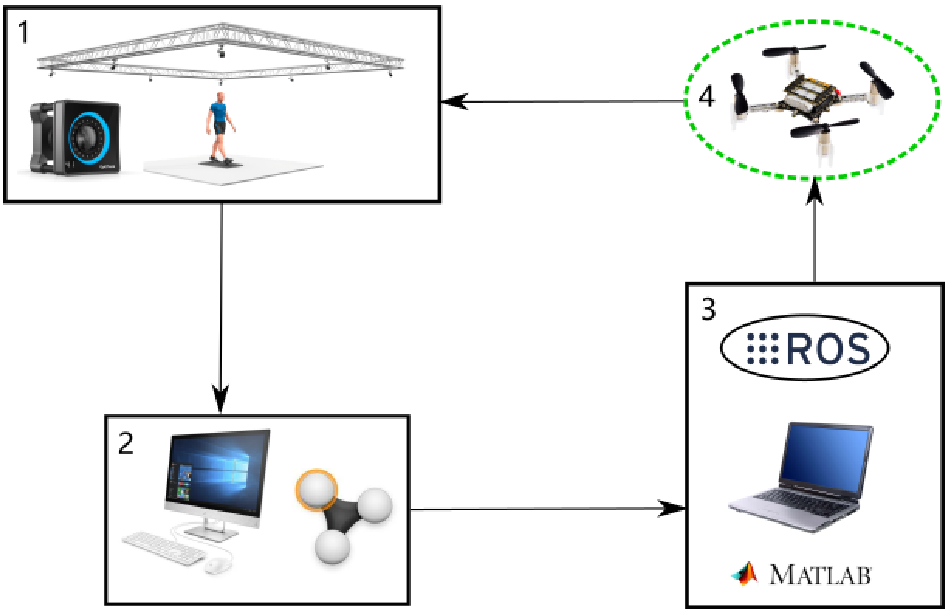


Figure 5-1: Scheme of the components used in the experiments

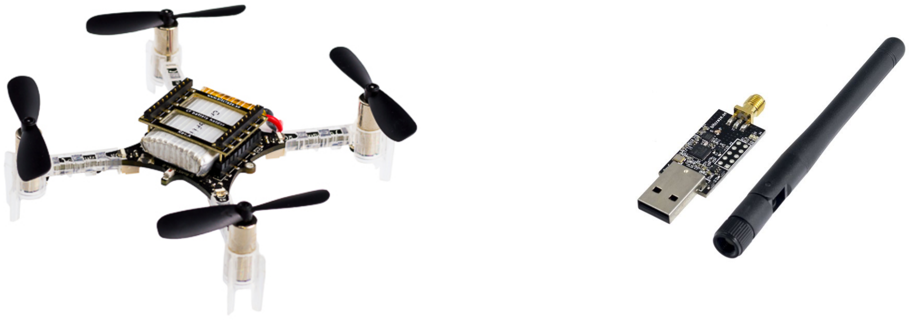
5-1-1 Quadrotor platform

The experiments are carried on the Crazyflie 2.1 quadrotor UAVs, open-source entry-level nano quadcopters targeting researchers and engineers. The quadcopter is shown as Fig. 5-2(a) and features a size of 92×92mm size. The small quadcopter (92×92mm) weights only 27g, making them suitable for indoor multi-quadrotor experiments. The Crazyflies are communicated with the operating laptop using a USB dongle, Crazyradio PA, as the Fig. 5-2(b) shows. The Crazyradio can communicate with the quadrotor swarm in different channels up to 1km away with a powerful amplifier. Therefore, only one Crazyradio is required in the experiment to send the messages.

The state of the quadrotor consists of the position $p \in \mathbb{R}^3$, the velocity $q = \hat{p}$ and the orientation $[\Phi; \Theta; \Psi]$, which represent the roll, pitch, and yaw. The control inputs of the quadrotors are the desired velocity $[q_p^x; q_p^y; q_p^z]$, and the desired orientation $[\Phi_p; \Theta_p; \Psi_p]$. The operating laptop sends the desired waypoints to the PID-based velocity controller, which will transfer the input to the quadrotors. The onboard microcontrollers implement the control input in the Crazyflies. Based on the posture information measured by the Internal Measurement Unit (IMU), the microcontrollers can output the required trusts of the four motors in terms of the control inputs.

It is also possible to send the desired velocity computed by the MPC directly to quadrotors. However, the microcontroller is not stable enough to control the robot's speed, as the output of the MPC is not equal to the PID controller. Due to the low onboard control precision of Crazyflie, a PID controller is required to stabilize the control velocity.

