Decentralized Collision Avoidance of Multi-Agent Systems in 3-Dimensional Space

Xiajing Li

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Decentralized Collision Avoidance of Multi-Agent Systems in 3-Dimensional Space

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Xiajing Li

May 10, 2019

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

Abstract

Multi-Agent Systems, often referred to a network of loosely connected autonomous units, are widely used to model the dynamics of crowds, vehicles, robots and swarms in traffic management, biological environment, distributed control and communication technologies. Recently, the study of multi-agent systems is rapidly growing due to the beneficial advantages of using a team of agents in logistics, mapping, search and rescue, etc.

In this thesis, we focus on the problem of decentralized collision avoidance among multiple intelligent moving agents. Imaging each agent to be a human, a moving car, or an aircraft, it is supposed to make its decision independently based on its perception of the local environment through sensors only. Two different collision avoidance protocols are presented to generate updated reference velocity continuously for each agent that leads to no future collision. The first method, the rotation based method, is adapted by the algorithm introduced in [1] for 2-Dimensional space and the second method, the potential fields based method, could be categorized as an example of Harmonic Potential Fields (HPF) discussed in [2]. Both methods employing position information of obstacles result in collision-free paths for each agent under the assumption that all other agents following similar maneuvers. Through simulations in MATLAB, satisfied performances are achieved for method 2 in all challenging scenarios we set while method 1 faced some difficulties dealing with multi-obstacle simultaneously as well as 3D scenarios.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Firstly, I would like to thank my supervisor Dr.ir. Giulia Giordano for her assistance during the whole processes of conducting this thesis. This thesis would not have accomplished as present without her kindness and encouragement.

Secondly, I would like to thank my committee members Dr.ir. Tamas Keviczky and Dr.ir. Riccardo Ferrari for spending their time reading and evaluating my thesis seriously.

Last and most importantly, I would like to thank my beloved family for their unconditional love and everlasting support during my study in TU Delft. I also owe my sincere gratitude to all my friends for their accompany during my master period in the Netherlands.

Delft, University of Technology Xiajing Li
May 10, 2019

Chapter 1

Introduction

1-1 Background

1-1-1 Networked-Systems and Multi-Agent Systems

Plenty of systems in the real world that are everywhere around us could be modelled as *networked systems* [6–8]. Networked systems are those dynamical systems that are composed of several interconnected dynamic units that are dynamically coupled and/or exchanging information with one another. Networked systems are typically very large-scale systems [9], resulting from the interaction of a huge number of units.

In nature, networked systems can be found at various scales and arise from the interaction of myriads of entities (chemical species, molecules, cells). At the chemical and bio-molecular level, the interaction of chemical species in chemical and biochemical processes gives rise to chemical reaction networks, and to gene regulatory networks and metabolic networks [10–26], which can be represented as interconnected dynamical systems [27,28]. Also biological systems at a larger scale, such as the ecosystems encountered in ecology, can be seen as networked systems arising from the interaction of various species sharing the same environment [29–35].

In the social science, economical networks, social networks and opinion dynamics have been successfully modelled as networked systems [36–39]. Consensus [40–45], which is related to opinion dynamics, is similar to the problem of synchronization in technological systems: these problems have been successfully addressed by studying interconnected systems [46].

An important class of networked systems are flow networks [47–53], which provide many examples of networked systems in technological networks. Applications range from water distribution networks [54, 55] to inventory management and production-distribution systems [56–63], from power networks and smart grids [64–67] to transportation networks [68, 69] as well as traffic and congestion management systems [70–73].

Another important application is related to information and communication technologies, where telecommunication and data communication networks, such as computer networks, can

be seen as interconnected dynamical systems to devise algorithms that drive and optimize the transmission of packets and data [70, 74–78].

Due to the increase of complexity and the huge scale of these systems, a centralized control method that uses all information for a single controller to determine a global optimal strategy is often inefficient, uneconomic or even physically impossible. A centralized controller could be unfeasible as the result of limitations on energy resources and communication bandwidth, computation constraints, and delays or packet losses that make the system more sensitive to failure and modeling errors. Furthermore, there is also a privacy issue. When the external global controller intends to access local information, it can be restricted or denied as that local information is private and confidential. Alternatively, it is of great advantage to implement multiple controllers with access to local information only. This approach goes under the name of distributed or decentralized control.

The difference between distributed and decentralized control approaches depends on whether there is communication between the individual controllers or not. In particular, distributed control makes use of information exchange between the controllers allocated within a specified control layer [79], while decentralized control is enforced by control units that are not aware of one another and do not exchange any information [80–83].

A large-scale system where information is exchanged among the subsystems is typically modelled as a graph or hypergraph [84], where the nodes in the graph represent the subsystems and the arcs refer to the interactions among the subsystems. A specific kind of decentralized control is called network-decentralized control [85–87]. While the general decentralized controllers act on the nodes of the graph, namely every subsystem has its own local controller, network-decentralized controllers are associated with the arcs of the graph, namely with the connections among subsystems. Network-decentralized methods could be used to control large-scale system also in the presence of structural constraints and uncertainties. Also distributed estimation [88, 89] and decentralized estimation [90–93], as well as network-decentralized estimation strategies [94] have been discussed in the literature.

A Multi-agent system, as a particular kind of networked dynamical systems, contains a group of autonomous agents, such as Unmanned Ground Vehicles(UGVs) and Unmanned Aerial Vehicles(UAVs). Typical problems such as vehicle platooning and formation flight of aircraft [71, 95–98] have been successfully tackled by modelling a bunch of vehicles or of aircraft as a multi-agent system.

1-1-2 Collision Avoidance for Multi-agent Systems

In this thesis, we focus on problems involving multiple aircraft, such as UAVs, moving in a three-dimensional space and design algorithms that avoid collision among them by modelling the swarm of UAVs as a multi-agent system, more precisely, several identical agents navigate independently and simultaneously in a three-dimensional environment shared with obstacles and other moving entities. Each agent enforces its own decision-making strategy based on sensor information within its detection zone only. The task is to make sure all agents are able to reach their goal positions without collision with one another or any obstacles. As there is no central station to control all the agents, it is a decentralized method where each of the agents makes its own decision autonomously, based on local information only (within its "detection zone"), and this still leads to a collision-free global behaviour. A decentralised

approach allows us to solve the problem and face complex physical constraints of energy, communication range, delay and drop as well as computation burden.

Collision avoidance, how to prevent agents from getting in each others way [99], is a crucial aspect for the navigation of multi-agent systems. Differing from motion planning, which focuses on the problem of dealing with one agent located in the known global environment and generating a complete path towards a goal configuration at once [100], collision avoidance deals with local sensed surrounding rather than global environment and new action is derived in every iteration. The action for each agent is computed simultaneously and independently, without communication among the agents or central coordination, just as a simple reaction to the sensed environment. The difficulty for collision avoidance of the multi-agent system is that each agent makes its own decision. The next movement of a selected agent depends on surrounding obstacle agents but the new velocity for the selected agent is not exactly predictable for agents nearby as they do not have full (prior) knowledge of the circumstance the selected agent is facing. But still, we would like to derive a collision-free path involving several agents that are cooperating in a cluttered environment.

Among the intensive literature that addressed the problem of collision avoidance, two main streams are the algorithms based on Velocity Obstacle (VO) [101] and the algorithms based on Artificial Potential Fields (PFs) [102].

VO-based algorithms utilize both position and velocity information to generate velocity obstacle set to select new velocity that avoids collisions while minimizing the change in velocity and are applicable to multi-agent scenarios in a complex environment. While the initial VO proposed in [101] is only capable to deal with moving obstacles with a known velocity, lots of extensions following the same core concept were proposed. Reciprocal Velocity Obstacle(RVO) introduced in [3] takes the reaction of other intelligent decision-making entities into account and share the responsibility of avoiding collision so that navigation procedure is safe and oscillation-free. Hybrid Reciprocal Velocity Obstacles (HRVO) [4] combines a reciprocal velocity obstacle and a velocity obstacle to overcome the drawback of "reciprocal dance" in RVO that the agents cannot reach agreement on which side to pass each other. While all aforementioned methods deal with a complex non-convex optimization problem solved via uniform sampling, Optimal Reciprocal Collision Avoidance (ORCA) [5] reduced the problem to a low-dimensional linear programming. Based on the formulation of ORCA, the application for multi-robot under differential-drive constrains or car-like non-holonomic model are represented in [103] [104], [105]. All algorithms could be adopted to 3-D as demonstrated in [106–109]. Implemention in physical quadrotor helicopters is elaborated in [110].

While the VO-based approaches require the estimation of velocity, potential fields based algorithm relies only on position estimation of surrounding obstacles. The typical potential field as mentioned in [102] is consisted of an attracting potential that placed at the target point to navigate the agent towards the goal and repulsive potentials that placed at the positions of obstacles to push the approaching agent away from the obstacle. The agent would always follow the path with the maximum negative gradient of the formulated potential field till the goal position which is the global minimum and unique zero of PF function. The inherent limitations of PFs methods summarized in [111] are trap situation due to local minima, no passage between closely spaced obstacles and oscillations in the presence of obstacles or oscillations in narrow passages. To address these issues, some modification has been introduced in [112], [113]. Harmonic Potential Field (HPF) is efficient to eliminate local minimam in a

cluttered environment [114]. Further study based on HPF is given in [2,115,116] etc. In [117] a potential field based method using gyroscopic forces and scalar potentials is proposed to create swarming behaviors for multi-agent systems while collisions among the agents and obstacles could be avoided.

Besides the methods included in these two categories, there are many other approaches to tackle collision avoidance problem such as the simplest rule based approach [118], where the agent switches between two distinct behaviors based on the sensor measurements, dynamic window approach [119] [120], safety-enhanced avoidance policy (SWAP) [121, 122] and geometric based approach like [123, 124]. Moreover, it is possible to employ Model Predictive Control (MPC) [125] as well as Deep Reinforcement Learning (DRL) [126] to resolve the hazard of collision.

1-2 Thesis Contribution

Algorithms for collision avoidance are often based on an intuition of how the agents should take steps to avoid hitting one another, as people do when they walk in crowded rooms or streets. The main contribution of this thesis is to translate the human intuition of dealing with collision avoidance problem into a mathematical representation that enables the design of rigorous algorithms for collision avoidance, targeting in particular agents that are moving in three-dimensional space.

We propose two different methods for collision avoidance maneuvers: the first is a rotation based approach, the second is a potential field based approach. In both cases, the algorithm requires current position information only. To rigorously validate the algorithms, their performance for crowded multi-agent scenarios and 3-dimensional spaces are investigated by simulation in MATLAB. Our investigation reveals that the second method outperforms the first (which can give rise to livelocks in 3D spaces). Hence, the second method is most extensively tested in simulations. For this method, in a wide variety of reasonable scenarios, our extensive simulation results suggest that collision-free paths leading the agents to their targets can be generated for 2-agent and multi-agent cases, in both 2D and 3D workspaces.

1-3 Outline

The remaining part of this thesis is organized as follows.

Chapter 2 reviews previous work in the context of multi-agent collision avoidance, in particular, velocity obstacle based approaches and potential fields based approaches.

Chapter 3 gives the formal problem formulation studied in this thesis. Also, the dynamic model and control strategy for a single agent are introduced. The evaluation metrics for collision avoidance behavior are provided as well.

Chapter 4 presents a rotation based collision avoidance protocol, as well as the choice of parameters and some simple 2-agent simulations.

Chapter 5 illustrates a potential field based collision avoidance protocol. The suggestion of tuning parameters and adaption to 3D are also elaborated.

Chapter 6 shows a more complex simulation for multi-agent scenarios in both 2-dimensional and 3-dimensional environments.

Chapter 7 concludes the work contained in this thesis and proposes some recommendations for future work.

Chapter 2

Prior work

The detailed explanation of the two main categories of collision avoidance methods is illustrated in this chapter. For both methods, the generated output would be the desired velocity. While the potential field based approach needs only position estimation of neighboring agents, velocity obstacle based approach also requires velocity estimation. The VO-based method can provide a sufficient condition that results in collision avoidance with other moving entities if applicable while minimizes velocity variation. Although potential field based approach lacks mathematical analysis of guaranteed collision avoidance, some methods provide satisfied practical results.

2-1 Velocity Obstacle Based Methods

The concept of velocity obstacle is first proposed in [101], where a sufficient and necessary condition for an agent selecting new velocity to avoid collision with obstacles moving at fixed velocities is given. The geometric interpretation of velocity obstacle is shown in Figure 2-1.
Let $\lambda(p, v) = \{p + tv | t \geq 0\}$ denote a ray starting at p and heading in the direction of v.
Let $X \oplus Y$ denote the Minkowski sum of sets X and Y, and let $-X$ denote the set X reflected in its reference point:

$$X \oplus Y = \{x + y | x \in X, y \in Y\}, \quad -X = \{-x | x \in X\}. \tag{2-1}$$

Definition 2.1. ***Velocity Obstacle:*** *Consider a circular agent A of radius r_A and an obstacle B of radius r_B currently at position p_A and p_B with velocities v_A and v_B.*
The velocity obstacle for B with velocity v_B from the point-view of A is defined as

$$VO_B^A(v_B) = \{v_A | \lambda(p_A, v_A - v_B) \cap \mathcal{A} \oplus -\mathcal{B} \neq \emptyset\} \tag{2-2}$$

This means that if $v_A \notin VO_B^A(v_B)$, A and B will never collide according to their current velocity. In each decision-making cycle, a velocity obstacles set $\bigcup_{B \in O} VO_B^A(v_B)$ for each

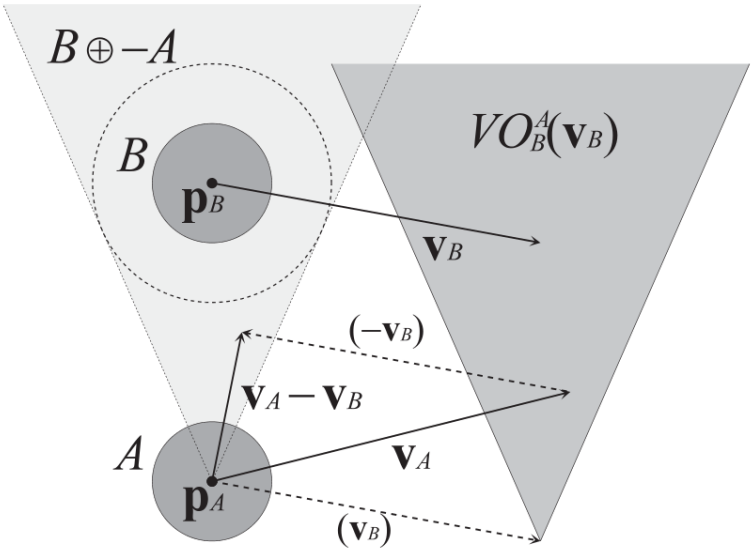


Figure 2-1: The velocity obstacle for obstacle $\mathcal{B}$ with velocity v_B from the point-view of agent $\mathcal{A}$. Figure taken from [3]

agent is composed by all velocities leading to a collision with obstacles. New velocity is selected as the one most directed towards the agents goal position while outside the VO set to guarantee a collision-free path towards its goal.
Some properties of velocity obstacle can be pointed out following [3]:

1. **Symmetry:** If agent A with velocity v_A will collide with agent B with velocity v_B, then agent B with velocity v_B will also collide with agent A with velocity v_A.

$$v_A \in VO^A_B(V_B) \Leftrightarrow v_B \in VO^A_B(V_A) \tag{2-3}$$

2. **Translation Invariance:** If agent A with velocity v_A will collide with agent B with velocity v_B, then agent A with velocity $v_A + v_0$ will also collide with agent B with velocity $v_B + v_0$.

$$v_A \in VO^A_B(V_B) \Leftrightarrow v_A + v_0 \in VO^A_B(V_B + v_0) \tag{2-4}$$

3. **Convexity:** A velocity v_A in the half-plane to the right(left) of $VO^A_B(V_B)$, $v_A \overrightarrow{\notin} (\overleftarrow{\notin}) VO^A_B(V_B)$, would let A pass B on the right(left) side. Any velocity in between two velocity v_A and v'_A in the right(left) half-plane is still in the same right(left) plane.

$$v_A \overrightarrow{\notin} VO^A_B(v_B) \cap v'_A \overrightarrow{\notin} VO^A_B(v_B) \Rightarrow (1-\alpha)v_A + \alpha v'_A \overrightarrow{\notin} VO^A_B(v_B), \quad \text{for} \quad 0 \leq \alpha \leq 1 \tag{2-5}$$

The outcome of the velocity obstacle is a one-step change in velocity to avoid a future collision under the assumption that other objects will move continuously with previous velocities, which does not take the interaction effect of other moving agents into account. A typical consequence would be the undesirable oscillation described in [3]. To tackle the oscillatory motions, the Reciprocal Velocity Obstacle (RVO) is introduced in [3]. Follow the geometric expression shown in Figure 2-2 . The Reciprocal Velocity Obstacle is identical to the Velocity Obstacles

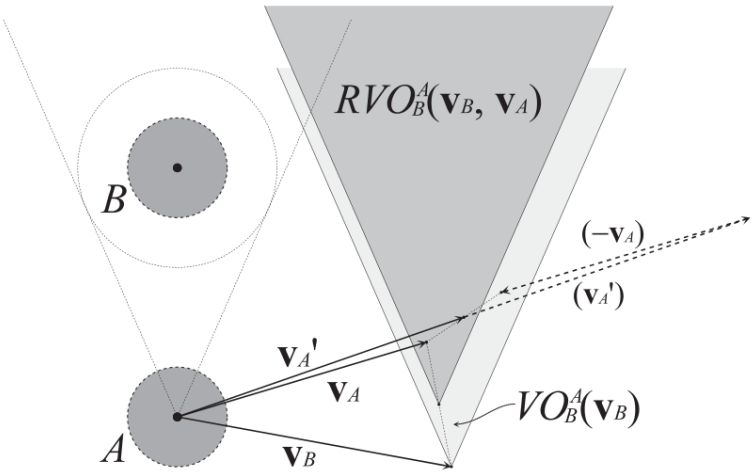


Figure 2-2: The Reciprocal Velocity Obstacle $RVO^A_B(v_A, v_B)$ of agent B to agent A. Figure taken from [3]

mentioned above in the relative frame, while the collision avoidance mission is equally shared between both interacting agents (both agent preserve half of the previous velocity). In global frame, the apex of velocity obstacle set shift from v_B in VO to $\frac{v_B+v_A}{2}$ in RVO.

Definition 2.2. ***Reciprocal Velocity Obstacle:***

$$RVO^A_B(v_A, v_B) = \{v'_A | 2v'_A - v_A \in VO^A_B(v_B)\} = \{v'_A | v'_A \in VO^A_B(\frac{v_B + v_A}{2})\} \tag{2-6}$$

The RVO can be used to generate collision-free and oscillation-free motions for each agent. Considering the shortcoming of reciprocal dance for RVO, Hybrid Reciprocal Velocity Obstacle(HRVO) is proposed in [4]. As illustrated in Figure 2-3,

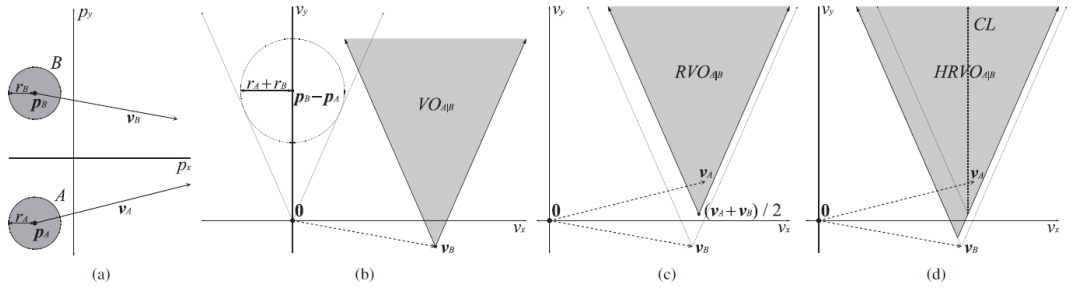


Figure 2-3: Comparison of several VO based approaches [4]

HRVO is a hybrid of a reciprocal velocity obstacles and a velocity obstacles, when v_A is to the right of the center line of RVO^A_B, it is preferred to choose a new velocity to the right of RVO^A_B to minimize the change of velocity. To encourage this, the unwilling side of RVO^A_B is replaced by VO^A_B to give agent B fully priority to modify v_B in order to avoid collision.

The previous method only take the velocity direction into consideration and assume the speed is kept constant the whole journey. By introducing a time window τ in [5], the task

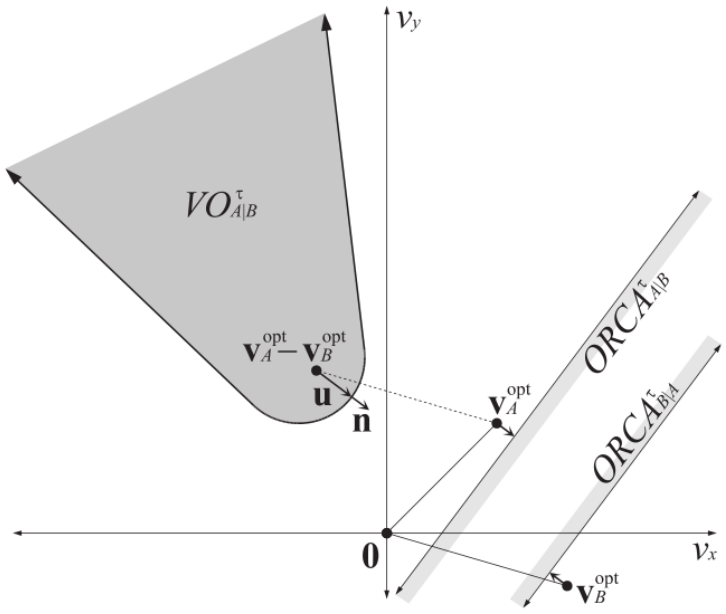


Figure 2-4: The Optimal Reciprocal Collision Avoidance. Figure taken from [5]

now becomes independently and simultaneously selecting a new velocity (both magnitude and direction could be modified) for each agent to guarantee collision-free with all agents for a time period τ. The velocity obstacle with time window $VO^{\tau}_{A|B}$, all relative velocities for A with respect to B that cause a collision before time τ, is defined as follows.

$$VO^{\tau}_{A|B} = \{v|\exists t \in [0,\tau], v \in D(\frac{P_B - P_A}{t}, \frac{r_A + r_B}{t})\} \tag{2-7}$$

where $D(x,r) = \{q|\|q-p\| < r\}$ denotes an open disc with center x and radius r. The set of collision-avoiding velocities $CA^{\tau}_{A|B}(v_B)$ for agent A given obstacle B selects its velocity from set V_B is define as

$$CA^{\tau}_{A|B}(V_B) = \{v|v \notin VO^{\tau}_{A|B} \oplus v_B\} \tag{2-8}$$

If $v_A = CA^{\tau}_{A|B}(v_B)$ and $v_B = CA^{\tau}_{A|B}(V_A)$, then V_A and V_B is reciprocally maximal.

Let w be the vector from $v_A - v_B$ to the closest point of the boundary of $VO^{\tau}_{A|B}$, as indicated in Figure 2-4.

$$w = (\arg\min_{v \in VO^{\tau}_{A|B}} ||v - (v_A - v_B)||_2) - (v_A - v_B) \tag{2-9}$$

The vector w represents the smallest change to the relative velocity $v_B - v_A$ that will provide a sufficient condition for no collision during the time window τ. The set of permitted velocities for A determined by B for a time window τ $ORCA^{\tau}_{A|B}$ is the half-plane (or half-space in 3D) in the direction of the outward normal of $VO^{\tau}_{A|B}$ at the point $v_A - v_B + w$ starting at the point $v_A + \alpha w$.

$$ORCA^{\tau}_{A|B} = v|v - (v_A + \alpha w) \cdot n \geq 0 \tag{2-10}$$

The reciprocity factor $\alpha \in [0,1]$ prescribes the share of avoid collision between A and B, while A adjust its velocities by αw, B take the remaining $1-\alpha$ part of the responsibility. In [5], the value $\alpha = \frac{1}{2}$ is used as both agent take equal responsibility. The set of permitted velocities

for A with respect to all agent is the intersection of half-planes in 2D or half-spaces in 3D introduced by each of the other agents. The permitted velocity set could be written as

$$ORCA_A^\tau = D(0, v_A^{max}) \cap \bigcap_{A \neq B} ORCA_{A|B}^\tau \tag{2-11}$$

The new velocity v'_A for A is selected as the one inside the region of permitted that is closet to its preferred velocity.

$$v'_A = arg \min_{v \in ORCA_A^\tau} \|v - v_A^{pref}\| \tag{2-12}$$

While VO/RVO/HRVO solves a uniform sampling problem, ORCA could be efficiently solved by linear programming as $ORCA_A^\tau$ is a convex region bounded by linear constraints induced by the half-planes of permitted velocities with respect to each of the other agents, which leads to fast running times and smooth. A extension of OCRA to 3D workspace for multiple simple-airplanes with kinematics and dynamics constrains is investigated in [107]

2-2 Potential Fields Based Approach

The other important type of collision avoidance approach is based on artificial potential fields which was originally proposed to deal with on-line collision avoidance for a robot with proximity sensors in [102]. This method generates an artificial potential field $U(p)$ that reflects (locally) the structure of the free space. It consists of an attractive potential function $U_{att}(p)$ to pull agent towards the goal and a repulsive potential $U_{rep}(p)$ around obstacles to push agent away from obstacles. The new velocity that navigates the agent to its destination free of collision is determined by the negative gradient of the total PFs.

$$v = -\nabla U(p) \tag{2-13}$$

The attraction potential function U_{att} is of the form:

$$U_{att}(p) = K_{att}\rho^m(p, p_g) \tag{2-14}$$

where K_{att} is a positive constant scaling factor and $\rho(p, p_g) = \|p - p_g\|$ is the Euclidean distance between agent's the current position p and its goal position p_g. $U_{att}(a)$ has unique minimum at p_g.

For $m = 1$, the attractive potential is conic in shape and the resulting attractive force,

$$V_{att}(p) = -K_{att}\frac{p - p_g}{\|p - p_g\|}$$

, is a unit vector directed towards the goal except for $p = p_g$, where U_{att} is singular at the goal(not stable) that might cause oscillations. For $m = 2$, the attractive potential is parabolic in shape and the attraction force,

$$V_{att}(p) = -K_{att}(p_g - p)$$

, is a vector directed towards the goal with magnitude linearly related to the distance from p to p_g. The attraction potential function $U_{att}(p)$ converges linearly to zero as p approaches

p_g which result in good stability but as p is far away from q_g, the attraction force grows with bound. A good idea would be a hybrid method to take the advantage of both parabolic and conic wells, that is choosing $m = 1$ when $\rho(p, p_g) > d$ and $m = 1$ otherwise.

An example of the repulsive potential U_{rep} could be chosen as:

$$U_{rep} = \begin{cases} \frac{1}{2} K_{rep} (\frac{1}{\rho(p)} - \frac{1}{\rho_0})^2 & \text{if } \rho(p) \leq \rho_0 \\ 0 & \text{if } \rho(p) > \rho_0 \end{cases} \tag{2-15}$$

Then a harmonic potential filed approach is demonstrated in [115]. In this paper, the control input is determined by three components: stationary obstacle avoidance component uo_i, purposed filed (PRF) component ug_i and conflict resolving field (CRF) component uc_i. As the result of ug_i and uo_i component, each individual agent is capable of safely reaching its target in the absence of the others. The CRF component uc_i intended to avoid other moving agents is the combined effort of pushing the other agent away and moving out of the way:

$$uc_i = ucr_i + uct_i \tag{2-16}$$

where ucr_i is the radius agent collision prevent component for the agent to prevent others moving towards it. uct_i is the deadlock prevent tangent component to avoid obstacles from block agent's way moving towards the target. (This component is a must especially for head-on case.) The proof of globally, asymptotically convergence to agents' respective destination without collision can be derived based on the theorem by LaSalle about extensions of Lyapunovs second method [127].

Chapter 3

Problem Formulation

3-1 Problem Statement

A group of n identical agents freely moving in 3-dimensional space is considered. For each agent i, p_i denotes its position, $v_i = \dot{p}_i$ is the velocity and $a_i = \ddot{p}_i$ is its acceleration. In every time step, a new force(acceleration) input for the agent must be derived based on local observations of the environment, so that the agent is free of collisions with other agents and the obstacles, while making progress towards a goal. While motion planning refers to the problem of finding a collision-free motion for a robot from a given start pose to a given destination pose, collision avoidance focuses on local scenario when obstacles are close enough for the agent to be able to detect them. Each agent is assumed to have perfect sensing, and is able to infer the exact shape, position and velocity of static obstacles and other agents in the same environment. While dynamic obstacles move passively through the environment without perception of their surroundings, the agents do perceive each other, and actively adapt their motions accordingly. Each agent runs its collision detection and avoidance protocol independently, no communication among agents nor a central coordination is needed, the computation could be processed fully parallelized. All the agents moving on the common workspace are assumed to follow the same decision-making strategy.

3-2 Dynamics for Single Agent

In this thesis, we consider multi-agent systems composed of identical agents as discussed in a previous report [1]. Following the presentation in [1], this section addresses the properties of each individual agent for multi-agent system considered in this thesis. Also, the backstepping control method adopted to control the multi-agent system and in which we embed also the collision-avoidance protocol is introduced.

3-2-1 Agent Dynamics

Each agent is assumed to have simple circular shape with a radius R_0. Furthermore, we assume that the agent is holonomic, i.e. there is no constraint of moving in any direction that is depended by the heading of agent, such that the control input of each robot is simply given by a two-dimensional velocity vector. The mass of agent $m(t)$ is time varying as the agents can possibly pick up or drop off objects during their whole assignment. The equation of motion of the ith agent can be expressed as below

$$\begin{bmatrix} \dot{x}_i \\ \ddot{x}_i \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I}{m_i(t)} \end{bmatrix} u_i(t) \tag{3-1}$$

There is an individual speed limit $v_{lim,i}$ as the upper constraint of $i$the agent's speed $v_i(t)$.

3-2-2 Backstepping Control

In this section, we will explain the control strategy used in simulation in absence of collision avoidance protocol. To address the constraint of maximum speed of each agent, an appropriate saturation function is introduced. The other thing we need to take into consideration is that the agent's mass is time varying in general. Although a force input is required for control purposes, it is more intuitive to provide a velocity input. Based on the issues above, the control method we chose is backstepping control, which is suited for a system that can be written as strict-feedback form. Especially, a backstepping controller allows to design a velocity control which gives a reference velocity based on the current position and goal of agents is to simply track the reference velocity by means of the preferred force input. We assume that the mass $m_i(t)$ of the ith agent at time t is known or can be measured and the velocity of agents shown in Equation (3-1) could be measured.

The idea of backstepping controller is to stabilize the lowest derivative (velocity in our case) with a feedback law and then step back until all states are stabilized and control input (force) is found. The first step is to calculate a reference velocity $\overline{\dot{x}}_i(t)$

$$\overline{\dot{x}}_i(t) = \underset{v_{lim,i}}{\text{sat}} [(\overline{r}_i - x_i(t))k_1] \tag{3-2}$$

where $\overline{r}_i$ is the goal position and k_1 is a gain for the difference between current location and destination. The next step is to obtain the input $u_i(t)$ based on the reference velocity $\overline{\dot{x}}_i(t)$ following Equation (3-2).

$$u_i(t) = (\overline{\dot{x}}_i(t) - \dot{x}_i(t))k_2 m_i(t) = (\underset{v_{lim,i}}{\text{sat}} [(\overline{r}_i - x_i(t))k_1] - \dot{x}_i)k_2 m_i(t) \tag{3-3}$$

where $x_i(t)$ and $\dot{x}_i(t)$ are the position and velocity of agent i in time t respectively and k_2 is the gain for the difference between reference we calculated and the actual velocity.

The state-space representation for each agent with the aforementioned input can be written as

$$\begin{bmatrix} \dot{x}_i \\ \ddot{x}_i \end{bmatrix} = \begin{bmatrix} 0 & I \\ -k_1 k_2 I & -k_2 I \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} k_1 k_2 \overline{r}_i \tag{3-4}$$

where 0_2 is the zero matrix of size 2×2 and I_2 is the 2×2 identity matrix. The eigenvalues of the system are given by

$$\lambda_{1,2} = \frac{-k_2 \pm \sqrt{k_2^2 - 4k_1k_2}}{2} \tag{3-5}$$

both λ_1 and λ_2 have algebraic multiplicity of 2. As it shown in the equation, the eigenvalues of the system are determined by the value of gains k_1 and k_2 only. The time varying mass is cancelled out in the new dynamic equation, in other words the behavior of the agent with a backstepping controller is irrelevant with the time varying mass of the agent and the overall system is linear time invariant (LTI).

3-3 Procedure

The overall process to accomplish our task is illustrated by the flow chart shown in Figure 3-1. The preferred velocity v_{pref} is simply a vector targeted to the goal of agents with maximum speed of the agent.

$$V_{pref,i} = v_{lim,i} \frac{P_{i,goal} - P_i}{\|P_{i,goal} - P_i\|} \tag{3-6}$$

The policy for obstacle detection and how reference velocity could be generated based on two different approaches would be explained in the following two chapters.

3-4 Evaluation Metrics

The following metrics can be introduced to analyze the collision avoidance behavior obtained from the simulations:

1. **Total Traveled Distance (TTD)**: The total distance traveled by an agent from initial position towards goal position employed collision avoidance maneuver is defined as:

$$TTD = \sum_{k^i<k<k^f} \|p(k) - p(k-1)\| \tag{3-7}$$

   with k^i being the first simulation step k starting the navigation maneuver from their initial positions, and k^f being the final simulation step k when the agent is at a distance equal or less to 0.01m from their final destinations.

2. **Time Steps to completion(TS)**: The time steps for an agent to complete the whole navigation procedure, which is defined as:

$$TS = k^f - k^i \tag{3-8}$$

3. **Total acceleration (T_{acc})**:The total acceleration over the simulation is calculated as:

$$A = \sum_{k^i<k<k^f} \| \frac{v(k) - v(k-1)}{dt} \| \tag{3-9}$$

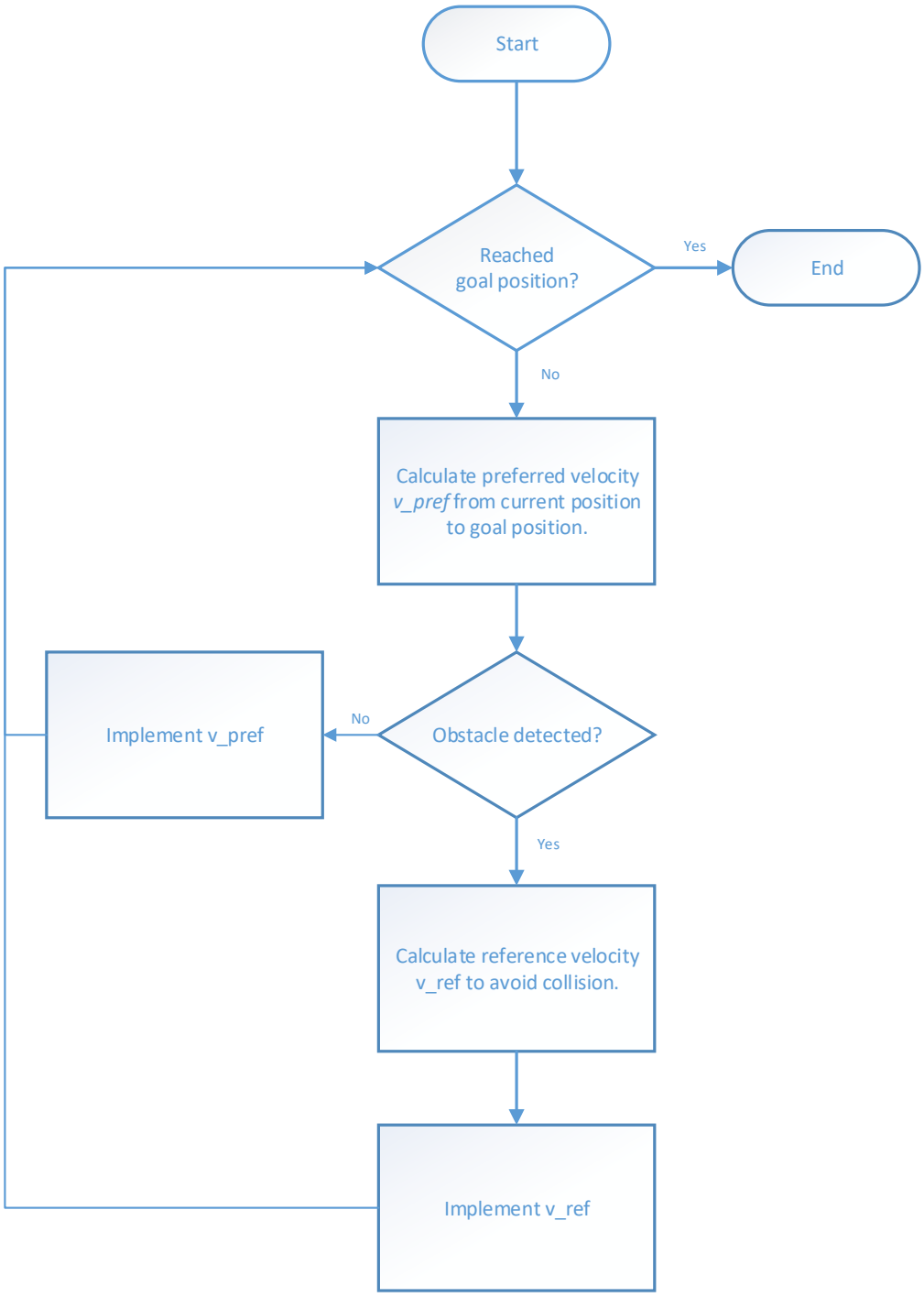


Figure 3-1: General Block Scheme

4. **Proportion of Collisions**: The proportion of collisions over the whole simulation is defined as:

$$C = \sum_{k^i<k<k^f} c(k) \tag{3-10}$$

where

$$c(k) = \begin{cases} 1 & \|p(k) - p_o(k)\| < 0.5m \\ 0 & \text{otherwise} \end{cases} \tag{3-11}$$

$p_o(k)$ refers to the position of nearest obstacle.

Other evaluation metrics could be computation time per time step and Minimum Distance (MD) from closest obstacle during the whole iteration.

Chapter 4

Collision Avoidance Protocol 1 -Rotation Based Method

The proposed collision avoidance algorithm is inspired by human behavior, that is turn aside when an obstacle is spotted. As long as there is no collision detected, the agent would go toward its goal position directly. When obstacle is detected, the collision avoidance protocol is activated. In our case all agents prefer to turn right (which results in counterclockwise rotations) to avoid collision.

4-1 Collision Detection

The proper detection zone is defined as Figure 4-1. The yellow circle represent the circular shape agent with radius R_0, the pink circle and blue circle refer to the static detection area with radius R_c and active detection area which is dependent on the speed of agent. The relation of active detection radius R_{cz} and speed is given as:

$$R_{cz} = R_c + \beta v(t) \tag{4-1}$$

For an agent $\mathcal{A}_i$, an obstacle $\mathcal{O}_i$ within a distance $d_{ij} < R_{cz}$ and at angle $\lambda_{ij} < \theta$ will be detected by sensors, where

$$\lambda_{ij} = \arccos\left(\frac{(p_i - p_{i,g}) \cdot (p_j - p_i)}{\|p_i - p_{i,g}\|\|p_j - p_i\|}\right) \tag{4-2}$$

p_i and p_j are the position of agents $\mathcal{A}_i$ and $\mathcal{A}_j$, and $p_{i,g}$ is the goal position for $\mathcal{A}_i$. In Figure 4-1, the detection zone is depicted by the pink area and red sector of range 2θ. When a possible collision is detected the agent scans the whole active detection area and will make a move around the obstacle which is closest to the agent until the collision detection zone is obstacle free. When $\theta = 90°$, it means that the agent will not react to the obstacle "behind" it, regardless it is within the detection shell or not. Note that this algorithm is not suited for

Figure 4-1: Collision Detection Zone of Rotation Based Method

detecting the whole blue circle, when $\theta = 180°$, as the agent will keep spinning around the obstacle when it is detected but the relative distance would never be larger than the radius of collision detection zone.