[1] https://store.bitcraze.io/collections/kits



(a) Bitcraze Crazyflie 2.0 quadrotor

(b) Crazyradio PA 2.4 GHz USB dongle

Figure 5-2: The target qradrotor (left) is controlled from a laptop using a USB dongle (right)[1]

5-2 Control Structure

The Robot Operating System (ROS) is an open-source framework for robot software development. It is mainly used to simplify the communications between different robotics platforms and components in Fig. 5-1. AROS software consists of ROS nodes, and the data is transformed as ROS topics. Each ROS node is a computation process that can subscribe and publish topics from other nodes. In our control systems, a group of ROS nodes constitutes a feedback controller connecting the Crazyfile 2.1 quadrotors, the MPC controllers, the Optitrack motion capture systems, and other equipment. The structure of our control system is shown in Fig. 5-3.

1. /**mocap_node**[19] This node is based on the ROS package **mocap_optitrack**. It is used to transform the position and orientation of robots measured by optitrack to the coordination of Matlab and the velocity controller.
2. /**position_velocity_orientation_filter**[3] The motion capture system can only measure the position and orientation of the robots. The velocity of the quadrotors is computed by a Kalman filter using the poses message transformed by /**mocap_node**.
3. /**MPC_planner** This node is our MPC planner in Matlab. Matlab can create its node using the Robotics System ToolBox. It will subscribe to the estimated position and velocities and publish the desired waypoint to the velocity controller.
4. /**crazyflie_controller** The PID-based velocity controller is based on the ROS package **crazyflie_controller**, which is a part of Crazyflie controller system **crazyflie_ros**[13]. Apart from publishing the desired velocity in terms of the

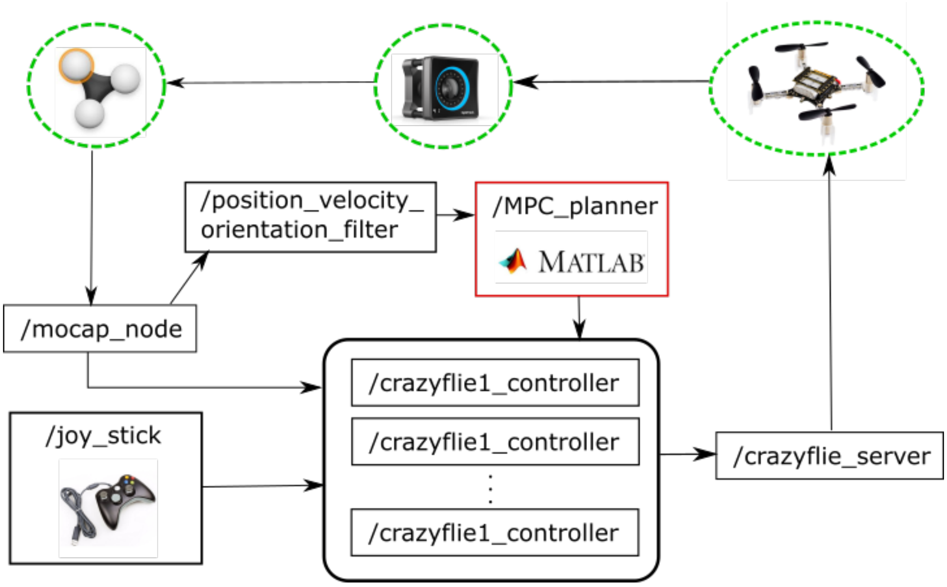


Figure 5-3: Scheme of the control structure of the quadrotors

MPC result, the controller can also issue the initialization and end commands. The takeoff and landing triggers are included in the controller, guiding the robot to the desired start and landing positions. The takeoff and landing functions are reorganized to fulfill the requirement of our experiment area. An emergency braking function is also designed to stop the experiments and land the quadrotors immediately.

5. /**joy_stick** This node will receive the actions of the Xbox controller. By pressing different buttons, I can send the takeoff command and landing command to activate the trigger in /**crazyflie_controller**. The joy sticker can also activate the emergency braking to avoid crushes.

6. /**crazyflie_server** This node is based on the ROS package **crazyflie_driver**, which is also a part of Crazyflie controller system **crazyflie_ros**. This node will listen to the commands of different quadrotors and send the message. It is the connection between the laptop and the Crazyradio.

The MPC planner receives the state of the quadrotors and computes a reference trajectory for each robot. The computations will start 5s after the quadrotors hovering at their start positions. The planning is performed continuously, which means as soon as one group of desired trajectories are computed, the next group of planning will start immediately. The desired horizontal positions are quadrotors are set to constant in the experiments, imitating the 2-dimensional environments in the simulations.