4-2 Collision Resolution

The collision avoidance manoeuvre for agent $\mathcal{A}_1$ to avoid a selected agent $\mathcal{A}_2$ is illustrated by Figure 4-2. As shown in the Figure 4-2, $\mathcal{A}_1$ detects an obstacle, $\mathcal{A}_2$, then $\mathcal{A}_1$ scans its entire surroundings, the whole blue circle in the Figure. A green circle with radius R_{dist} and central position of $\mathcal{A}_2$ as the center is drawn, where R_{dist} refers to the preferred distance from obstacle, which is slightly smaller than the radius of detection zone R_{cz}. When rotating the vector from $\mathcal{A}_2$ to $\mathcal{A}_1$, $(p_1 - p_2)$ counterclockwise with an angle α, there would be a point c_i in green circle lying on the rotated vector or its extension line. According to this cross point, a new reference velocity with the same direction as vector $(c_i - p_i)$ could be obtained. The margin of reference velocity is fixed as maximum speed. Both α and R_{dist} play important part of the collision avoidance behavior.

4-3 Choice of Parameters

To ensure the collision free trajectory and leave a safe distance between agent and obstacle, the collision avoidance maneuver is designed with a desired behavior that the new velocity would pull away the agent from selected obstacle when their distance is smaller than the preferred

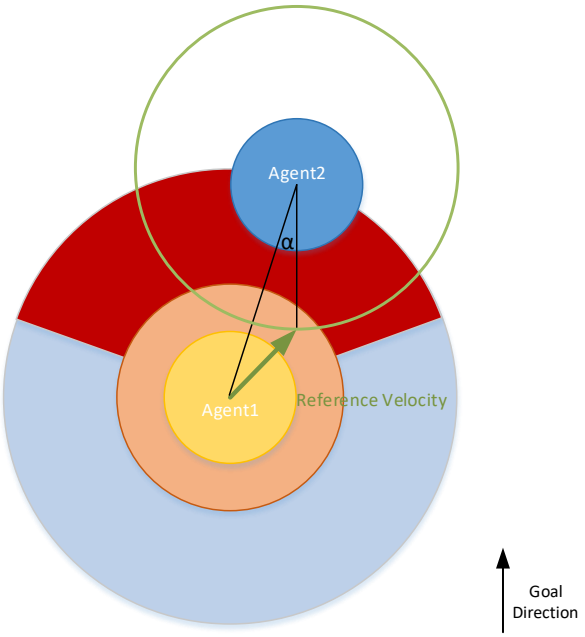


Figure 4-2: Collision Avoidance Maneuver of Rotation Based Method

Algorithm 1 Collision Avoidance Maneuver of Rotation Based Method

Given agents set $\mathcal{A}$ whose position p, velocity v and shape information could be obtained from sensor.

loop
  for all $\mathcal{A}_i \in \mathcal{A}$ with position p_i and velocity v_i **do**
    for all $\mathcal{A}_j \in \mathcal{A}$ **and** $j \neq i$ **do**
      if $\mathcal{A}_j$ with distance $\|P_{ji}\| \leq R_{cz}$ and angle $\lambda_{ij} \leq \theta$. **then**
        Collision detected sign $cd_j = 1$
      end if
    end for
    for all $cd_j = 1$ **do**
      Pick a v_{ref} based on the current position as illustrated in collision resolution section
    end for
    Compute control input from $v_{new,i}$ and apply it to the actuator of $\mathcal{A}_i$.
  end for
end loop

distance R_{dist}. In other words, we would like to make sure the minimum distance between agent and obstacle is always larger than preferred distance. For a list of fixed parameters shown in Table below, collision avoidance would be achieved for a certain range of rotation angle for collision protocol α. Both analytically deduction and simulation for obtain the constraint of α is given as following.

Parameter	Value	Description
R_0	0.15m	Radius of Agent
dt	0.05s	Simulation Sample Time
R_{cz}	0.8m	Radius of Detection Zone
R_{dist}	0.5m	Preferred Distance From Obstacle
θ	90°	Detection Range Angle
α		Rotation Angle for Collision Protocol

Table 4-1: List of Parameters in Rotation Based Collision Avoidance Method

4-3-1 Tuning Rotation Angle α

To analyse the effect of varying α, an extensive number of of simulations are carried out. The result is shown in Figure 4-3. All of them following a simple head-on scenario where start positions are $[0, 1]$ and $[0, -1]$ separately and the corresponding goal positions are $[0, -1]$ and $[0, 1]$. In every simulation, both agents have a speed limit of 1m/s and in every time step we renew the velocity direction only and keep the same velocity margin when the goal position is relatively far away. In these simulations, the gain for difference of position k_1 in backstepping control is set as 4, the agent would slow down when the goal position is within a distance of 0.25m. The gain for difference of velocity k_2 is chosen as 20, which is equal to $1/dt$, so that the tracking efficiency is 100%, the reference velocity we generate from collision avoidance protocol is the same as the one implemented by actuator.

α	5°	10°	15°	30°	45°	60°	75°	80°	85°
TTD(m)	2.2530	2.1572	2.1402	2.1204	2.1043	2.0849	2.0693	2.0600	1.100
MD(m)	0.4667	0.4759	0.4885	0.4845	0.4378	0.3819	0.3174	0.3063	0.2847
TS	55	53	52	52	52	51	51	51	null
T_{acc}	200.22	107.10	89.69	84.04	87.16	82.72	87.56	77.61	52.63

Table 4-2: Value of Evaluation Metrics with Different Rotation Angle α

Through our simulations, we can see that collision-free path could be obtained for all listed rotation angle α except for $\alpha = 85°$. As α increases, the angle between the vector from agent to obstacle p_{ji} and new velocity decreases, which leads to a more preservative reaction to avoid obstacle, it could also be reflected by a smaller total traveled distance(TTD). When $\alpha = 5°$, the avoidance behavior is quite aggressive and as a consequence oscillation is spotted. In other words, the relative distance changing between values larger than preferred distance $R_{dist} = 0.5$m and values smaller than 0.5m repeatedly several times. This result in a longer total traveled distance(TTD), total steps(TS) and total acceleration(T_{acc}). This problem could be solved by choosing a smaller time step, or slow down when collision detected. It

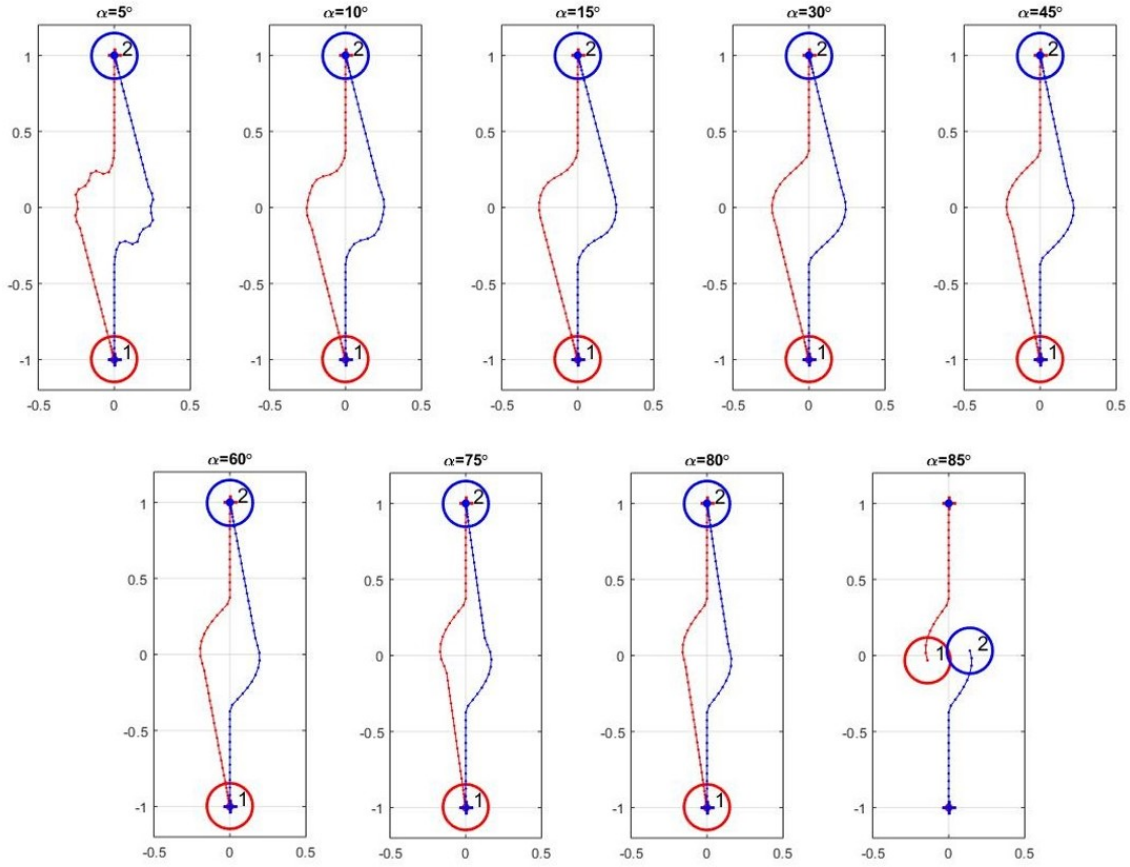


Figure 4-3: Head-on Trajectory for Different Rotation Angle α

especially works for $2r_0 \sin(\alpha/2) >> v_i dt$, when position c_i is not reached in next step. Another modifying suggestion would be using the central point of agent and obstacle instead the position of obstacle as the rotation center in collision resolution step. When $\alpha > 30°$, the minimum distance(MD) is decreasing when α increases, and it reflects a more conservative and dangerous reaction to avoid obstacle.

4-3-2 Choosing Detection Radius R_{cz} and Rotation Radius R_{dist}

The performance is also tested for cross case where start positions are $[1, 2]$ and $[1, -2]$ separately and the corresponding goal positions are $[-1, -2]$ and $[-1, 2]$ as well as side scenario where start positions are $[1, -2]$ and $[-1, -2]$ separately and the corresponding goal positions are $[-1, 2]$ and $[1, 2]$. The maximum speed for both agents is 1m/s for both scenarios. Using the parameters in Table 4-1 and setting $\alpha = 15°$, the trajectory is shown as Figure 4-4. There

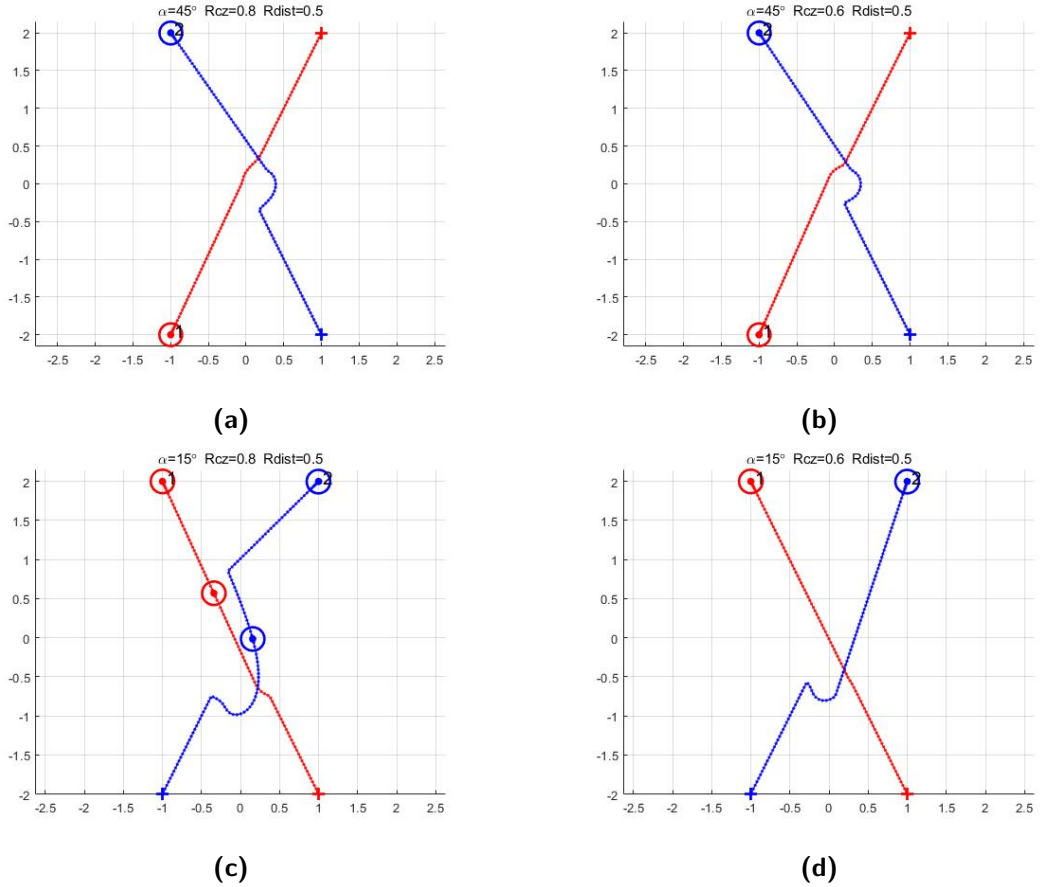


Figure 4-4: Cross and Side Trajectory for Rotation Based Method with $\alpha = 15°$ and left: $R_c = 0.8m$, right: $R_c = 0.6m$

are no obvious difference for different R_{dist} in cross case on the top but the trajectory for side case is varying. Due to the counterclockwise rotation policy, agent 1 and agent 2 detected each other when distance between them is smaller than $R_{cz} = 0.8$m, the new velocity derived based on their relative position, which is horizontal, has the opposite direction. As suggested

on the figure, the turning rate from goal direction to new velocity of agent 1 is much smaller than agent 2, which makes agent 2 having a longer recovering from the effect of obstacle. As the position at time step equals to 60 (t=3s) marked in the figure, agent 1 goes towards to its destination directly which implies that no obstacle is within the detection shell as agent 1 is "behind" it. As for agent 2, the angle between the vector from agent 2 to its target position $p_{i,g}$ and the vector from agent 2 to obstacle agent 1 p_{21} is smaller than detection range angle $\theta = 90°$. As the relative distance is 0.6639m, agent 2 regards agent 1 as an obstacle at t=3s. Agent 1 seems to be trapped by agent 2 for a certain long period. A feasible solution would be reduce detection zone radius R_{cz}. As shown in Figure 4-4d, when R_{cz} is reduced to 0.6m, a better performance is obtained.

4-3-3 Different Speed Scenarios

Two scenarios with different speed, head on and takeover case is present in this subsection. For the head-on case, agent with maximum speed 1m/s starts at [0,2] and goes towards [0,-2]. The other agent with maximum speed of 2m/s initiates at [0,-2] and the destination is [0,2]. For the takeover case, agent with maximum speed of 1m/s starts at [-1,0] and ends at [1,0], the other agent with maximum speed of 2m/s goes from [-2,0] towards [2,0]. The result is shown as Figure 4-5. For the most left column, the detection zone radius R_cz is chosen as a fixed number of 0.6m and preferred distance R_{dist} is fixed at 0.5m. The rest $R_{cz} = R_c + \beta v(t)$,and $R_{dist} = R_c + (\beta - 0.1)v(t)$ where $R_c = 0.4m$ and β increasing from 0.1 to 0.4. Collision free path is achieved for all simulations. As β increasing, the reactive distance increases as well as the minimum relative distance. When $\beta > 0.3$, the agent with high speed limit takes almost all the responsibility to avoid collision.

With a proper choice of rotation angle α and preferred distance R_{dist}, collision-free path could be guaranteed with collision avoidance protocol proposed above under the assumption that only one obstacle is present in the detection zone. To show this, we can get an intuition from geometric expect. In fact, let us work under the assumption that obstacle would follow the same protocol or not move. The agent would always go away from the obstacle if $\cos(\alpha) > \frac{d_c}{R_{dist}}$, where d_c the distance from agent to obstacle. So that the distance between agent and obstacle is guaranteed to be not smaller than d_c in next step so that the path wold be collision-free.

4-4 Extension to Multi-agent Case and 3-Dimensional Space

4-4-1 Remarks on Multi-agent Case

When there are more than one obstacle is within the detection shell, the agent only reacts to the nearest one. In case there are more than one obstacle with (almost)same distance, the agent react to the one with minimum angle to the right edge of detection zone.

4-4-2 3-Dimensional Space

In three-dimensional space, what we need to find is a rotation axis instead of a rotation center in two-dimensional space. One option would be using projection on a horizontal plane and

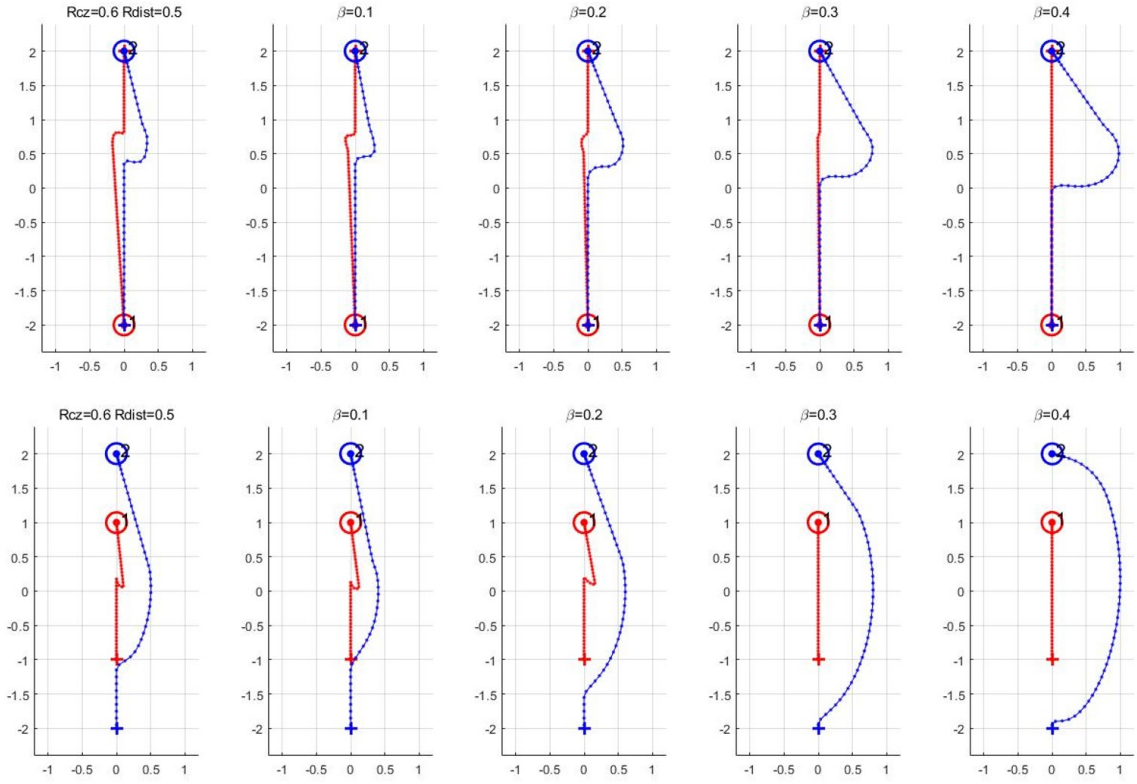


Figure 4-5: Head-on and Overtake Trajectory for Different β, the Speed Limit of Red Agent (Agent1) is 1m/s and 2m/s for Blue Agent(Agent2)

if the collision is avoided in that plane, a collision-free path would be guaranteed in three-dimensional space. An exception would be agent and obstacle have the same coordinates in x and y axis and the only difference is in z axis. In this case, the new velocity would point to the positive direction of x-axis for the agent on the top and point to the negative direction of x-axis for the agent on the bottom. The examples for two agents switch top and bottom positions [0,0,-1] and [0,0,1] as well as exchange diagonal positions [-1,-1,-1] and [1,1,1] are shown in Figure 4-6.

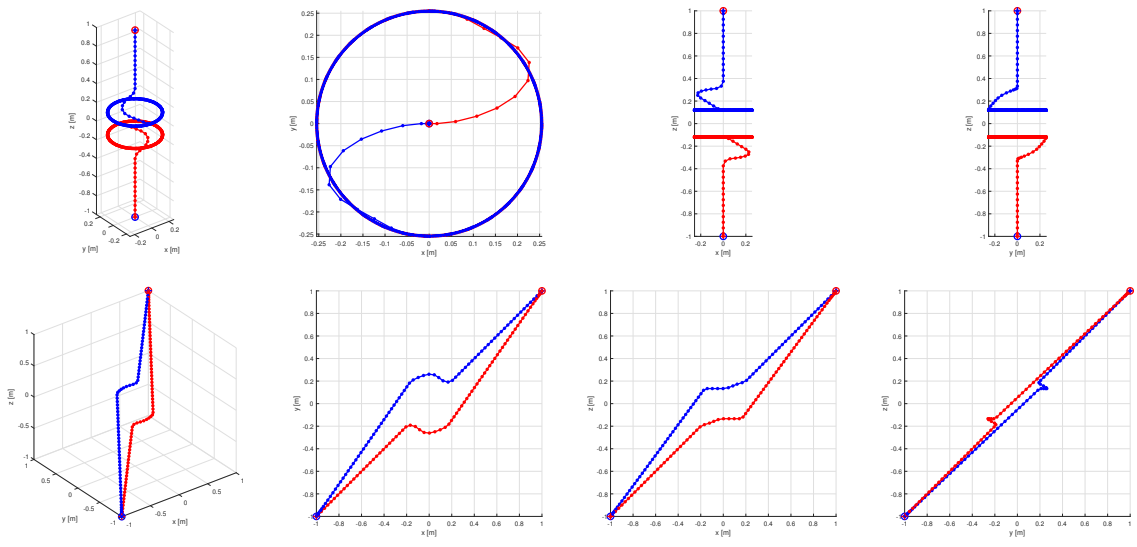


Figure 4-6: Trajectories of Two-agent Head on Scenarios for Rotation Based Method. Top line: exchange position between [0,0,-1] and [0,0,1]. Top line: exchange position between [-1,-1,-1] and [1,1,1]. From left column to right, view in 3D, in x-y plane, x-z plane and y-z plane

For the first scenario, the agents are stuck in a deadlock in z-direction and livelock in x-y plane. While the movement in z-direction is forbidden and the relative distance in x-y plane start from a value smaller than the rotation radius $R_{dist} = 0.5$m, the agent are trapped in a circle with the same radius. For the second scenario, the collision avoidance is successfully achieve by moving in the plane y=z and the movement in z-direction is retard as shown in the x-z plane. The resolution relies most on the behavior on horizontal plane which is similar to a 2D scenario.

4-5 Discussion

- For tuning the parameters, it is important to get an insight of how the parameters would affect the behavior of agents. This is crucial to understand how to choose the right parameters. Those parameters include rotation angle α, preferred distance R_{dist} and detection angle θ. The constrains to optimally choose detection angle θ have not been well studied yet and are left for future work. The trade off is between a preserve performance with smaller minimum distance but smaller total travelled distance and an aggressive reaction to collision with larger minimum distance and larger total travelled distance.

- Collision avoidance part is decoupled with navigation part, and collision avoidance maneuver may require abrupt changes in the velocity that may be unfeasible due to kinematic constraint of maximum turning rate and maximum input force. The switch from navigation to the goal position and collision avoidance leads to sudden change in velocity and not smooth trajectory. The result depends strongly on the geometric configuration of agents.
- The situation of a livelock with the agents trapped in a circle is spotted in 3D case. The reason may be due to failure of the 3D equivalent of the requirement for a feasible solution for 2D case namely: the smallest distance between start point and obstacle should always be larger than the preferred distance R_{dist} and the smallest distance between goal position and obstacle should always larger than detection distance R_{cz}.

4-6 Summary

In this chapter, a rotation based collision avoidance protocol is introduced. Through our simulations, we find that when there is only one obstacle need to take into consideration, the algorithm can always come up with a feasible collision-free path. The suggestion for extension to multi-agent and 3-dimensional case is also given in section 4-4. Some failure situation when the agents are trapped in a livelock is also shown.

Chapter 5

Collision Avoidance Protocol 2 -Potential Field Based Method

Another possible collision avoidance paradigm is based on the concept of artificial Potential Fields (PFs).This method creates a potential field function of the position of the agent that usually consists of an attractive potential component pointed to the goal and local repulsive potentials around detected obstacles. The next movement of the agent is determined by the gradient of the overall summed potential. This is a simple but effective way for collision resolution.

5-1 Potential Field Functions

In this thesis, the total potential function is formed by three parts

$$U = U_g + U_r + U_t \tag{5-1}$$

The navigation component U_g leads agent to its goal position

$$U_g = \begin{cases} K_g \|p_i - p_{i,g}\| & \text{if } \rho(p_i, p_{i,g}) \geq r_d \\ \frac{1}{2} K_g \|p_i - p_{i,g}\|^2 & \text{if } \rho(p_i, p_{i,g}) < r_d \end{cases} \tag{5-2}$$

The radius collision avoidance component is presented as

$$U_r = \frac{1}{2} \sum_j K_r (\frac{1}{\|p_i - p_j\|} - \frac{1}{\rho_o})^2, \quad \text{if } \rho(p_i, p_j) \leq \rho_0 \tag{5-3}$$

and a tangential collision avoidance part of the form

$$U_t = \frac{1}{2} \sum_j K_t (\frac{1}{\|p_i - p_j\|} - \frac{1}{\rho_o})^2 \times S_i(p_i, p_j), \quad \text{if } \rho(p_i, p_j) \leq \rho_0 \tag{5-4}$$

When the agent is close to the goal, within a radius of $r_d > 0$, the agent would slow down according to U_g. Constant gains $K_g \geq 0$, $K_r \geq 0$ and $K_t \geq 0$ are used to weight the influence of navigation and collision avoidance components. The negative gradient of the total PFs is used to navigate the agent to its destination while avoiding collision:

$$v = -\nabla U = -\nabla U_g - \nabla U_r - \nabla U_t \tag{5-5}$$

with

$$-\nabla U_g = -K_g \mu_0 \nabla \rho(p_i, p_{i,g}) \tag{5-6}$$

$$-\nabla U_r = \sum_j K_r (\frac{1}{\rho(p_i, p_j)} - \frac{1}{\rho_0})(\frac{1}{\rho^2(p_i, p_j)}) \nabla \rho(p_i, p_j) \tag{5-7}$$

$$-\nabla U_t = \sum_j K_t (\frac{1}{\rho(p_i, p_j)} - \frac{1}{\rho_0})(\frac{1}{\rho^2(p_i, p_j)}) \nabla \times S_i(p_i, p_j)), \quad \nabla \cdot S_i \equiv 0 \tag{5-8}$$

where $\mu_0 = \min(1, \frac{\rho(p_i - p_{i,g})}{r_d})$ is a factor to slow down agent when it is close to destination,$\nabla \cdot$ is the divergence operator, $-\nabla \rho(p_i, p_{i,g})$ is a unit vector directed to $p_{i,g}$ from p_i, $\nabla \rho(p_i, p_j)$ is a unit vector directed to p_i from p_j, and S_i is a vector potential field selected so that its gauge is zero.

$$\nabla \rho(p_i, p_j)^T \nabla \times S_i(p_i, p_j) \equiv 0 \tag{5-9}$$

When the obstacle is on the edge of the detection shell, ($\rho(p_i, p_j) = \rho_0$), the repulsive function starts from zero and increasing as the agent approaching obstacle. The vector potential field (S_i) can only generate a tangent circulating field. For the local tangent fields to form a continuous global tangential action that has the potential to push the interacting agents out of each others way and prevent deadlock, all the individual tangent fields must circulate along the same direction. [115]
For 2D scenario, the tangential component for avoiding collision could be simple chosen as of the direction as rotated 90° counterclockwise of vector $\nabla \rho(p_i, p_j)$.

$$\nabla \times S_i(p_i, p_j) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \nabla \rho(p_i, p_j) \quad , \quad \forall j \in \mathcal{O}_i \tag{5-10}$$

Algorithm 2 Collision Avoidance Maneuver of Potential Fields Based Method

Given agents set $\mathcal{A}$ whose position p, velocity v and shape information could be obtained from sensor.
loop
  for all $\mathcal{A}_i \in \mathcal{A}$ with position p_i and velocity v_i **do**
    for all $\mathcal{A}_j \in \mathcal{A}$ **and** $j \neq i$ **do**
      if $\mathcal{A}_j$ within distance $\|P_{ji}\| \leq R_{cz}$ **then**
        Collision detected sign $cd_j = 1$
      end if
    end for
    for all $cd_j = 1$ **do**
      Calculate v_{ref} according to equation (5-5)
    end for
    Compute control input from $v_{new,i}$ and apply it to the actuator of $\mathcal{A}_i$.
  end for
end loop

5-2 Tuning Parameters

When a possible collision is detected, the new velocity of agent $\mathcal{A}_i$ is composed by three components pointing in different directions, namely

1. v_g: a unit vector directed towards goal position $p_{i,g}$ from current position p_i
2. v_r: a unit vector directed towards current position p_i from obstacle $\mathcal{O}_j$'s position p_j.
3. v_t: a unit vector tangent to v_r.

The overall weighted velocity can be the result of different sums/combinations:

1. $v_g + v_r$
2. $v_g + v_t$
3. $v_g + v_r + v_t$

In order to accomplish our task, namely that all agents can would reach their goals eventually without collision with other agents or obstacles, the gains of the PFs should be chosen carefully based on some constrains for the norm of each component as pointed below.

5-2-1 Choosing Radius Avoidance Factor K_r

For the first protocol, to ensure collision avoidance, the total PFs affect needs to pull agent away from the obstacle when obstacle is within a small range $\|p_i - p_j\| < R_c$. In other words,

the norm of attractive component should be smaller than repulsive component when obstacle is close enough. The following relation should hold:

$$\| - \nabla U_g \| < \| - \nabla U_r \|, \quad \text{when} \ \ \|p_i - p_j\| < R_c \tag{5-11}$$

For a single obstacle condition, in the worst case that the distance to the destination is larger than r_d, Equation (5-11) could be rewritten as:

$$K_g < (\frac{1}{R_c} - \frac{1}{\rho_0})(\frac{1}{R_c^2})K_r \tag{5-12}$$

As a consequence, the negative gradient of the total PFs for $\mathcal{A}_i$ would pull it away from obstacle ($v_{new} \cdot p_{ij} < 0$) or result in a zero velocity when p_i, p_j and $p_{i,g}$ lie on the same line, a typical example would be head-on collisions. In either case, the agents move away from each other or not move at all, no collision would occur. As the deadlock-prone propriety of this PFs, some change should be introduced to ensure deadlock-free or zero-velocity-free path such as the solutions proposed in [112] and [128]. Alternatively, harmonic potential fields introduced in [2] could handle this issue well.

5-2-2 Choosing Tangent Avoidance Factor K_t

For most situation, it is not necessary to go completely backward (or slow down till zero-speed) to avoid obstacle, moving aside would be sufficient enough. For the second protocol, consider a tangent collision avoidance only. What we need is to make sure is that the generalized new velocity is out the velocity obstacle that $\sin(\angle(p_{ji}, v_{new,i})) > \frac{(r_i+r_j)}{\|p_{ji}\|}$. To make sure the agent would be repulsive from each other with a relative distance R_c, the following equation should hold:

$$\frac{\| - \nabla U_g \|}{\| - \nabla U_t \|} < \frac{\sqrt{R_c^2 - (r_i + r_j)^2}}{R_c} \tag{5-13}$$

Based on our proposed potential field functions, under the condition that the distance to the destination is larger that r_d, above Equation (5-13) could be rewritten as

$$K_g < \frac{\sqrt{R_c^2 - (r_i + r_j)^2}}{R_c}(\frac{1}{R_c} - \frac{1}{\rho_0})(\frac{1}{R_c^2})K_t \tag{5-14}$$

The simulation for head-on case with different value of K_t is carried as following. Predefined parameters are chosen as shown in Table 5-1 below, K_g is chosen as 1 so that the velocity margin would not have a sudden decrease once an obstacle is spotted.

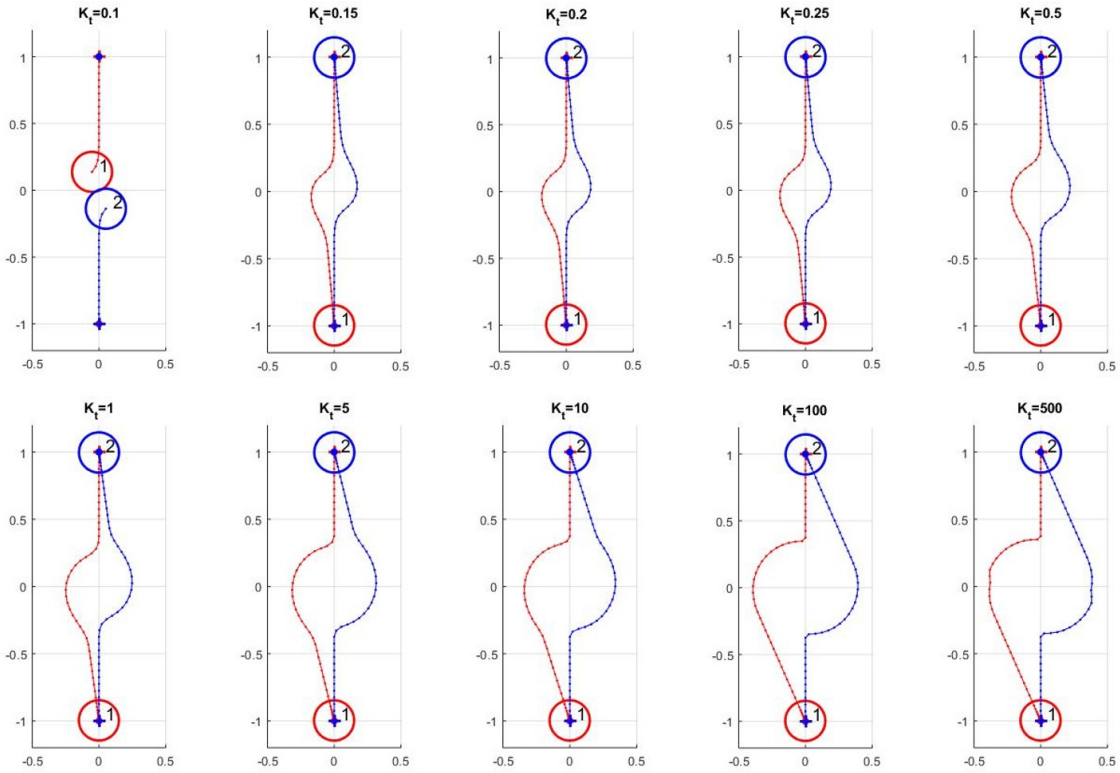


Figure 5-1: Head-on Trajectory for Different Tangent Avoidance Factor K_t

Parameter	Value	Description
R_0	0.15m	Radius of agent
dt	0.05s	Simulation sample time
ρ_0	0.8m	Detection zone radius
R_c	0.5m	Safe distance
r_d	0.25m	Slow down distance
K_g	1	Factor for navigation
K_r	0	Factor for avoidance in radius direction
K_t		Factor for avoidance in tangent direction

Table 5-1: Parameters in PFs based Collision Avoidance Maneuver

The start positions are fixed at $[0, 1]$ and $[0, -1]$ separately and the corresponding goal positions are $[0, -1]$ and $[0, 1]$. In every decision-making circle, both agents have a speed limit of 1m/s, the direction and margin of their velocity would be renewed. The result is shown in Figure 5-1.

K_t	0.15	0.2	0.25	0.5	1	5	10	100	500
TTD(m)	2.0845	2.0925	2.0992	2.1223	2.1488	2.2131	2.2409	2.3188	2.3339
MD(m)	0.3065	0.3307	0.3499	0.4107	0.4719	0.6000	0.6440	0.6984	0.7011
TS	51	51	52	52	52	54	54	56	56
T_{acc}	94.40	96.53	98.08	100.58	104.71	109.83	111.46	107.27	129.22

Table 5-2: Metrics in Potential Fields Based Collision Avoidance Method with Varying K_t

According to Equation 5-14 and the values in Table 5-1, the minimum distance is always larger than 0.3, the sum of radius of two agents, so that no collision would happen if $K_t/K_g > 1/2.4 = 0.4167$. Collision is only spotted when K_t is as small as 0.1 and minimum distance increases with K_t, thus the simulation result verified our argument. In fact, this is only a sufficient condition to guarantee collision-free path, Equation 5-14 means that no collision would take place in the future if the agent keep the new velocity infinitely. A larger K_t reflects a more aggressive behavior to avoid collision, which could be demonstrated by a longer total travelled distance and bigger total acceleration. When $K_t >> K_g$, the tangent avoidance component plays the majority role that the agent almost rotates around the origin of coordination, which is also the center of two agents.

To enhance the collision avoidance impact, it is reasonable to introduce a breaking element that allows the agent to slow down when it is too close to one another or an obstacle. What we do here is to introduce a scaling factor for navigation component

$$K_{sg} = \frac{\rho(p_i, p_j)}{\rho_0} \tag{5-15}$$

5-2-3 Combined Effort of K_r and K_t

An alternative option would be like the third protocol denotes, we combine radius component and tangent component. Potential field method with a tangent component is called harmonic potential field, whose deadlock free propriety is proved in [115]. In this case, some investigation for the tuning of factor K_r and K_t is carried. The same head-on scenario is carried on $K_r = 1/10, 1/3, 1/2.$ with K_t varying from 0.01 to 10.

As the result shown in Figure 5-2, the agents could avoid each other with a smaller K_t thanks to the introduce of K_r. When k_t is relatively small with regard to K_r, (for instance $K_t/K_r<1/3$), some low speed steps are spotted when the obstacle becomes detectable. When radius avoidance and navigation components play the majority role, the agent would come to a velocity close to zero in radius direction and as tangent component is relative small, the velocity in tangent direction is also minor. The third one combines the advantage and solves the problem of both case, while we avoid obstacle, it is not really aggressive, or easy to get stuck in deadlock. We have to make sure the proportions of K_g and K_r as well as K_t and K_r are within a certain range. That is to slow down when an obstacle is approaching, ($< \rho_0$), guarantee the agent would go away when obstacle is really close ($< R_c$). The overall velocity is preferred not be zero all the time to avoid deadlock. When K_g is relative big, the agent cannot slow down on time and lead to collide. When K_g is relative small, avoidance components would play an important part as long as the obstacle touch the detection shell,

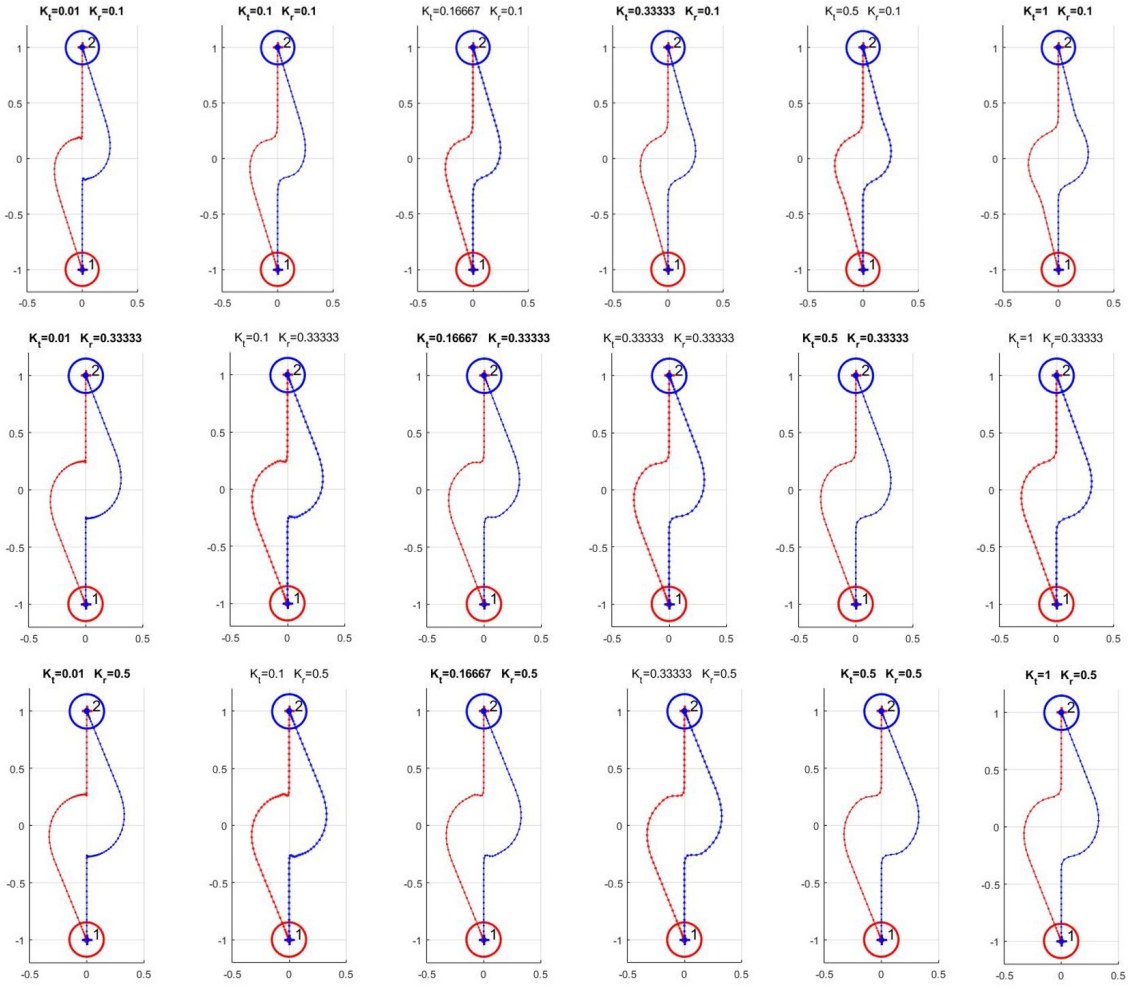


Figure 5-2: Head-on Trajectory for Different Tangent Avoidance Factor K_t and Radius Avoidance Factor K_r

turn backward aggressively, which is not really necessary and leads to problems such as oscillation, extra acceleration and not smooth trajectory.