The basic settings of the MPC controller, including the time horizon, the sampling rate, and the motion limitation of the robot models, are mostly the same as the simulations in chapter 4. The only difference is the size of the robots and obstacles.

5-3 Experiment Results

Collision risk elements and congestion risk elements are the main topic of my thesis. Therefore, I have designed two experiments to test the performance of collision-risk-aware and congestion-risk-aware MPC motion planners. Similar to the simulations, only one quadrotor is used to test collision risks. Five static obstacles are placed in the workspace. Due to the limitation of experiments, the positions of the static obstacles are virtually defined in the MPC planner but not perceived by the optical systems. It will not influence the result of the experiments. As for the congestion risk elements, three quadrotors swap their positions, which is similar to the symmetric swapping scenarios in section 4-4.

The experiments are conducted at the Cyberzoo of the Faculty of Aerospace Engineering, Delft University of Technology, whose space is large enough to perform the experiments. The size of the experiment area is $8 \times 8m$, which is smaller than the simulations. The Fig. 5-4 shows the workspace for the experiments.

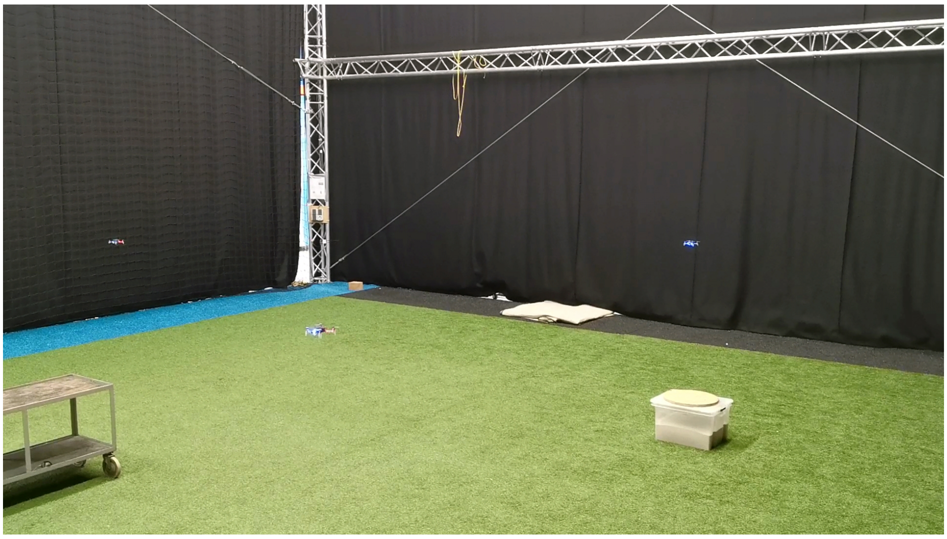


Figure 5-4: The workspace for the experiments. The box and the mental shelves are the takeoff platform of the quadrotors. The three nano quadrotors are the Crazyflie 2.1 UAVs

5-3-1 Collision Risk

As the Fig. 5-6 show, the five static obstacles are placed in the workspace. The sizes of the obstacles are equal to the robots. The experiment scenario is similar to the random corridor scenarios in section 4-3. The locations of the obstacles are similar to the examples in section 4-3. The three subfigures 5-6(a), 5-6(b) and 5-6(c) are the measured waypoints of the robots. The simulation results are not always equal to the experiments because of the imprecise quadrotor actuators. But the trend of the trajectories is similar to the simulations. The experiments prove that the planner considering the Distance to the Centroid of the Free Space can guide the robot to follow a safer trajectory, which can increase the clearance simultaneously.

The Fig. 5-5 shows how the clearances between the robot and the closed obstacle perform when the three planners are executed. The robot is placed close to an obstacle at the beginning. All three planners keep the robot at an appropriate distance to the obstacles in the middle of the trajectories. The clearance is also high enough to avoid collisions. As the goal is closed to the obstacles, the collision risk increases at the end of the trajectories. As the *PF* and *Clear* result in similar paths, the performances of the clearance are also closed to each other. The *Dis2Centr* avoids the narrow passage and chooses to work in the broader region. The experiments show that *Dis2Center* assess the risk more reasonably compared with other planners. As the obstacle density is considered in the planning, the robot prefers to choose a trajectory not surrounded by the obstacles. The robot's velocity makes no difference to the planners so that the inaccurate velocity actuator will not influence the performance of the proposed planner. Therefore, the planner can complete the required motion planning task in practice.