Under the assumption that all agents follow the same protocol, it is necessary to make sure the reference velocity $u_i - u_j$ is collision-free based on the new velocity of each agent choose independently. Two agents with dynamics given in previous chapter 3-4 and both applying potential fields based collision avoidance protocol mentioned in this chapter will not collide.Indeed, consider two agent i and j at a distance $R_c < d_{ij} < R_{cz}$. From the perspective of agent i, it would leave away from obstacle when the distance is smaller that R_c according to Equation 5-12. With the help of tangent avoidance component, the agent would be deadlock free as the velocity is nonzero all the time.

5-2-4 Remarks on Multi-agent Case

For multi-agent case, when more than one obstacles are detected, the effect of avoidance component might be cancelled out. A solution would be to eliminate the navigation component.

5-3 3d Extension

For agents like UAVs that work in 3-Dimensional workspace, although no conflict in horizontal plane would be sufficient enough to make sure collision-free in general, it is still possible and beneficial to take advantage of the vertical dimension to avoid collisions. Aircraft like fixed-wing airplanes are more maneuverable vertically than in the horizontal plane. For 3D collision avoidance scenario, a trick problem would be that there is no such a simply "turn right priority" rule as there is for the 2D scenario. A straight forward explanation would be there are infinite orthogonal vector for a vector in 3D, it is hard to come up with an agreement for both agent to avoid collision. When there is no communication between two agents that would collide potentially, both agents probably perform the same reaction, both decided to turn north or to go up, make the situation even worse.

5-3-1 Proposed Modification

In this section, a method that combines both horizontal and vertical maneuver to guarantee collision-free is proposed, which does not rely on the communication between agents. Assume that the height dynamics is decoupled from the other state variable. The obstacle detection is separated in the horizontal plane.

$$\|p_i^H - p_j^H + (u_i^H - u_j^H)t\| \leq r_i + r_j, \forall t \in [0.\tau] \tag{5-16}$$

and the vertical component.

$$\|p_i^z - p_j^z + (u_i^z - u_j^z)t\| \leq h_i + h_j \tag{5-17}$$

when one of them is satisfied, we mark agent $\mathcal{A}_i$ as an obstacle for agent $\mathcal{A}_j$ and vice versa. Since we consider spherical agents regardless of posture (instead of fix-wing aircraft whose posture is determined by three angles, namely roll angle, pitch angle and yaw angle, so that a

unique tangent direction in the same plane of the agent posture could be determined), we have to propose some approach to select the tangent direction. When p_{ij} and v_{ij} are not parallel, the tangent direction could be determined by Gram-Scmidt orthonormalization. The unit tangent vector in the same plane with vector of relative position p_{ij} and vector of relative velocity v_{ij} could be derived as

$$v_t = \frac{v_{ij} - (v_r \cdot v_{ij})v_r}{\|v_{ij} - (v_r \cdot v_{ij})v_r\|} \tag{5-18}$$

where $v_r = [v_{rx}, v_{ry}, v_{rz}]^T$ is the unit vector with the same direction as p_{ij}.
When p_{ij} and v_{ij} are parallel, we can choose to avoid collision in horizontal plane or vertical plane. In horizontal plane, the policy is the same as 2-dimensional case. The unit tangent vector in horizontal plane ($v_{tz} = 0$) is derived as

$$v_{t,h} = \frac{1}{\sqrt{v_{rx}^2 + v_{ry}^2}} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} v_r = \frac{1}{\sqrt{v_{rx}^2 + v_{ry}^2}} \begin{bmatrix} -v_{rx} \\ v_{ry} \\ 0 \end{bmatrix} \tag{5-19}$$

In vertical plane, the tangent vector should be orthogonal to v_r and the tangent vector in horizontal plane $v_{t,h}$. That could be represent by the following equation

$$V_{t,v} = v_r \times v_{t,h} = \begin{bmatrix} 0 & -v_{rz} & v_{ry} \\ v_{rz} & 0 & -v_{rx} \\ -v_{ry} & v_{rx} & 0 \end{bmatrix} v_{t,h} = \begin{bmatrix} -v_{rz}v_{rx} \\ -v_{rz}v_{ry} \\ v_{rx}^2 + v_{ry}^2 \end{bmatrix} \tag{5-20}$$

Note that for vector p_{ij} and p_{ji}, the tangent vector we computed in vertical plane $V_{t,v}$ is the same. It is necessary to define a positive direction to distinguish them. Here we choose the sign $sign_t$ of the one with $(y > 0 | (y = 0 \& x > 0))$ as 1 other wise $(y < 0 | (y = 0 \& x < 0))$ as -1. So the final unit tangent vector could be given as

$$v_{t,v} = sign_t V_{t,v} \tag{5-21}$$

An exception would be $v_{rx} = v_{ry} = 0$. In this case, there are infinite number of horizontal tangent vector and no vertical tangent vector. We manually choose the tangent vector as $v_t = [\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0]^T$ for the agent on the top ($v_r = [0, 0, 1]^T$) and $v_t = [\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0]^T$ for the agent on the top ($v_r = [0, 0, -1]^T$).

5-3-2 Simulations of Simple 2-agent Scenarios

The result of two typical head-on scenarios involves z-direction, namely position exchange of[0,0,1] and [0,0,-1] as well as [1,1,1] and [-1,-1,-1], is shown in Figure 5-3. While the combined tangent vector would be better to avoid collision in a vertical plane (first and third line), the horizontal tangent vector (second and forth line) would take the advantage of three directions in different way.

The different K_t and K_r is further studied. In Figure 5-4, K_r is fixed at 1/3 while $K_t = \{1/6, 1/2, 3, 1/20, 1/100\}$. A larger K_t would lead to a larger spin trace in horizontal plane which result in a longer total travelled distance and total acceleration. While $K_t << K_r$, the

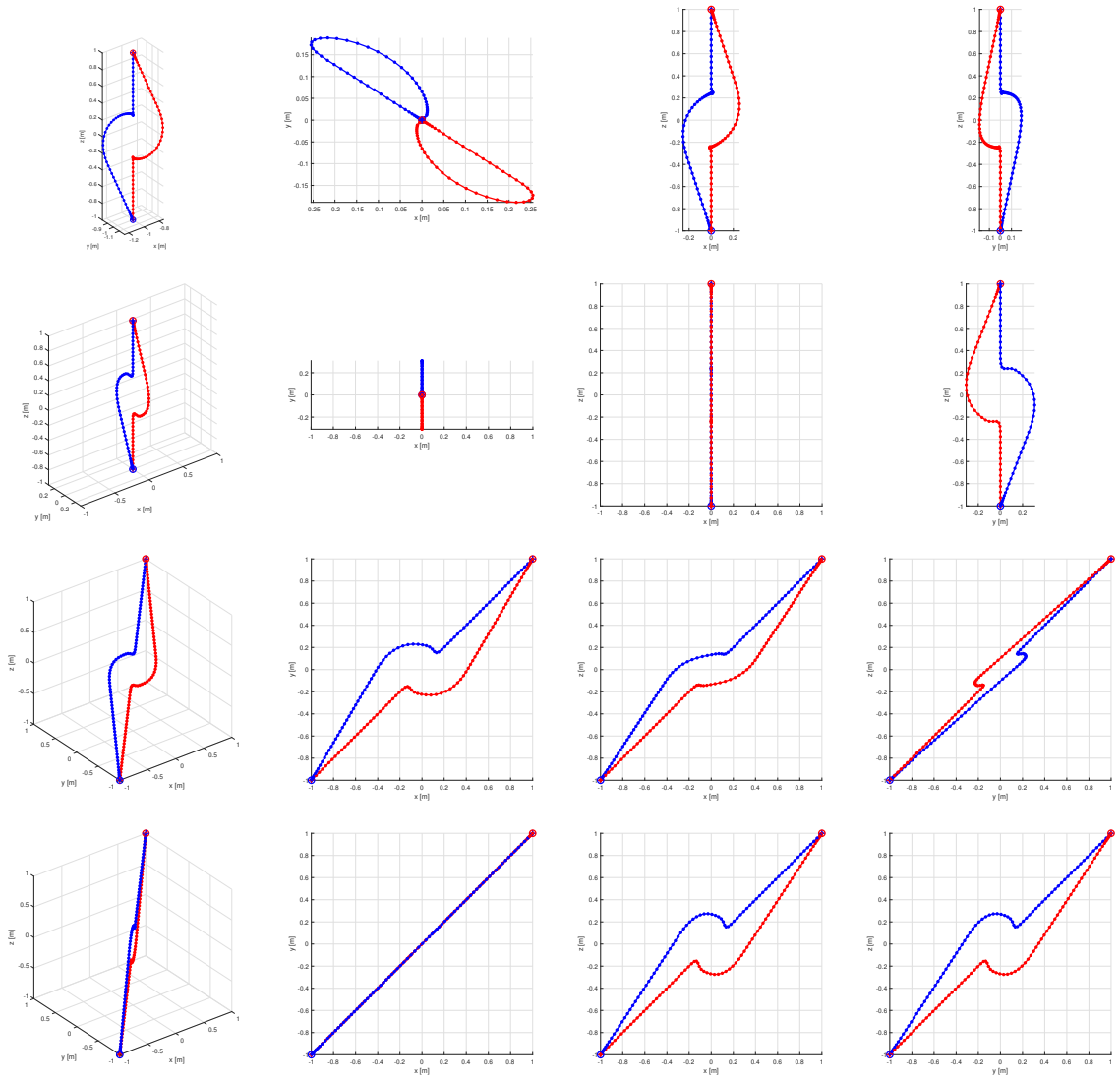


Figure 5-3: Trajectories of Two-agent Head on Scenarios for PFs Based Method.

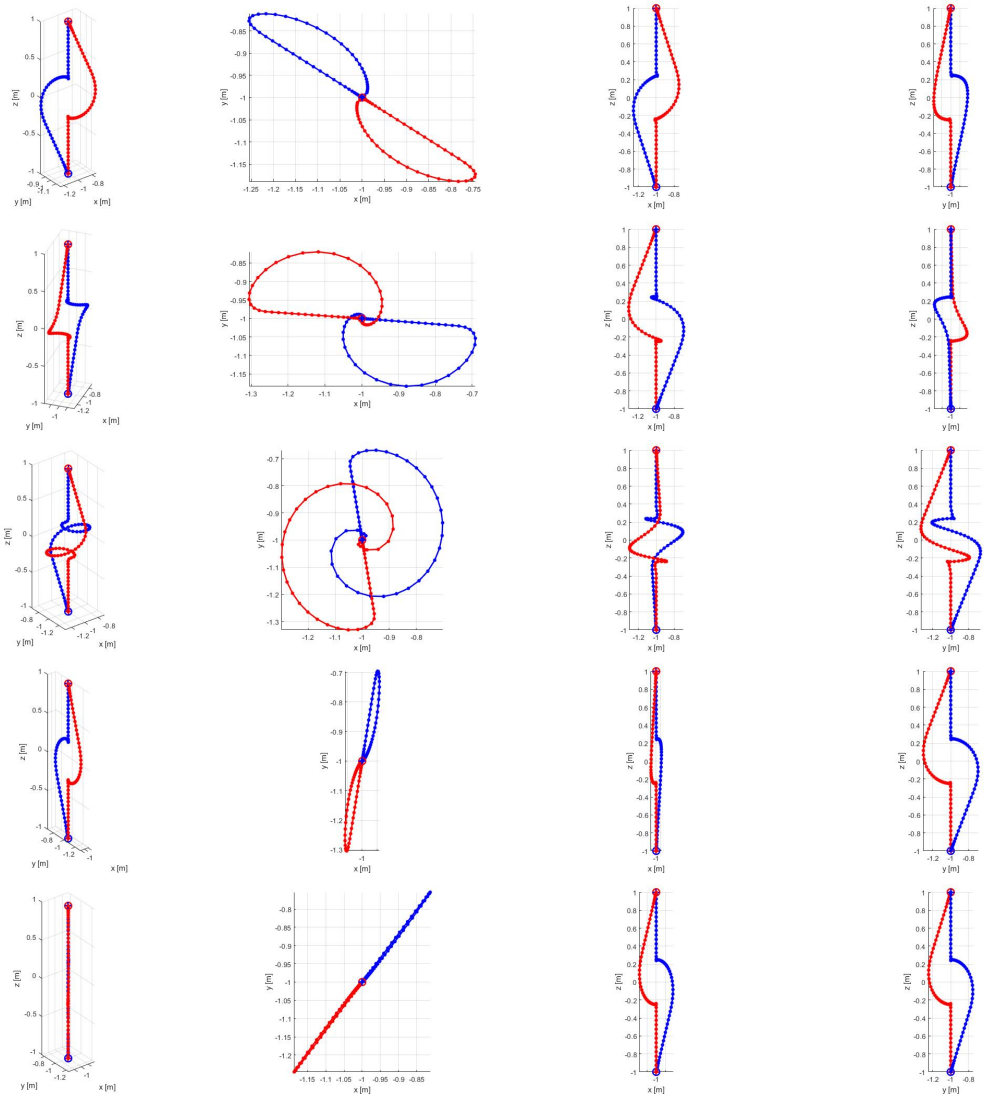


Figure 5-4: Trajectories of Two-agent Head on Scenarios for PFs Based Method Method with $K_t = \{1/6, 1/2, 3, 1/20, 1/100\}$ from Top to Bottom

impact of tangent avoidance component to move aside is little that the agent keeping a low speed for a long period to avoid deadlock so that total time step would be increased.

In Figure 5-5, K_t is fixed at $1/6$ while $K_r = \{1/3, 1, 6, 1/6, 1/15\}$. Similar to previous case, the ratio of K_t/K_r is positive correlation to the length of spin trace in x-y plane. As mentioned in 2D, K_r determines the minimum distance between agents. When $K_r >> K_g$, the obstacle would travel through the border of detection radius frequently that oscillation in the third and forth subfigures of the third line is spotted. In order to make sure agent would goes away from the obstacle before collision, K_r is advised to be chosen larger than $1/20$ according to Equation 5-12.

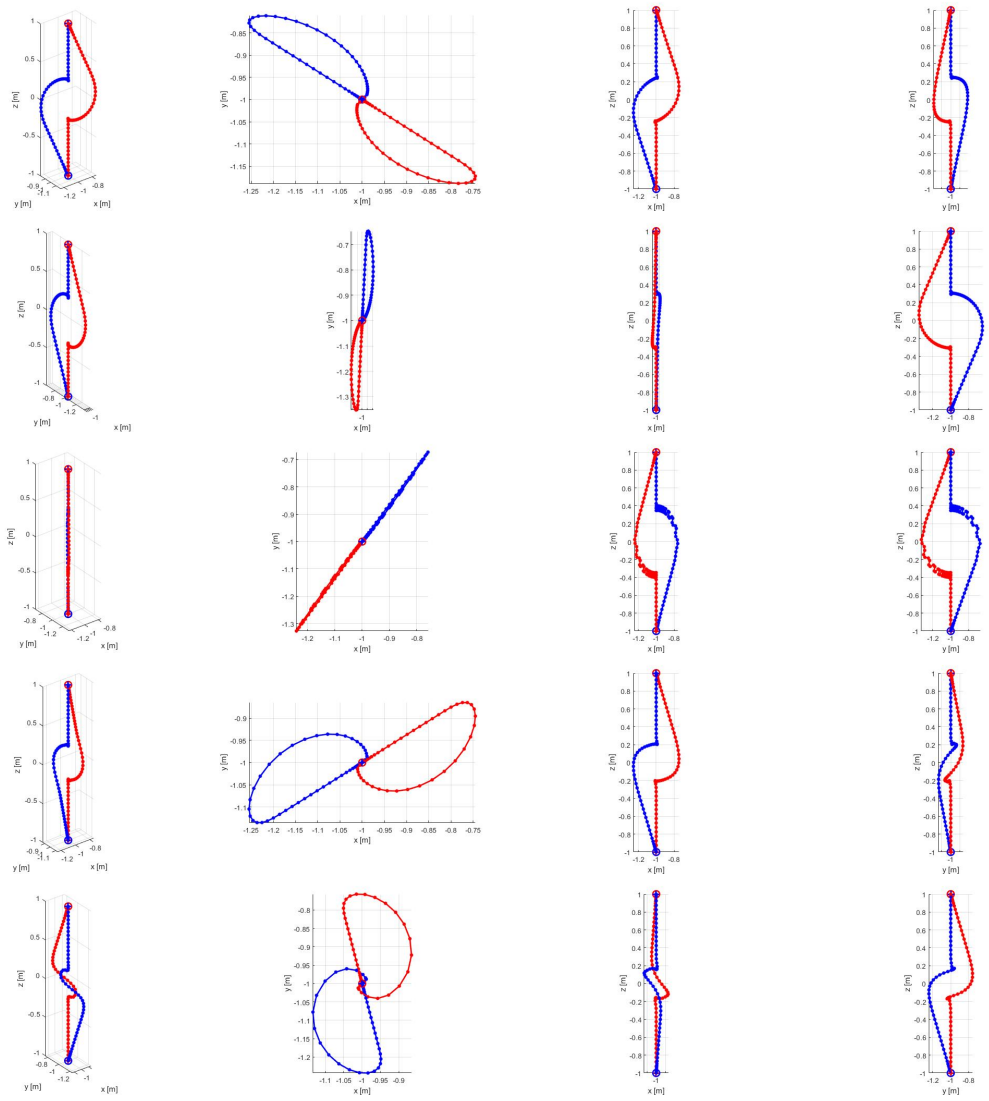


Figure 5-5: Trajectories of Two-agent Head on Scenarios for PFs Based Method with K_t=1/6 and $K_r = \{1/3, 1, 6, 1/6, 1/15\}$ from Top to Bottom

5-4 Discussion

1. Monotonic functions starting from zero on the edge of detection shell for avoidance component could smooth the impact of obstacles on the border of detection zone.

2. In our study, states vector is the position of agent only, posture(heading) is not taken into consideration. An improved fully 3-dimensional PF could be developed with posture information.

3. The potential field based algorithm discussed in this chapter only use the current position estimates of other agent to navigate and avoid collisions. The agents tend to do detours and slow down when near collision, which is a preserve action and sometimes not necessary but more safe when the motion of other agents is unknown. When considering also the velocity estimation or target information to add a predictive element, a more sufficient trajectory could be obtained.

5-5 Summary

In this chapter, a potential field based collision avoidance algorithm is introduced. The guarantee of collision avoidance is testified by some simple simulation in two-agent scenarios. The performance for multi-agent and 3-dimensional case will be further studied in next chapter.

Chapter 6

Simulation Results

This chapter investigates in depth the performance of the method proposed in Chapter 5. Especially 3D scenarios.

6-1 Simulation Scenarios

6-1-1 Two-Agent Scenarios

The simulation is carried on MATLAB. To start with, four typical two-agent scenarios utilized in [118] are introduced to validate the proposed method.

1. **Head-on**: An agent starts from $[0, 2, 0]$m and aims at $[0, -2, 0]$m, and vice-versa for the other agent. Both of the agents have a maximum speed limit of 1 m/s.
2. **Cross**: An agent starts at $[1, 2, 0]$m and goes towards position $[-1, -2, 0]$m. The other agent starts at $[1, -2, 0]$m and with the goal position $[-1, 2, 0]$m. Their maximum speed are both 1 m/s while their trajectories will cross at the origin without collision avoidance strategy.
3. **Side**: An agent starts at $[1, -2, 0]$m with the target position $[-1, 2, 0]$m. The other agent goes from $[-1, -2, 0]$m and aims to go to $[1, 2, 0]$m. Maximum speed for both agents are 1m/s.
4. **Overtake**; An agent starts at $[0, -1, 0]$m and ends at $[0, 1, 0]$m with a speed limit of 1 m/s. The other agent initiates at $[0, -2, 0]$m and its destination is $[0, 2, 0]$m. The speed limit is 2 m/s.

Based on the discussion about the choice of factors in the previous chapter, when only K_t is used (and not K_r), the ratio of K_g/K_t should be smaller than 2.4. The following result shown in Figure 6-1 is under the condition that $K_g = 1$, $K_t = 1/2 = 0.5$ and $K_r = 0$. Note

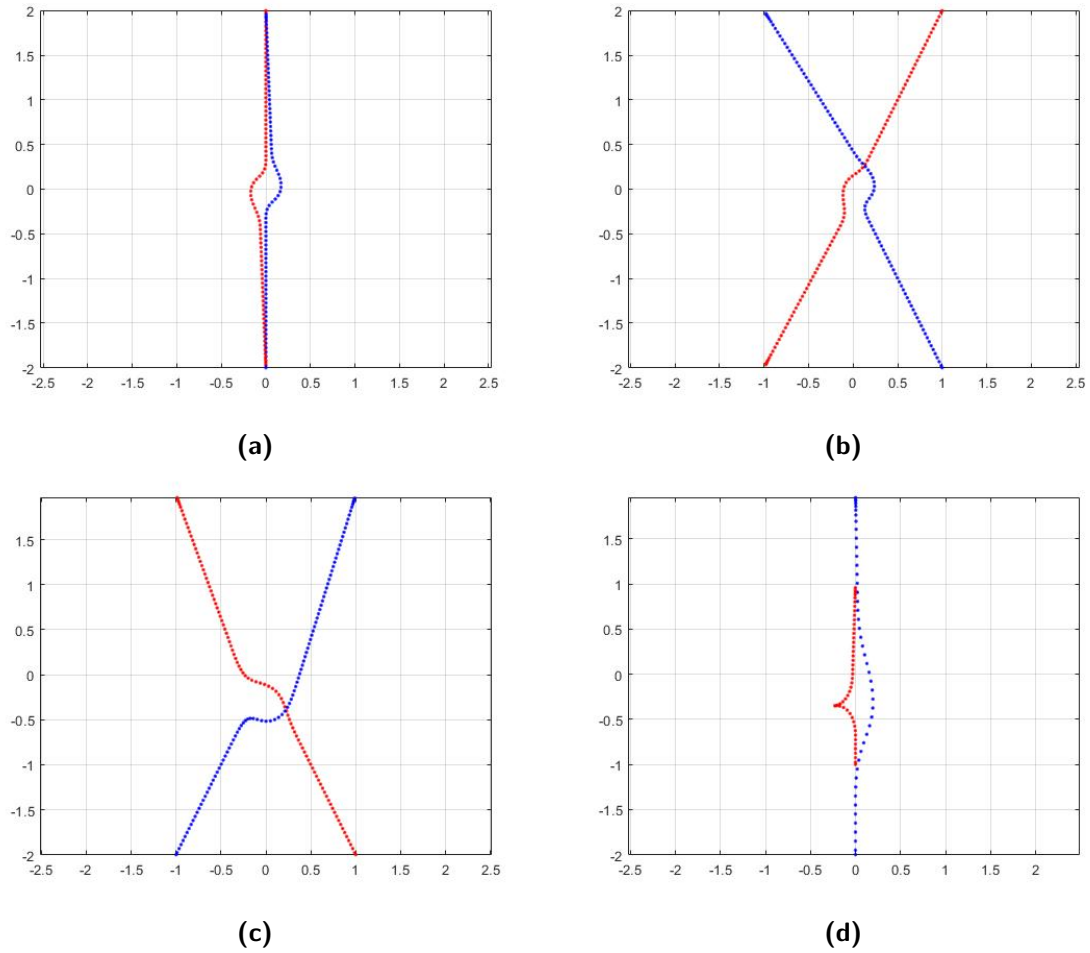
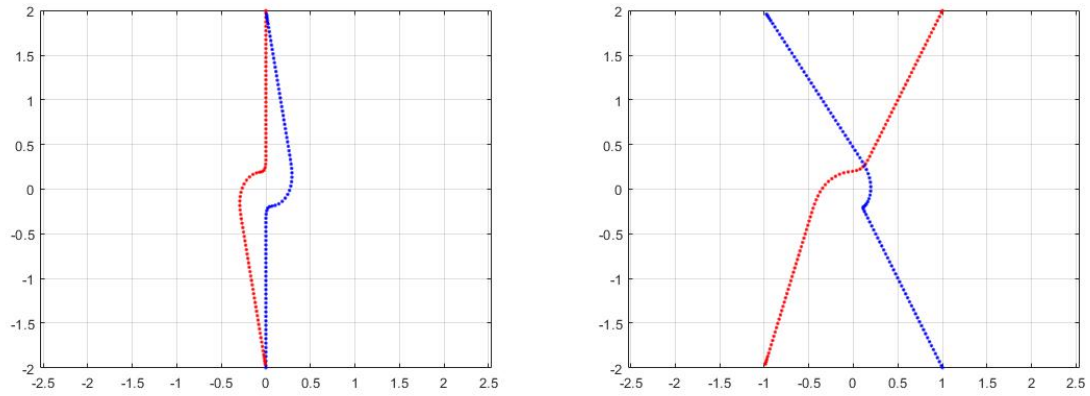


Figure 6-1: Two-agent Scenarios for PFs Based Method with $K_t = 1/2$ and $K_r = 0$

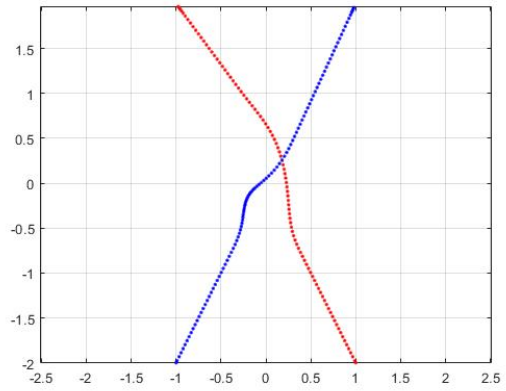
Parameter	Description
$R_0 = 0.15m$	Radius of agent
$dt = 0.05s$	Simulation sample time
$\rho_0 = 0.8m$	Detection zone radius
$R_c = 0.5m$	Safe distance
$r_d = 0.25m$	Slow down distance
K_g	Factor for navigation
K_r	Factor for avoidance in radius direction
K_t	Factor for avoidance in tangent direction

Table 6-1: Parameters in PFs based Collision Avoidance Maneuver

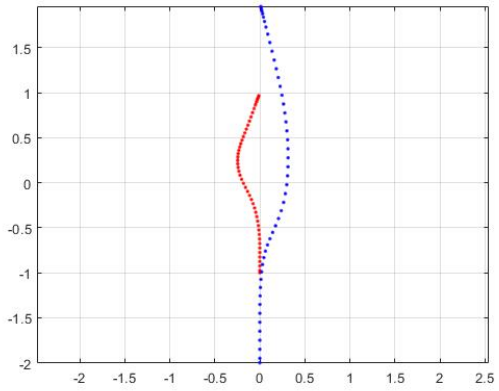


(a)

(b)



(c)



(d)

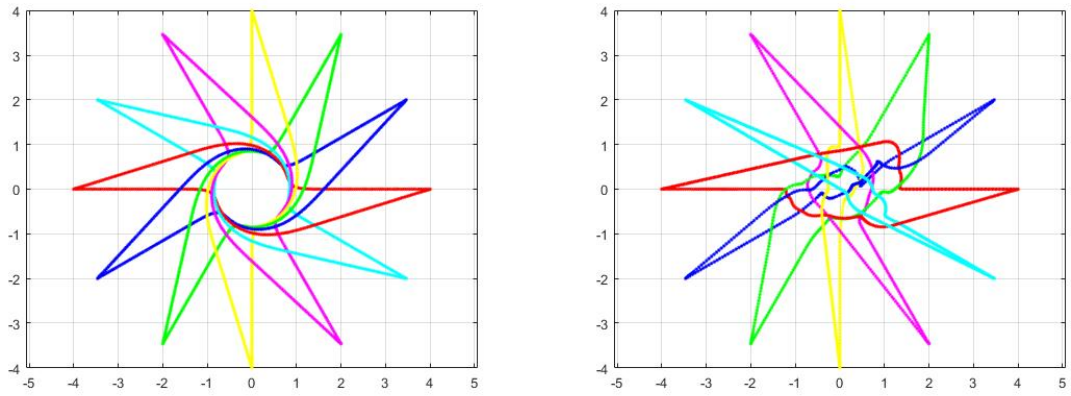
Figure 6-2: Two-agent Scenarios for PFs Based Method with $K_t = 1/6$ and $K_r = 1/3$

that as $K_g = 1$, the new velocity could be of the maximum speed at the primary step when the obstacle touches the detection zone. Choosing a smaller K_g but with the same ratio could decrease the possibility of collision. Also, this is only a sufficient condition, according to our test, the collision would happen when $K_t = 3$, as we set a safe distance R_c of $0.5m$ so that the agent would move away , there are some tolerance over there. The difference is the smallest distance between two agent or the proportion of collision as we will discuss in following section 4-2. In Figure 6-2, two agent performance with both tangential component and radius component is given.

6-1-2 Multi-agent Scenarios in 2D

There we present the result of several multi-agent scenarios.

1. **Circle**: Twelve agents crossing each other to antipodal points in a circle with radius of 4m. All agents' speed limit are set as 1m/s. The result is given in Figure 6-3a. It is also works when speed limit is chosen randomly between 0.5m/s and 1.5m/s. The speed limit used in random case shown in Figure 6-3b are [0.6576 1.4706 1.4572 0.9854 1.3003 0.6419 0.9218 1.4157 1.2922 1.4595 1.1557 0.5357] (m/s). Both Figure is based on potential field method with $K_g = 1$, $K_t = 1/6$ and $K_r = 1/3$.



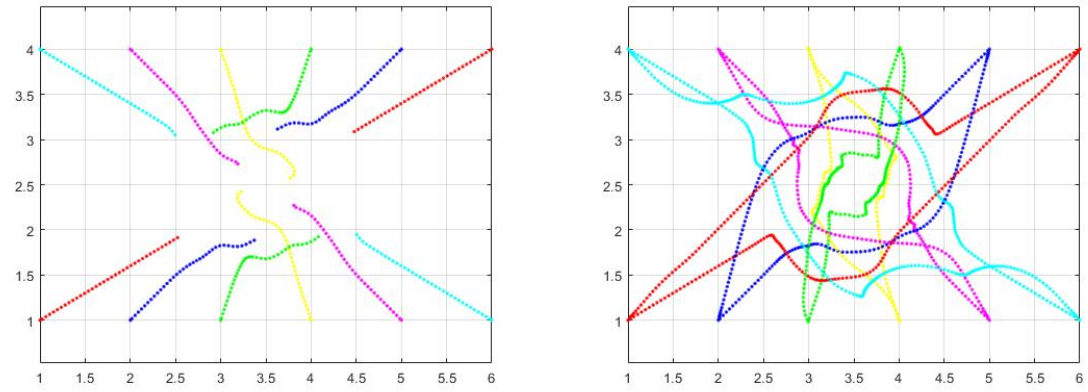
(a) Identical Speed Limit Circle **(b)** Random Speed Limit Circle

Figure 6-3: Circle Scenarios for PFs Based Method

2. **Line-Cross**: There are twelve agents in the work space, six of them line in the y=1 with x=1:6 and other six of them line in the y=4 with x=6:-1:1 at the very beginning. The target position is central symmetric of the point (3.5,2.5).

$$start = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

$$goal = \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 & 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 4 & 4 & 4 & 4 & 4 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



(a) Line-cross with $K_g = 1$, $K_t = 1/2$, $K_r = 0$ **(b)** Line-cross with $K_g = 1$, $K_t = 1/6$, $K_r = 1/3$

Figure 6-4: Line-cross Scenarios for PFs Based Method

The trajectory for potential field based method with $K_g = 1$, $K_t = 1/2$, $K_r = 0$ as well as $K_g = 1$, $K_t = 1/6$ and $K_r = 1/3$ are shown in Figure 6-4. As the failure of Figure 6-4a shown, tangent avoidance component only is no longer feasible for dense and complex simulation scenarios.

6-1-3 Scenarios in 3D

If not mentioned specifically, all the simulation given in this section is based on potential field method with $K_g = 1$, $K_t = 1/6$, $K_r = 1/3$. All other parameters is set as shown in Table 6-1 and the maximum speed of all agents is chosen as 1m/s.

- **Pyramid**: Four agents placed in the vertices of a regular triangular pyramid are supposed to reach their antipodal. While their center be the origin of coordinate, the initial positions are $[\sqrt{3}, 1, -\sqrt{3}/2]$, $[-\sqrt{3}, 1, -\sqrt{3}/2]$, $[0, -2, -\sqrt{3}/2]$, $[0, 0, -3\sqrt{3}/2]$ respectively. The result trajectory is shown in Figure 6-5.

  All agents go directly towards the target position until they are gathered in the middle of the pyramid, then collision avoidance maneuver becomes activated. All agents are "frozen" in z-direction, agent 1-3 follow the behaviour of 2D without the central agent 4 as shown in trajectory projected in x-y plane (Figure 6-5(b)). Then they continue moving in z-direction as the hazard of collision is resolved in horizontal plane. The relative distance of Agent 1 to obstacles is shown in Figure 6-6. The plot for Agent 2 is coincident with that of Agent 3. In fact, due to the symmetry of Agent 1-3, the view from the perspective of Agent 1-3 is the same. As the minimum relative distance over time is always larger than the safe distance 0.5m shown in yellow line, collision-free path is validated.

- **Cube**: Here we have eight agents starting at the vertices of a cube with side length of 4m and the goals are symmetric to start positions with the origin as symmetric center. The trajectories for the method combine tangent vector obtained by Gram-

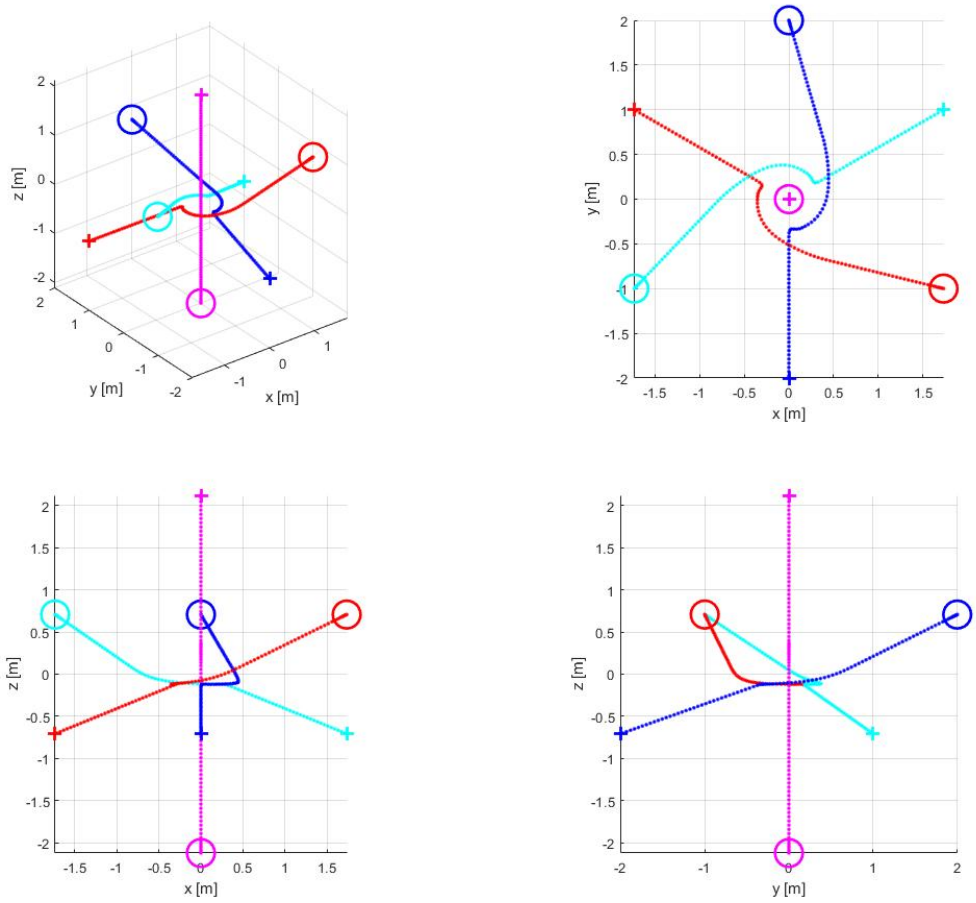


Figure 6-5: Trajectories of Pyramid Scenario for PFs Based Method with Horizontal Tangent Vector in 3D view (a), x-y plane (b), x-z plane (c) and y-z plane (d).