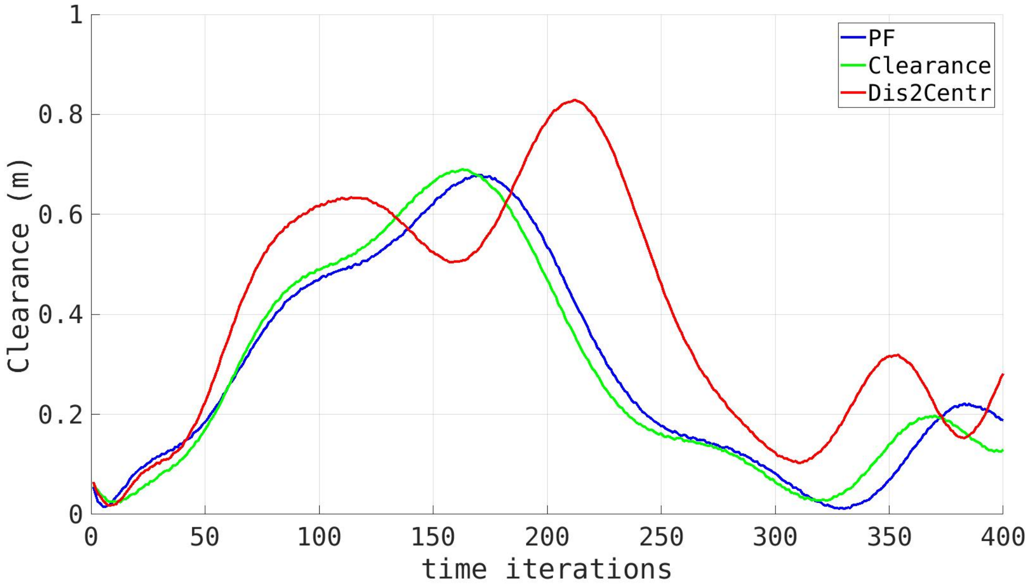


Figure 5-5: The workspace for the experiments. The box and the mental shelves are the takeoff platform of the quadrotors. The three nano quadrotors are the Crazyflie 2.1 UAVs

5-3-2 Congestion Risk

The Fig. 5-7 show the performances of the quadrotor group in experiments. Three quadrotors are used in the experiments. The initial positions are separated equally on a circle with a radius of 2.4m. The goal positions are opposite to the origin. As we mentioned in the previous sections, the velocity tracking of Crazyflie 2.1 is inaccurate. And using the planned velocity of the MPC will lead to higher error than using the intended positions in combination with a PID velocity controller. Therefore, the actual speed is different from the planned result. The movement of the quadrotors can be regarded as hovering between different desired waypoints.

And the linear velocities of the quadrotors are much smaller than expected.