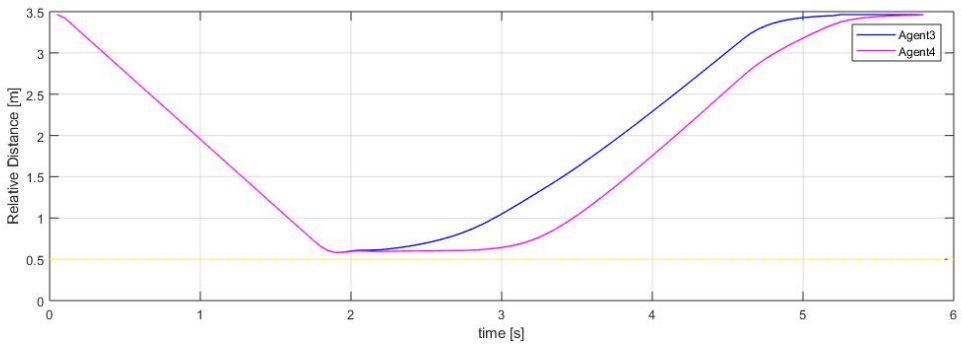
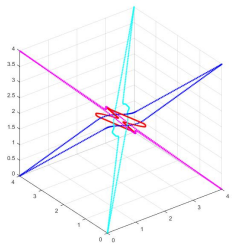
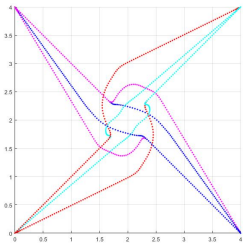


Figure 6-6: Relative Distance over time of Agent 1 in Pyramid Scenario for PF Based Method with Horizontal Tangent Vector

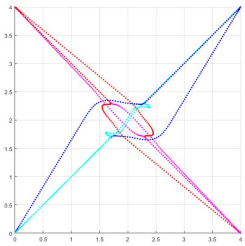
Scmidt orthonormalization and horizontal plane are shown in Figure 6-7. The trajectory for using horizontal tangent vector only is given in Figure 6-8



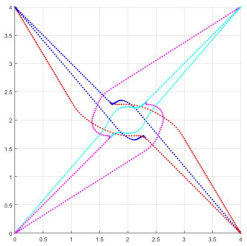
(a) Cube Scenario Trajectory in General



(b) Cube Scenario Trajectory in x-y Plane



(c) Cube Scenario Trajectory in x-z Plane



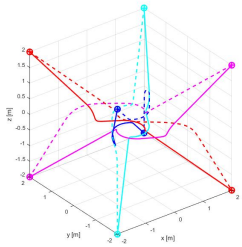
(d) Cube Scenario Trajectory in y-z Plane

Figure 6-7: Cube Scenarios for PFs Based Method with Combined Tangent Vector

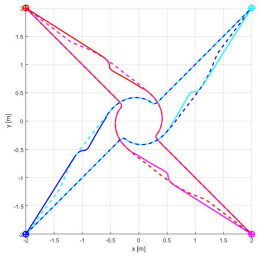
Dashed lines represent for the agents starting at the top and solid lines represent for the agents start from the bottom of the cube. As shown in the cube case above, the total travelled distance for top-agent is 8.7979m and for bottom-agent is 8.8823m. Bottom-agent spends 8 more steps to reach goal position when top-agent spend 177 steps. Total acceleration is $134.7973 m/s^2$ and $127.1833 m/s^2$ respectively. As shown in 6-9, all agent detected collision around time 3s, when Pz become steady, Px, Py becomes different for magenta and red from cyan and blue agents. Then the collision avoidance protocol is activated in horizontal plane where two groups of four agents response to the collision separately. The hazard in x-y plane is resolved around time 5s when the dashed lines and solid lines separate and the agents starts to move in z-direction. No obstacle is detected until a time after 6s when all agents' velocity in every direction becomes constancy.

- **Random Grid**: This scenario contains 12 agents start and reach a random integer point with a error of 0.1 of the $3 \times 3 \times 3$ cube. Due to the density of agents, the possibility of agents come close enough to detected each other is relative high, as shown in the minimum relative distance plot in Figure 6-12. While the agent are not likely to deal with several agents at the same time, collision hazard could be eliminated in an early stage.

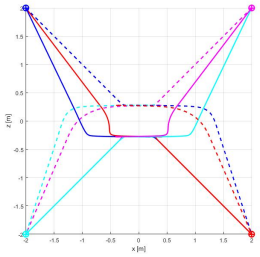
- **Random Points on Sphere Surface**: 12 agents starts at a random position on the surface of a sphere with radius 3m. A remarkable conflicted situation in the middle of the sphere is occurred as all agents will arrive at the center without having activated the collision protocol before. As the result shown in Figure 6-13, collision could be resolved successfully.



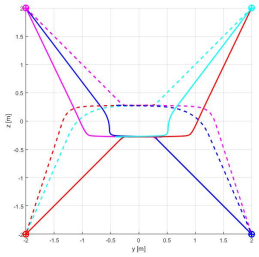
(a) Cube Scenario Trajectory in General



(b) Cube Scenario Trajectory in x-y Plane



(c) Cube Scenario Trajectory in x-z Plane



(d) Cube Scenario Trajectory in y-z Plane

Figure 6-8: Cube Scenarios for PFs Based Method with Horizontal Tangent Vector

Propriety in 3D:

- For PF Based Method with a horizontal tangent vector, the trajectory in x-y plane is the same as 2D case if there are no conflict in z-direction in the whole procedure.
- Radius avoidance component part must be included ($K_r > 0$) in order to make sure agent could 'stop' in z-direction while the tangent avoidance component is responsible for resolving collision in horizontal plane.
- The effort for collision avoidance is different for agent starting from bottom and starting from top even for a symmetric cube scenario as Figure 6-8. The symmetric behavior in x-y plane, however, maintains as in 2D cases.

6-2 Discussion

- Although some of the methods are originally used for static obstacle, it is also possible to adapt it to the dynamic (multi-agent) case. As the maximum speed of agent is a fixed number, we could use the prior information of the maximum max speed of all the agent in the workspace and extend the radius of moving obstacle with a shell of its maximum speed multiplied by some time window τ. If we can make sure for every iteration the selected agent would not touch the border of extended shell of obstacle before time τ, then we could say the overall path for agent would be collision-free. (not necessary but sufficient condition)
- We start with a complex multi-agent high dimensional collision avoidance problem, then transfer it to a more simple two-agent 2-D case for which we have collision avoidance

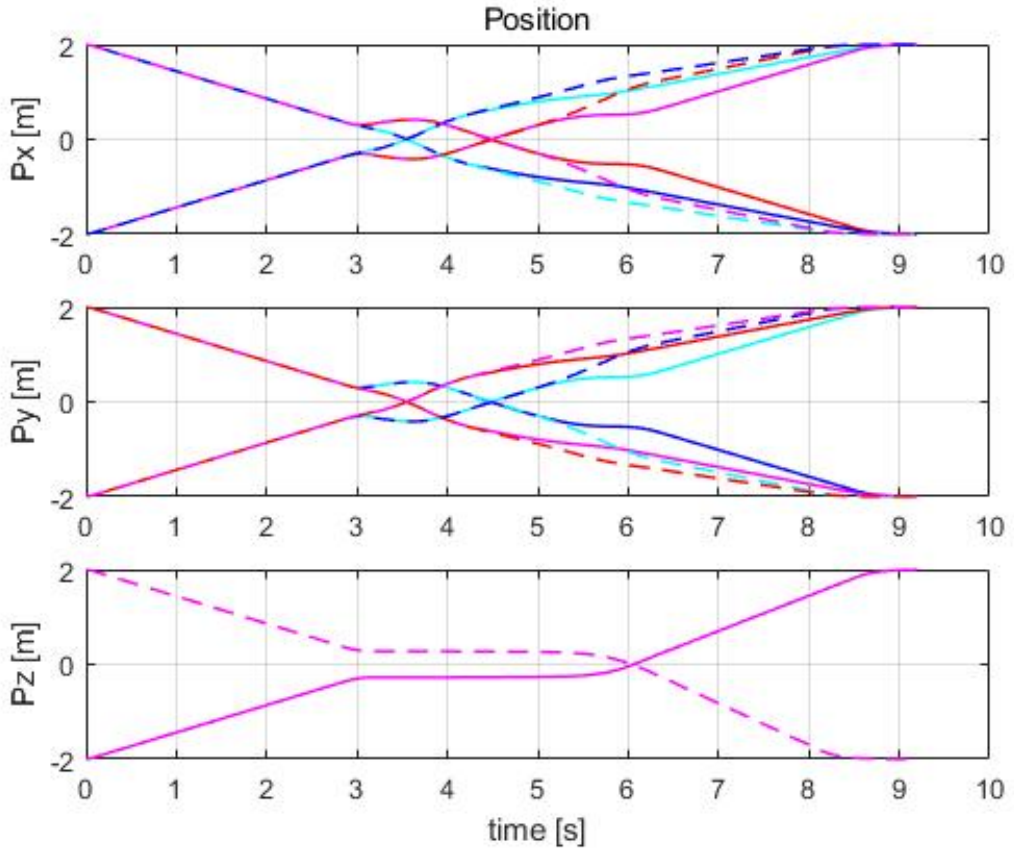


Figure 6-9: Position change over time of Cube Scenarios for PF Based Method with Horizontal Tangent Vector

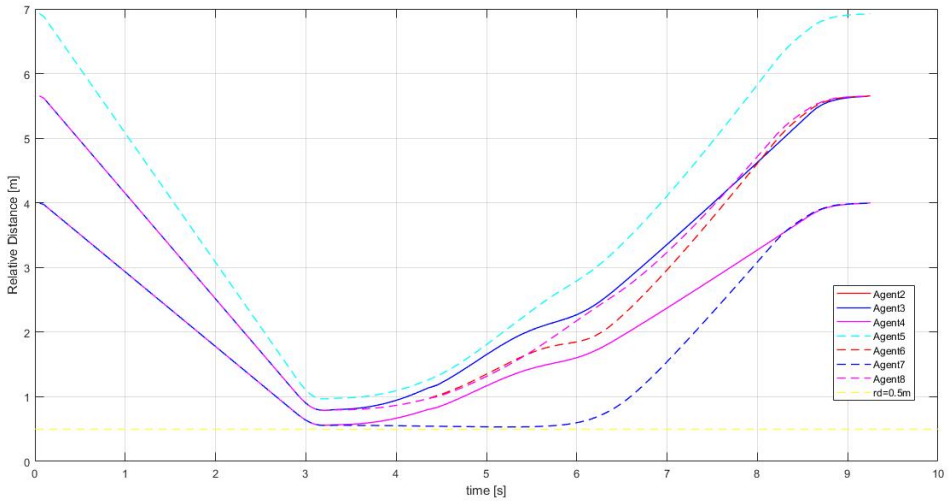


Figure 6-10: Relative Distance change among time of Agent 1 in Cube Scenarios for PF Based Method with Horizontal Tangent Vector

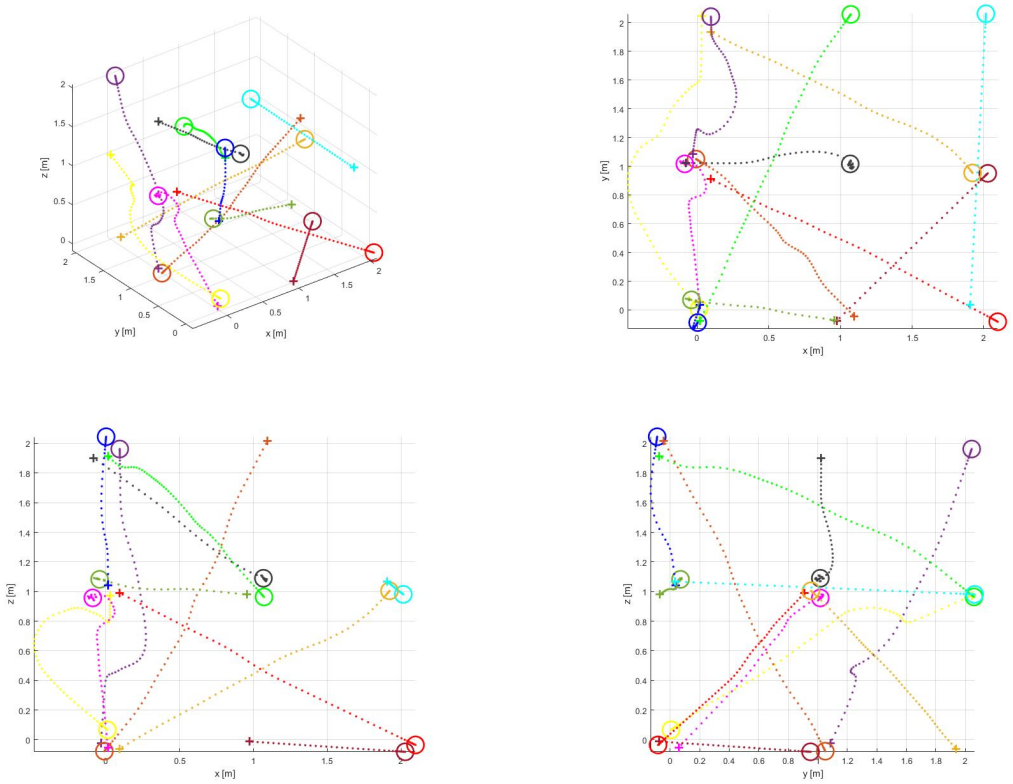


Figure 6-11: Trajectories of Random Grid Scenario for PFs Based Method with Horizontal Tangent Vector in 3D view (a), x-y plane (b), x-z plane (c) and y-z plane (d).

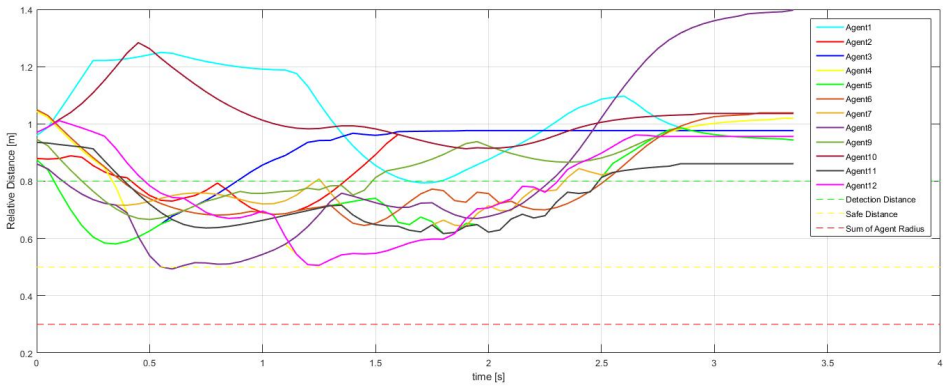


Figure 6-12: Minimum Relative Distance over Time in Random Grid Scenarios for PF Based Method with Horizontal Tangent Vector

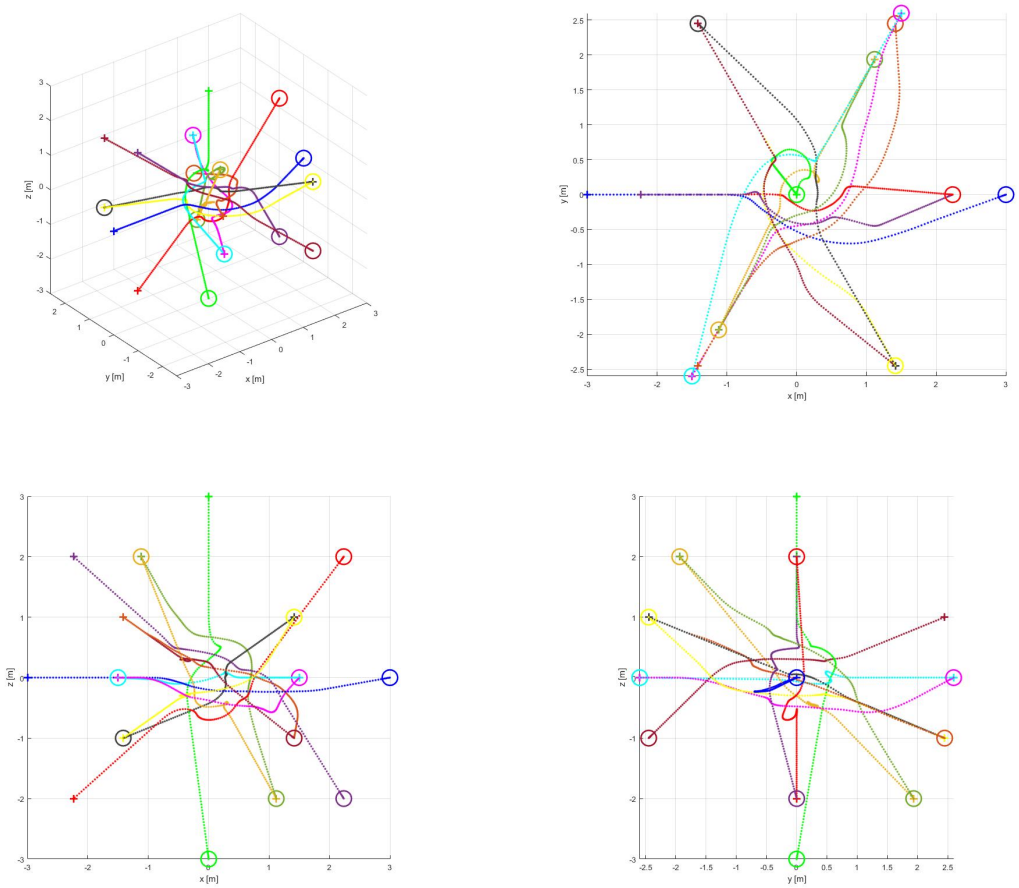


Figure 6-13: Trajectories of Sphere Scenario for PFs Based Method with Horizontal Tangent Vector in 3D view (a), x-y plane (b), x-z plane (c) and y-z plane (d).

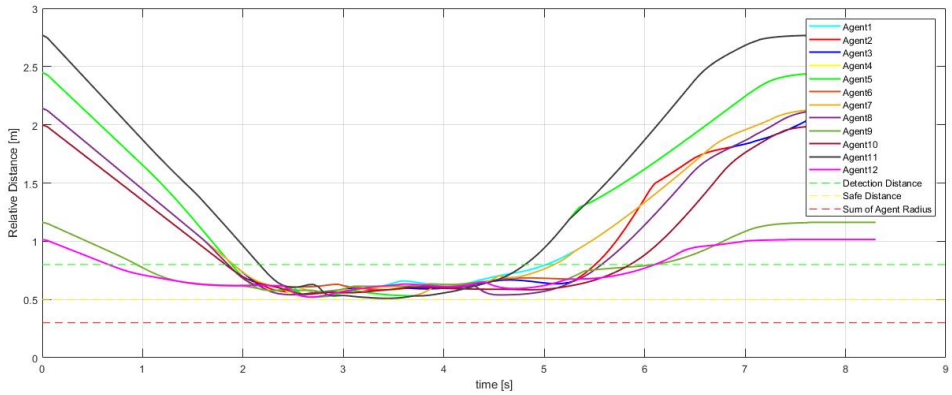


Figure 6-14: Minimum Relative Distance over Time in Sphere Scenarios for PF Based Method with Horizontal Tangent Vector

guarantee. During this process, we might eliminate some optimal solution that leads to a smaller total travelled distance or less acceleration, but the most important thing is that the task could be accomplished, reach the target without collision.

- The deadlock situations (zero-velocity) are marginal and unlikely when tangent avoidance component is introduced. It is definitely worth stressing that, no deadlock or livelock has been spotted for the scenarios we considered so far which is not extremely crowded and confined. Of course, in principle livelocks or deadlocks may occur. But still, this kind of classical failing for local method could be solved by a higher-level motion planner that gets agent out of their deadlocks/livelocks.

6-3 Summary

In this chapter, extensive simulations have been performed, for many different scenarios both in 2D and in 3D. Utilizing the method proposed in Chapter 5 has always enabled the agents to reach their targets, also in quite crowded environments with tight boundaries and obstacles, without any collision and without experiencing livelocks or deadlocks.

Chapter 7

Conclusion and Recommendations

7-1 Conclusion

In this thesis, we presented two kinds of collision avoidance methods: a rotation-based method (method 1) and a potential-field-based method (method 2). With proper choice of parameters, collision-free paths could be generated for two agent scenarios in 2-dimensional for both methods. Although some livelock is spotted for method 1 which is based on rotation in 3-dimensional space, the other method based on potential fields works smoothly without livelock in 3-dimensional space even in a crowded multi-agent and significant conflict situation. It is worth mentioning that all the methods discussed in this thesis are guaranteed to work and provide collision-free paths only in the case of single encounters; while method 2 appears to work also in more crowded environments and in the case of multiple obstacles, method 1 appears to have more problems dealing with multi-obstacle simultaneously. The thesis objective is to give some mathematical representation and simulation of methods that are based on human intuition to react to a potential collision. No collision is visually observed during our simulation campaigns described in Chapter 6. The analytical reasoning to guarantee collision-free paths for multi-agent case is not rigorously proved for now. Most of the analysis treats dynamic obstacles as if they were static ones, which means that there is no literally cooperation among the agents (who see other agents only as obstacles). To guarantee a collision-free trajectory, some extra safe distance is required so that the trajectory could be preserved and detour spotted all the time.

7-2 Recommendations

As the presented methods are only implemented in ideal simulation environment, it would be a huge progress if we could apply them in real world hardware. In this case, we need to drop all our simplifying assumptions and take the uncertainty in radius, position and velocity, noise of sensor into account.

For now, the single agent model is holonomic with whole sphere field of view in 3D and no force input constrains (velocity can change to the desired value almost instantly). It is suggested to take the dynamic constraints of acceleration, maximum torque, etc and field of view limitation into consideration. Extending the methods to deal with agents with kinematic constraints such as differential-drive, car-like, etc non-holonomic model that introduced heading information would also be important.

The future work on collision avoidance could be processed in the context of formations, namely considering groups of agents that are constrained to move in a particular pattern, where the relative positioning between agents is quite important.

We could also allow agents to use their destination information as well as some optimization process to improve the performance. With the internal goal information of each agent shared through local communication, it is possible to address the problem of decentralized goal assignment and trajectory generation for multi-agent network.

Bibliography

[1] D. van Wijk, "Network-decentralized control with collision avoidance for multi-agent systems," 2018.

[2] S. A. Masoud and A. A. Masoud, "Motion planning in the presence of directional and regional avoidance constraints using nonlinear, anisotropic, harmonic potential fields: a physical metaphor," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 32, no. 6, pp. 705–723, 2002.

[3] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 1928–1935, IEEE, 2008.

[4] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, 2011.

[5] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*, pp. 3–19, Springer, 2011.

[6] A. Barrat, M. Barthelemy, and A. Vespignani, *Dynamical processes on complex networks*. Cambridge university press, 2008.

[7] W. Ren and Y. Cao, *Distributed coordination of multi-agent networks: emergent problems, models, and issues*. Springer Science & Business Media, 2010.

[8] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*, vol. 33. Princeton University Press, 2010.

[9] X. G. Yan, J. Lam, H. S. Li, and I. M. Chen, "Decentralized control of nonlinear large-scale systems using dynamic output feedback," *Journal of Optimization Theory and Applications*, vol. 104, pp. 459–475, Feb 2000.

[10] A. J. Lotka, "Undamped oscillations derived from the law of mass action," *Journal of American Chemical Society*, vol. 42, pp. 1595–1599, 1920.

[11] M. Feinberg, "Chemical reaction network structure and the stability of complex isothermal reactors: I. the deficiency zero and deficiency one theorems," *Chemical Engineering Science*, vol. 42, no. 10, pp. 2229–2268, 1987.

[12] M. Feinberg, "Chemical reaction network structure and the stability of complex isothermal reactors: II. multiple steady states for networks of deficiency one," *Chemical Engineering Science*, vol. 43, no. 1, pp. 1–25, 1988.

[13] M. Feinberg, "The existence and uniqueness of steady states for a class of chemical reaction networks," *Archive for Rational Mechanics and Analysis*, vol. 132, no. 4, pp. 311–370, 1995.

[14] M. Feinberg, "Multiple steady states for chemical reaction networks of deficiency one," *Archive for Rational Mechanics and Analysis*, vol. 132, no. 4, pp. 371–406, 1995.

[15] G. Craciun and M. Feinberg, "Multiple equilibria in complex chemical reaction networks: I. the injectivity property," *SIAM Journal on Applied Mathematics*, vol. 65, no. 5, pp. 1526–1546, 2005.

[16] U. Alon, *An introduction to systems biology: design principles of biological circuits*. Chapman and Hall/CRC, 2006.

[17] M. Chaves, "Stability of rate-controlled zero-deficiency networks," in *45th IEEE Conference on Decision and Control*, pp. 5766–5771, 2006.

[18] G. Craciun and M. Feinberg, "Multiple equilibria in complex chemical reaction networks: II. the species-reaction graph," *SIAM Journal on Applied Mathematics*, vol. 66, no. 4, pp. 1321–1338, 2006.

[19] M. Mincheva and G. Craciun, "Multigraph conditions for multistability, oscillations and pattern formation in biochemical reaction networks," *Proceedings of the IEEE*, vol. 96, no. 8, pp. 1281–1291, 2008.

[20] M. Domijan and M. Kirkilionis, "Bistability and oscillations in chemical reaction networks," *Journal of mathematical biology*, vol. 59, no. 4, pp. 467–501, 2009.

[21] L. Chen, R. Wang, C. Li, and K. Aihara, *Modeling Biomolecular Networks in Cells*. Springer, 2005.

[22] G. Iacono and C. Altafini, "Monotonicity, frustration, and ordered response: an analysis of the energy landscape of perturbed large-scale biological networks," *BMC Systems Biology*, vol. 108, p. 83, 2010.

[23] K. M. Hangos, "Engineering model reduction and entropy-based Lyapunov functions in chemical reaction kinetics," *Entropy*, vol. 12, no. 4, pp. 772–797, 2010.

[24] M. Domijan and E. Pécou, "The interaction graph structure of mass-action reaction networks," *Journal of Mathematical Biology*, vol. 65, no. 2, pp. 375–402, 2012.

[25] C. Cosentino and D. Bates, *Feedback control in systems biology*. Crc Press, 2011.

[26] D. Del Vecchio and R. M. Murray, *Biomolecular feedback systems*. Princeton University Press Princeton, NJ, 2015.

[27] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, p. 651, 2000.

[28] A. Vinayagam, T. E. Gibson, H.-J. Lee, B. Yilmazel, C. Roesel, Y. Hu, Y. Kwon, A. Sharma, Y.-Y. Liu, N. Perrimon, *et al.*, "Controllability analysis of the directed human protein interaction network identifies disease genes and drug targets," *Proceedings of the National Academy of Sciences*, vol. 113, no. 18, pp. 4976–4981, 2016.

[29] V. Volterra, "Fluctuations in the abundance of a species considered mathematically," *Nature*, vol. 118, pp. 558–560, 1926.

[30] R. Levins, *Evolution in changing environments: some theoretical explorations*. Princeton University Press, 1968.

[31] R. Levins, "The qualitative analysis of partially specified systems," *Annals of the New York Academy of Science*, vol. 231, pp. 123–138, 1974.

[32] R. Levins, "Evolution in communities near equilibrium," in *Ecology and evolution of communities* (M. Cody and J. M. Diamond, eds.), pp. 16–50, Harvard University Press, 1975.

[33] R. M. May, *Stability and Complexity in Model Ecosystems, 2nd ed.* Princeton University Press, 1974.

[34] C. J. Puccia and R. Levins, *Qualitative Modeling of Complex Systems*. Harvard University Press, Cambridge, MA, 1985.

[35] M. P. Marzloff, J. M. Dambacher, C. R. Johnson, L. R. Little, and S. D. Frusher, "Exploring alternative states in ecological systems with a qualitative analysis of community feedback," *Ecological Modelling*, vol. 222, no. 15, pp. 2651–2662, 2011.

[36] N. E. Friedkin and E. C. Johnsen, "Social influence networks and opinion change," in *Advances in Group Processes* (E. J. Lawler and M. W. Macy, eds.), vol. 16, pp. 1–29, JAI Press, 1999.

[37] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar, "Opinion fluctuations and disagreement in social networks," *Mathematics of Operations Research*, vol. 38, no. 1, pp. 1–27, 2013.

[38] N. E. Friedkin, "The problem of social control and coordination of complex systems in sociology," *IEEE Control Systems Magazine*, pp. 40–51, June 2015.

[39] C. Altafini and G. Lini, "Predictable dynamics of opinion forming for networks with antagonistic interactions," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 342–357, 2015.

[40] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[41] D. Bauso, L. Giarré, and R. Pesenti, "Non-linear protocols for optimal distributed consensus in networks of dynamic agents," *Systems & Control Letters*, vol. 55, no. 11, pp. 918–928, 2006.

[42] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[43] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control: collective group behavior through local interaction," *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, 2007.

[44] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *Automatica*, vol. 44, no. 3, pp. 671–684, 2008.

[45] Y. G. Sun, L. Wang, and G. Xie, "Average consensus in networks of dynamic agents with switching topologies and multiple time-varying delays," *Systems & Control Letters*, vol. 57, no. 2, pp. 175–183, 2008.

[46] R. Carli and S. Zampieri, "Network clock synchronization based on the second-order linear consensus algorithm," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 409–422, 2014.

[47] A. Atamtürk and M. Zhang, "Two-stage robust network flow and design under demand uncertainty," *Operations Research*, vol. 55, no. 4, pp. 662–673, 2007.

[48] F. Ordóñez and J. Zhao, "Robust capacity expansion of network flows," *Networks: An International Journal*, vol. 50, no. 2, pp. 136–145, 2007.

[49] D. Bauso, F. Blanchini, and R. Pesenti, "Optimization of long-run average-flow cost in networks with time-varying unknown demand," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 20–31, 2010.

[50] J. Wei and A. J. van der Schaft, "Stability of dynamical distribution networks with arbitrary flow constraints and unknown in/outflows," in *Proceedings of the IEEE Conference on Decision and Control*, (Firenze, Italy), pp. 55–60, 2013.

[51] J. Wei and A. J. van der Schaft, "Load balancing of dynamical distribution networks with flow constraints and unknown in/outflows," *Systems & Control Letters*, vol. 62, no. 11, pp. 1001–1008, 2013.

[52] C. Danielson, F. Borrelli, D. Oliver, D. Anderson, and T. Phillips, "Constrained flow control in storage networks: Capacity maximization and balancing," *Automatica*, vol. 49, no. 9, pp. 2612–2621, 2013.

[53] M. Bürger and C. D. Persis, "Dynamic coupling design for nonlinear output agreement and time-varying flow control," *Automatica*, vol. 51, no. 1, pp. 210–222, 2015.

[54] R. E. Larson and W. G. Keckler, "Applications of dynamic programming to the control of water resource systems," *Automatica*, vol. 5, no. 1, pp. 15–26, 1969.

[55] D. Bauso, F. Blanchini, L. Giarré, and R. Pesenti, "The linear saturated decentralized strategy for constrained flow control is asymptotically optimal," *Automatica*, vol. 49, no. 7, pp. 2206–2212, 2013.

[56] E. A. Silver and R. Peterson, *Decision System for Inventory Management and Production Planning*. Wiley, New York, NY, 1985.

[57] E. K. Boukas, H. Yang, and Q. Zhang, "Minimax production planning in failure-prone manufacturing systems," *Journal of Optimization Theory and Applications*, vol. 87, no. 2, pp. 269–286, 1995.

[58] F. Blanchini, F. Rinaldi, and W. Ukovich, "Least inventory control of multi-storage systems with non-stochastic unknown input," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 5, pp. 633–645, 1997.

[59] F. Blanchini, S. Miani, and W. Ukovich, "Control of production-distribution systems with unknown inputs and system failures," *IEEE Transactions on Automatic Control*, vol. 45, no. 6, pp. 1072–1081, 2000.

[60] F. Blanchini, S. Miani, and F. Rinaldi, "Guaranteed cost control for multi-inventory systems with uncertain demand," vol. 40, no. 2, pp. 213–224, 2004.

[61] D. Bertsimas and A. Thiele, "A robust optimization approach to inventory theory," *Operations Research*, vol. 54, no. 1, pp. 150–168, 2006.

[62] H. Sarimveisa, P. Patrinos, C. D. Tarantilis, and C. T. Kiranoudis, "Dynamic modeling and control of supply chain systems: A review," *Computers & Operations Research*, vol. 35, pp. 3530–3561, 2008.

[63] M. Bürger, C. De Persis, and F. Allgöwer, "Dynamic pricing control for constrained distribution networks with storage," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 1, pp. 88–97, 2015.

[64] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," in *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pp. 1–8, IEEE, 2009.

[65] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power and Energy Magazine*, vol. 7, no. 2, pp. 52–62, 2009.

[66] H. Farhangi, "The path of the smart grid," *IEEE power and energy magazine*, vol. 8, no. 1, 2010.

[67] C.-H. Lo and N. Ansari, "Decentralized controls and communications for autonomous distribution networks in smart grid," *IEEE transactions on smart grid*, vol. 4, no. 1, pp. 66–77, 2013.

[68] B. Ataşlar and A. İftar, "A decentralized control approach for transportation networks," *IFAC Proceedings Volumes*, vol. 31, no. 20, pp. 343–348, 1998.

[69] S. Mudchanatongsuk, F. Ordóñez, and J. Liu, "Robust solutions for network design under transportation cost and demand uncertainty," *Journal of the Operational Research Society*, vol. 59, no. 5, pp. 652–662, 2008.

[70] A. Iftar and E. J. Davison, "Decentralized robust control for dynamic routing of large scale networks," in *American Control Conference, 1990*, pp. 441–446, IEEE, 1990.

[71] R. D'Andrea, "A linear matrix inequality approach to decentralized control of distributed parameter systems," in *American Control Conference, 1998. Proceedings of the 1998*, vol. 3, pp. 1350–1354, IEEE, 1998.

[72] A. İftar, "A linear programming based decentralized routing controller for congested highways," *Automatica*, vol. 35, no. 2, pp. 279–292, 1999.

[73] A. İftar and E. J. Davison, "Decentralized control strategies for dynamic routing," *Optimal Control Applications and Methods*, vol. 23, no. 6, pp. 329–355, 2002.

[74] F. H. Moss and A. Segall, "An optimal control approach to dynamic routing in networks," *IEEE Transactions on Automatic Control*, vol. 27, no. 2, pp. 329–339, 1982.

[75] A. Ephremides and S. Verdú, "Control and optimisation methods in communication networks," *IEEE Transactions on Automatic Control*, vol. 34, no. 9, pp. 930–942, 1989.

[76] A. İftar and E. Davison, "A decentralized control strategy for dynamic routing," in *Decision and Control, 1989., Proceedings of the 28th IEEE Conference on*, pp. 828–834, IEEE, 1989.

[77] A. İftar and E. Davison, "A decentralized control strategy with discrete-time updating for dynamic routing," *IFAC Proceedings Volumes*, vol. 23, no. 8, pp. 111–116, 1990.

[78] F. Blanchini, G. Giordano, and P. L. Montessoro, "Network-decentralized robust congestion control with node traffic splitting," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 2901–2906, IEEE, 2014.

[79] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2013.

[80] S.-H. Wang and E. Davison, "On the stabilization of decentralized control systems," *IEEE Transactions on Automatic Control*, vol. 18, no. 5, pp. 473–478, 1973.

[81] L. Bakule, "Decentralized control: An overview," *Annual reviews in control*, vol. 32, no. 1, pp. 87–98, 2008.

[82] D. D. Siljak, *Decentralized control of complex systems*. Courier Corporation, 2011.

[83] B. Labibi, H. J. Marquez, and T. Chen, "Lmi optimization approach to robust decentralized controller design," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 8, pp. 904–924, 2011.

[84] B. Bollobas, *Modern Graph Theory*. New York: Springer-Verlag, 1998.

[85] F. Blanchini, E. Franco, and G. Giordano, "Structured-lmi conditions for stabilizing network-decentralized control," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 6880–6885, IEEE, 2013.

[86] F. Blanchini, E. Franco, and G. Giordano, "Network-decentralized control strategies for stabilization," *IEEE Transactions on Automatic Control*, vol. 60, no. 2, pp. 491–496, 2015.

[87] F. Blanchini, E. Franco, G. Giordano, V. Mardanlou, and P. L. Montessoro, "Compartmental flow control: Decentralization, robustness and optimality," *Automatica*, vol. 64, pp. 18–28, 2016.

[88] V. Ugrinovskii, "Distributed robust estimation over randomly switching networks using H∞ consensus," *Automatica*, vol. 49, no. 1, pp. 160–168, 2013.

[89] H. Liu and T. Zhou, "Distributed observer design for networked dynamical systems," in *Control and Decision Conference (CCDC), 2015 27th Chinese*, pp. 3791–3796, IEEE, 2015.

[90] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic, "Consensus based overlapping decentralized estimator," in *American Control Conference, 2007. ACC'07*, pp. 2744–2749, IEEE, 2007.

[91] J. Cortés, "Global and robust formation-shape stabilization of relative sensing networks," *Automatica*, vol. 45, no. 12, pp. 2754–2762, 2009.

[92] T. Ishizaki, Y. Sakai, K. Kashima, and J.-i. Imura, "Hierarchical decentralized observer design for linearly coupled network systems," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pp. 7831–7836, IEEE, 2011.

[93] M. Franceschelli and A. Gasparri, "Gossip-based centroid and common reference frame estimation in multiagent systems.," *IEEE Trans. Robotics*, vol. 30, no. 2, pp. 524–531, 2014.

[94] G. Giordano, F. Blanchini, E. Franco, V. Mardanlou, and P. L. Montessoro, "The smallest eigenvalue of the generalized laplacian matrix, with application to network-decentralized estimation for homogeneous systems," *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 4, pp. 312–324, 2016.

[95] J. Wolfe, D. Chichka, and J. Speyer, "Decentralized controllers for unmanned aerial vehicle formation flight," *AIAA*, vol. 96–3833, 1996.

[96] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE transactions on automatic control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[97] D. M. Stipanović, G. Inalhan, R. Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.

[98] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33, 2008.

[99] F. Liu and A. Narayanan, "A human-inspired collision avoidance method for multi-robot and mobile autonomous robots," in *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 181–196, Springer, 2013.

[100] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[101] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[102] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*, pp. 396–404, Springer, 1986.

[103] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "Smooth and collision-free navigation for multiple robots under differential-drive constraints," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4584–4589, IEEE, 2010.

[104] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car-like robots," in *2012 IEEE International Conference on Robotics and Automation*, pp. 360–366, IEEE, 2012.

[105] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems*, pp. 203–216, Springer, 2013.

[106] M. Choi, A. Rubenecia, T. Shon, and H. Choi, "Velocity obstacle based 3d collision avoidance scheme for low-cost micro uavs," *Sustainability*, vol. 9, no. 7, p. 1174, 2017.

[107] J. Snape and D. Manocha, "Navigating multiple simple-airplanes in 3d workspace," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 3974–3980, IEEE, 2010.

[108] J. Alonso-Mora, T. Naegeli, R. Siegwart, and P. Beardsley, "Collision avoidance for aerial vehicles in multi-agent scenarios," *Autonomous Robots*, vol. 39, no. 1, pp. 101–121, 2015.

[109] S. Roelofsen, D. Gillet, and A. Martinoli, "Reciprocal collision avoidance for quadrotors using on-board visual detection," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4810–4817, IEEE, 2015.

[110] P. Conroy, D. Bareiss, M. Beall, and J. v. d. Berg, "3-d reciprocal collision avoidance on physical quadrotor helicopters with on-board sensing for relative positioning," *arXiv preprint arXiv:1411.3794*, 2014.

[111] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 1398–1404, IEEE, 1991.

[112] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on robotics and automation*, vol. 16, no. 5, pp. 615–620, 2000.

[113] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous robots*, vol. 13, no. 3, pp. 207–222, 2002.

[114] J.-O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.

[115] A. A. Masoud, "Decentralized self-organizing potential field-based control for individually motivated mobile agents in a cluttered environment: A vector-harmonic potential field approach," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 3, pp. 372–390, 2007.

[116] A. A. Masoud, "A harmonic potential approach for simultaneous planning and control of a generic uav platform," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 153–173, 2012.

[117] D. E. Chang, S. C. Shadden, J. E. Marsden, and R. Olfati-Saber, "Collision avoidance for multiple agent systems," 2003.

[118] S. A. Roelofsen, "Vision-based sense and avoid algorithms for unmanned aerial vehicles," tech. rep., EPFL, 2017.

[119] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[120] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 1, pp. 341–346, IEEE, 1999.

[121] E. Ferrera, J. Capitán, A. R. Castaño, and P. J. Marron, "Decentralized safe conflict resolution for multiple robots in dense scenarios," *Robotics and Autonomous Systems*, vol. 91, pp. 179–193, 2017.

[122] E. Ferrera, A. Alcántara, J. Capitán, A. Castaño, P. Marrón, and A. Ollero, "Decentralized 3d collision avoidance for multiple uavs in outdoor environments," *Sensors*, vol. 18, no. 12, p. 4101, 2018.

[123] C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo, "A novel 3d geometric algorithm for aircraft autonomous collision avoidance," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 1580–1585, IEEE, 2006.

[124] M. Schwager, D. Rus, and J.-J. Slotine, "Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 371–383, 2011.

[125] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[126] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 285–292, IEEE, 2017.

[127] J. LaSalle, "Some extensions of liapunov's second method," *IRE Transactions on circuit theory*, vol. 7, no. 4, pp. 520–527, 1960.

[128] D. E. Chang and J. E. Marsden, "Gyroscopic forces and collision avoidance with convex obstacles," in *New trends in nonlinear dynamics and control and their applications*, pp. 145–159, Springer, 2003.