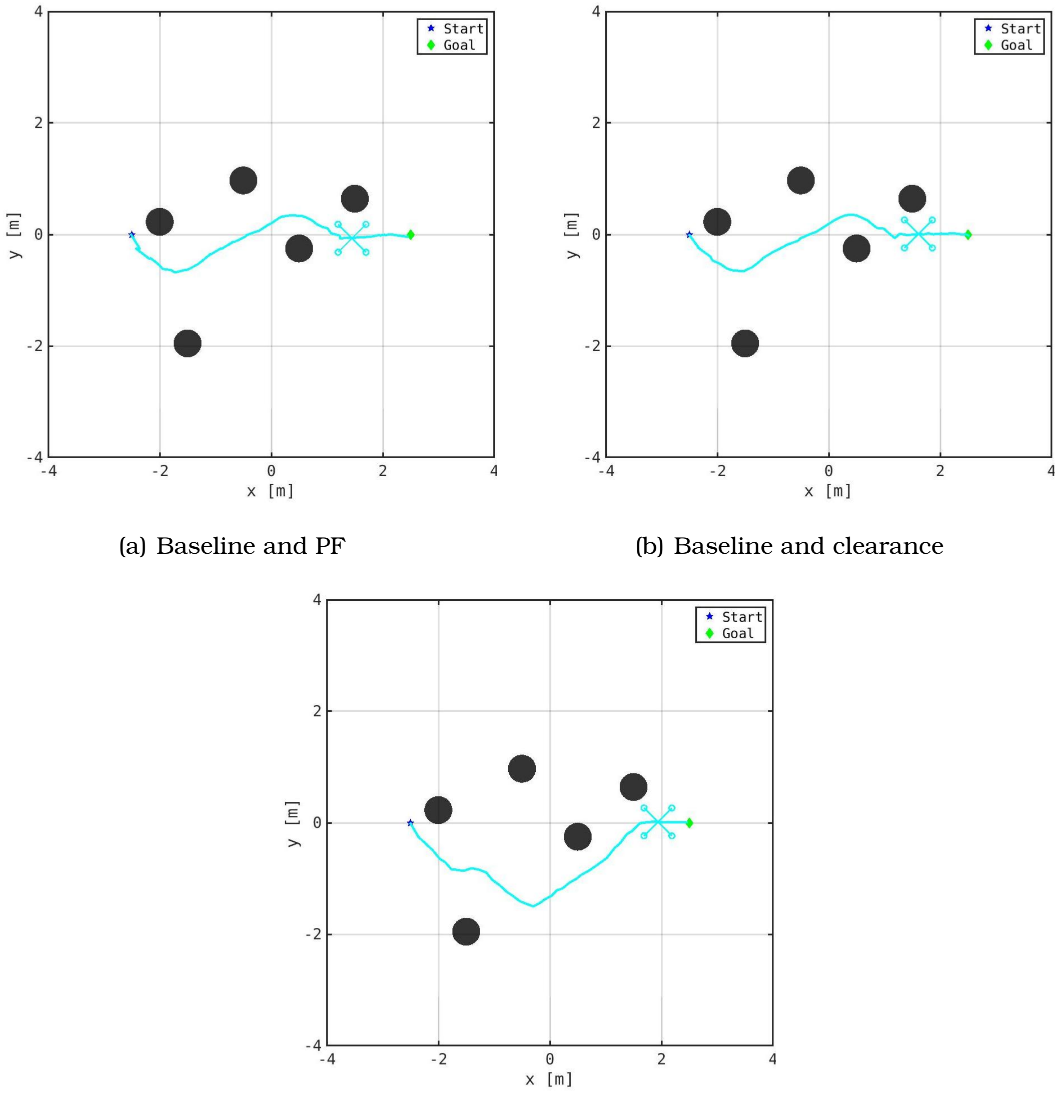
The experiments show that all three controllers can guide the robot to finish the paths without collision and congestion. But the shape and the length of the trajectories are similar. The clearance between quadrotors is safe enough to avoid collisions. And The robot will not congest in the middle. Several simulations have also been performed before the experiments, and the result shows that the trajectories planned by *P2G* are the shortest. *RLS* also performs better than the baseline in the simulations. But the superiority disappears in the experiments.

After analyzing the quadrotors' estimated and planned velocities, I conclude that poor velocity tracking leads to unexpected results. The planned speeds usually are two to three times higher than the estimated. The lower velocities will weaken the congestion in the planning, and the possibility of deadlock is lower than expected. *P2G* and *RLS* consider the relative velocity as an essential parameter to evaluate the congestion, and the basic structures of the planners are similar. Therefore, the low relative velocities will eliminate the difference between the three planners. Due to the limitations of the equipment and time, I have not performed further experiments on other quadrotors. But I will discuss my plan to solve the problems and show the expected performance of the controllers in section 5-4.

5-4 Discussion

In this chapter, the proposed method is validated in the experiments to show the performance in real environments. A multi-robot testing system is built up to control and communicate with the target quadrotor Crazyflie 2.1. The Crazyflie 2.1 is small and easy to maintain. The open-source control software reduces the development difficulty. However, the velocity tracking of the Crazyflie 2.1 can not meet the requirement. Therefore, I developed a position tracking control platform to track the desired waypoints in the workspace.

The experiments show that, *Dis2Centr* can assess collision risk more reasonable compared with *PF* and *Clear*. The quadrotor can choose the heading positions considering the obstacle density. Therefore, the risk of the planned trajectories becomes lower. However, in the experiments to test the performance of *PF*, *RLS* and *P2G*, the result paths are similar. The poor velocity tracking ability is the main cause of this problem. As we mentioned in the section 5-1, the **crazyflie_control** node is adapted to control the output velocities. Using different control functions can promote the quality of the trajectory. And adjusting the PID parameters may also promote the control qualities. Another solution is using different bands of quadrotors. By using quadrotors with a larger size, the control stability can be enhanced. The final solution is to change the dynamic functions of my controllers into 3-dimensional nonlinear functions, whose state values consist of roll, pitch, yaw, and horizontal trust. It is also possible to use the classic Euler angles and trust as the input commands of the quadrotors. The document from Crazyflie 2.1 website shows that using Euler angles and trust can achieve expected control performance. Therefore, it is possible to modify the equality constraints of the MPC to output desired Euler angles and trust, which can be sent to the quadrotors directly.

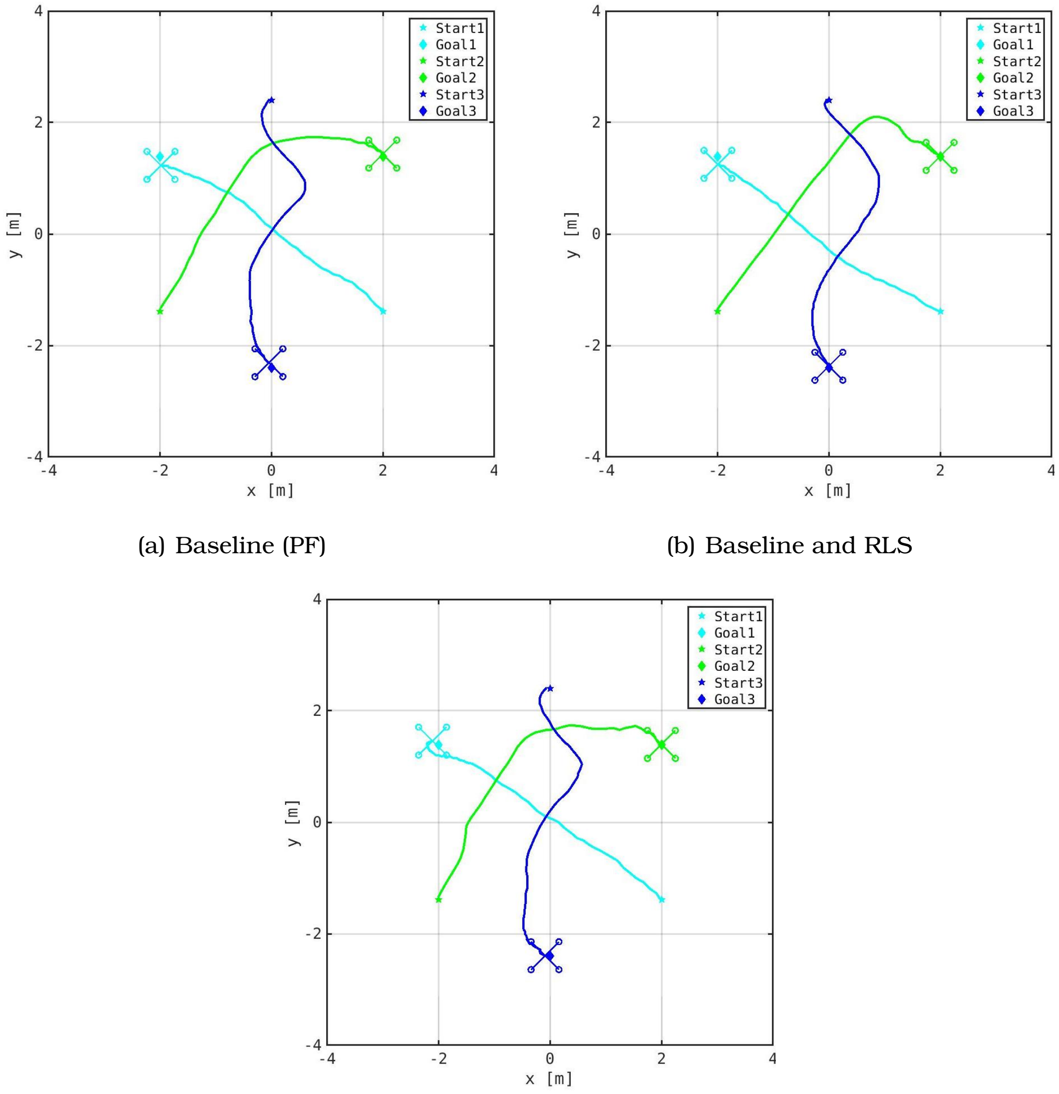


(a) Baseline and PF

(b) Baseline and clearance

(c) Baseline and Dis2Cnetr

Figure 5-6: Experiment results of the collision-risk-aware motion planner. The blue star and the green diamond represent the initial and goal positions. The cyan trajectories represent the estimated positions of quadrotors.



(a) Baseline (PF)

(b) Baseline and RLS

(c) Baseline and P2G

Figure 5-7: Experiment results of the congestion-risk-aware motion planner. The stars and the diamonds represent the corresponding initial and goal positions.

Chapter 6

Conclusion and Future Work

6-1 Conclusions

The goal of this thesis is to propose a risk-aware distributed motion planner for multi-robot systems. The risk is the relative likelihood of the robot not being able to finish the paths. This thesis focuses on navigation risk, including collision risk and congestion risk in motion planning.

To evaluate collision risk, I have developed the deformed distance to the centroid of free space (*Dis2Centr*), which represents the obstacle density and the distance to the optimal positions. The free space is the intersection of the valid free region and the origin free space, eliminating the influence of invalid obstacles and boundaries.

It is more complex to parameterize congestion risk in cluttered environments. I have proposed the Potential to Goal (*P2G*), twisting the navigation cost by the intention of motion and velocity skewing parameter. The intention of motion represents the angle between the two vectors. The first connects the initial, the objective robot, and the terminal, the moving obstacle robot. The second connects the initial, the moving obstacle robot, and the terminal, the goal position. The skewing parameter deforms the peak of the intention of motion to the velocity of the moving obstacle robot. The navigation cost twisted by *P2G* is defined as the distance to the goal. The robot entails high risk when the moving robots locate in the directions to the goal position, and low risk when the objective robot is close to its goal or distant from other agents in workspaces.

To assess the navigation risk in motion planning, I have designed a risk-aware distributed model-predictive-control(MPC)-based multi-robot motion planner. The positions of other robots in the future are predicted by the solutions of the corresponding MPC controller. Moreover, the collision constraints of the MPC are defined as the original free spaces, the polygons in 2-dimensional workspaces. Laying on the foundations of the baseline, I have involved the risk elements in the objective functions of the MPC to test the performance in different scenarios.

In the simulations of collision risk elements, the trajectories planned by *Dis2Centr* show higher clearance to static obstacles, and the number of collisions is lower than other planners, which proves that *Dis2Centr* can assess collision risk better. In the simulations of congestion risk elements, the robots planned by *P2G* and potential field (*PF*) show the highest average velocities, shortest trajectory time, and lower number of deadlocks, which has proved that the congestion risk is significantly controlled. In the scenarios showing high congestion risk, the robots adopted *P2G+PF* can also keep high clearance to other agents, which proves the planner can also minimize collision risk. I have also compared the performance of the robots under different uncertainties. High position uncertainties will increase collision risk and reduce congestion risk. In all scenarios, the planners with *Dis2Centr* and *P2G+PF* show expected performance compared to other classic planners. The numbers of collisions and deadlocks vary slightly by the uncertainties.

To validate the performance of the proposed methods, we have tested the planners on the Crazyflie 2.1 quadrotors with the motion capture system Optitrack. The experiment results of the collision-risk-aware planner show that, the proposed *Dis2Centr* can plan a safer trajectory than the other planners. The robot prefers to choose a broader passage to pass through the obstacle group. As for the congestion-risk-aware planner, the experiment results show similar performance among the three controllers. The result shows that accurate velocity tracking makes a significant difference in the congestion risk assessment. Therefore, a different experiment platform should be applied in the future to test the performance of the proposed method.

In conclusion, it is necessary to minimize the collision and congestion risk in motion planning to prevent failures. The risk-aware multi-robot motion planner *Dis2Centr* and *P2G+PF* controls the risk in planning under uncertainty well in various scenarios. *Dis2Centr* and *P2G+PF* are sufficient risk-aware multi-robot motion planners to produce trajectories with slight navigation risk.

6-2 Future Work

During the design of the collision and congestion risk elements, I have tried different formulations of the functions to make the method more reasonable. Furthermore, the result has shown me how challenging to develop a new numerical formulation to assess risks. I hide the process of formulating the elements, as the elements I attempted are invalid to evaluate the risk. I have used a contour map and simulation results to prove my elements. In the future, more mathematical derivations should be taken to complete my theory.

In the planner with *Dis2Centr*, the free space area in the formulations is assumed as a constant during optimization, which will weaken the assessment of obstacle density. My approximation will increase the computation speed, but a more precise approximation formulation may promote the result. As for the planner with *P2G*, I only test its performance of the potential in assessing the collision risk. Further tests can be generated to compare the influence of collision risk elements. I have not tested the performance of the planner with *P2G+Dis2Centr* because *Dis2Centr* will show a negative influence on the congestion risk assessment. Therefore, a more suitable method can be generated to solve the problems.

As for the MPC controller, I have used the corresponding planning result as the future positions, eliminating prediction errors. In workspaces with dynamic obstacles, accurate prediction methods are required to prevent errors. Classic methods, like the constant velocity model (CVM), are insufficient in cluttered environments. A recurrent neural network (RNN) can predict the robot positions by the testing samples. Therefore, the risk-aware MPC-based motion planner can be transformed into a decentralized planner by using prediction methods, reducing communication costs, and preventing communication errors.

The problem of the experiments is the inaccurate velocity tracking of the quadrotors. As the relative velocities are used to assess congestion risk, this problem will weaken the risk evaluation of the controllers. Changing another quadrotor platform may solve the problems. Another solution is to transform the dynamic equality constraints in the MPC into 3-d and send the Euler Angles and horizontal trust as the input message of the quadrotors. The planners that are not related to velocities, like Î1extitPF and *Dis2Centr*, will not be influenced by the problem. More experiments are required to evaluate the performance of *P2G* and *RLS*.

Bibliography

[1] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

[2] Javad Amiryan and Mansour Jamzad. Adaptive motion planning with artificial potential fields using a prior path. In *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, pages 731–736, 2015.

[3] bebop_autonomy. https://bebop-autonomy.readthedocs.io/en/latest/.

[4] Eric Boivin, Andre Desbiens, and Eric Gagnon. Uav collision avoidance using cooperative predictive control. In *2008 16th Mediterranean Conference on Control and Automation*, pages 682–688, 2008.

[5] C Burstedde, K Klauck, A Schadschneider, and J Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3):507–525, 2001.

[6] S. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4):837–855, 2018.

[7] A. Domahidi and J. Jerez. *Forces professional.* Embotech Gmbh(http://embotech. com/forces-pro/), 2014.

[8] Carlos Bordons Alba Eduardo F. Camacho. *Model Predictive Control.* Springer-Verlag London Limited, 2st edition, 2003.

[9] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[10] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1):23–33, 1997.

[11] Yi Guo and L.E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 3, pages 2612–2619 vol.3, 2002.

[12] J. Hardy and M. Campbell. Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. *IEEE Transactions on Robotics*, 29(4):913–929, 2013.

[13] Wolfgang Hönig and Nora Ayanian. *Flying Multiple UAVs Using ROS*, pages 83–118. Springer International Publishing, 2017.

[14] S. Hoshino and K. Maki. Safe and efficient motion planning of multiple mobile robots based on artificial potential for human behavior and robot congestion. *Advanced Robotics*, 29(17):1095–1109, 2015.

[15] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian. Trajectory planning for quadrotor swarms. *IEEE Transactions on Robotics*, 34(4):856–869, 2018.

[16] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, 1985.

[17] Jean-Claude Latombe. *Robot Motion Planning*. Springer Science & Business Media, 2012, 1991.

[18] Anirudha Majumdar and Marco Pavone. How should a robot assess risk? towards an axiomatic theory of risk in robotics. In Nancy M. Amato, Greg Hager, Shawna Thomas, and Miguel Torres-Torriti, editors, *Robotics Research*, 2020.

[19] mocap_optitrack. `https://github.com/ros-drivers`.

[20] Alyssa Pierson, Wilko Schwarting, Sertac Karaman, and Daniela Rus. Navigating congested environments with risk level sets. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5712–5719, 2018.

[21] Luciano C. A. Pimenta, Nathan Michael, Renato C. Mesquita, Guilherme A. S. Pereira, and Vijay Kumar. Control of swarms based on hydrodynamic models. In *2008 IEEE International Conference on Robotics and Automation*, pages 1948–1953, 2008.

[22] Alexei Yu. Uteshev and Marina V. Goncharova. Point−to−ellipse and point−to−ellipsoid distance equation analysis. *Journal of Computational and Applied Mathematics*, 328:232–251, 2018.

[23] J. van den Berg, Ming Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935, 2008.

[24] Jur van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In Cédric Pradalier, Roland Siegwart, and Gerhard Hirzinger, editors, *Robotics Research*, pages 3–19, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[25] X. Xiao, J. Dufek, and R. Murphy. Explicit motion risk representation. In *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 278–283, 2019.

[26] Xuesu Xiao. Risk-aware path and motion planning for a tethered aerial visual assistant in unstructured or confined environments. *CoRR*, abs/2007.09595, 2020.

[27] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager. Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells. *IEEE Robotics and Automation Letters*, 2(2):1047–1054, 2017.

[28] H. Zhu and J. Alonso-Mora. B-uavc: Buffered uncertainty-aware voronoi cells for probabilistic multi-robot collision avoidance. In *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 162–168, 2019.

[29] H. Zhu and J. Alonso-Mora. Chance-constrained collision avoidance for mavs in dynamic environments. *IEEE Robotics and Automation Letters*, 4(2):776–783, 2019.

Glossary

List of Acronyms

MPC	model predictive control
VO	velocity obstacle
RVO	reciprocal velocity obstacle
ORCA	optimal reciprocal collision-avoidance
ARF	artificial potential field
BVC	Buffered Voronoi Cells
MPC	model predictive control
Dis2Centr	distance to the centroid of the free space
RLS	Risk Level Set
P2G	Potential to Goal