Data-Driven Risk-sensitive Model Predictive Control for Safe Navigation in Multi-Robot Systems

Dual Degree Project - I (ME57003) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Master of Technology

in

Mechanical Engineering with specialization in Mechanical Systems Design

by

Atharva Navsalkar

(18ME33002)

Under the supervision of

Prof. Arun Kumar Samantaray
and
Prof. Ashish Ranjan Hota

Department of Mechanical Engineering

Indian Institute of Technology Kharagpur

Autumn Semester, 2022-23

November 08, 2022

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: November 08, 2022 (Atharva Navsalkar)

Place: Kharagpur (18ME33002)

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Data-Driven Risk-sensitive Model Predictive Control for Safe Navigation in Multi-Robot Systems**" submitted by **Atharva Navsalkar** (Roll No. 18ME33002) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Master of Technology in Mechanical Engineering with specialization in Mechanical Systems Design is a record of bona fide work carried out by him under my supervision and guidance during Autumn Semester, 2022-23.

Prof. Arun Kumar Samantaray
Department of Mechanical Engineering,
Indian Institute of Technology
Kharagpur
Kharagpur - 721302, India

Prof. Ashish Ranjan Hota
Department of Electrical Engineering,
Indian Institute of Technology
Kharagpur
Kharagpur - 721302, India

Abstract

Name of the student: **Atharva Navsalkar** Roll No: **18ME33002**

Degree for which submitted: **Master of Technology**

Department: **Department of Mechanical Engineering**

Thesis title: **Data-Driven Risk-sensitive Model Predictive Control for Safe Navigation in Multi-Robot Systems**

Thesis supervisors: **Prof. Arun K. Samantaray** and **Prof. Ashish R. Hota**

Month and year of thesis submission: **November 08, 2022**

Safe navigation is a fundamental challenge in multi-robot systems due to the uncertainty surrounding the future trajectory of the robots that act as obstacles for each other. In this work, a principled data-driven approach is proposed where each robot repeatedly solves a finite horizon optimization problem subject to collision avoidance constraints with latter being formulated as distributionally robust conditional value-at-risk (CVaR) of the distance between the agent and a polyhedral obstacle geometry. Specifically, the CVaR constraints are required to hold for all distributions that are close to the empirical distribution constructed from observed samples of prediction error collected during execution. The generality of the approach allows us to robustify against prediction errors that arise under commonly imposed assumptions in both distributed and decentralized settings. Tractable finite-dimensional approximations of this class of constraints are derived by leveraging convex and minmax duality results for Wasserstein distributionally robust optimization problems. The effectiveness of the proposed approach is illustrated in a multi-drone navigation setting implemented in Gazebo platform.

Acknowledgements

Firstly, I sincerely thank my thesis supervisors Prof. Ashish Hota, Department of Electrical Engineering, IIT Kharagpur and Prof. Arun Kumar Samantaray, Department of Mechanical Engineering for the guidance throughout the project duration and previously as well. I also thank AI & Robotics Technology Park, Indian Institute of Science for providing financial support for equipment purchase for this project through Innovation Grant Program. Lastly, I thank my friends and family for supporting me during the work of this thesis.

Contents

Chapter 1

Introduction

1.1 Overview

Multi-robot systems, including drones, ground robots, and autonomous vehicles, have seen tremendous development due to applications ranging from military, search and rescue missions, advanced mobility, cave explorations, indoor motion, warehouses and entertainment purposes. Motion planning of multiple robots in such a diverse set of environments remains an essential challenge. Each individual agent must safely avoid obstacles and other members of the group. The real world is, however, not completely deterministic in terms of both dynamics and state measurements. The primary goal of this work is to develop and implement computationally efficient distributed model predictive control (MPC) algorithms for risk-sensitive safe navigation of multiple robots under uncertainty.

MPC has been widely used for motion planning in robotic systems and drones and provides flexibility to modify the problem formulations for different settings. It is powerful tool that solves a finite horizon numerical optimization problem repeatedly to compute the control commands. The primary advantage of using MPC is its ability to accommodate various performance metrics and state/input constraints. Real world systems have an element of uncertainty and noise, and also surrounded by non-cooperative robots or obstacles. To handle the uncertain motion of obstacles and other agents, various types of constraints can be employed. Particularly, I capture

the risk of collision in terms of conditional value-at-risk (CVaR) of the distance between the robot and obstacle (or other agent) over the entire prediction horizon. Specifically, the proposed methods uses real-time onboard observed data to construct a family of probability distributions close to the observed samples and formulates a distributionally robust risk constrained optimal control problem which can be solved in a computationally feasible manner.

For the implementation, hexacopter drone as a robot is considered, with realistic physics simulations using Gazebo. Nevertheless, the theoretical formulation is quite general and can be used on other systems like unmanned ground vehicles, self-driving cars and manipulators. I now summarize the three key goals for this work.

1. **Collision Avoidance**: Avoiding collision with surrounding cooperative or non-cooperative agents/obstacles is key for safe navigation. I represent the surrounding in term of 3D polyhedrons, for accurate representation, instead of circular or spherical assumption prevalent in prior work and formulate distributionally robust collision avoidance constraints from past data collected onboard which are then included in the MPC formulation.

2. **Distributed Computation**: A single centralized problem has a large size and is computationally infeasible for real-time deployment. Therefore, I present a distributed approach that effectively exploits the parallel computation and inter-agent communication while ensuring robustness to prediction and measurement error.

3. **Handling Uncertainty**: Finally, risk-sensitive behaviour of the agent depends on how the uncertainty in handled by the MPC. Based on MPC predictions of other agents and collected state data, I developed techniques to formulate the data-driven risk constraints for collision avoidance under uncertainty in a principled manner by building upon past works Hota et al. (2019); Hakobyan et al. (2019); Hakobyan and Yang (2021).

Further in this chapter, I highlight the gaps in the existing literature in and show how my work addresses the gap. Chapter 2 describes the system dynamics, and basics of model predictive controllers for drones. The representation of the obstacles as polyhedrons and reformulation of the constraints is elaborated in section 3.1.

Table 1.1: Classification of previous works

Computation	Deterministic	Robust	Chance constraints	CVaR
Single-agent	Zhang et al. (2020)	Soloperto et al. (2019)	Castillo-Lopez et al. (2020); Zhu and Alonso-Mora (2019) Batkovic et al. (2020); de Groot et al. (2021)	Gao et al. (2021); Hakobyan and Yang (2021); Dixit et al. (2022)
Decentralised	Arul and Manocha (2020); Berg et al. (2011); Cheng et al. (2017)	Park and Kim (2020); Kamel et al. (2017)	Gopalakrishnan et al. (2017); Arul and Manocha (2021); Zhang et al. (2021)	**Our work**
Distributed	Firoozi et al. (2020); Luis et al. (2020)	Dai et al. (2022)	Katriniok et al. (2018)	**Our work**

The main theoretical results on handling the uncertainty and distributionally robust constraint formulation is described in section 3.2. In chapter 4, I demonstrate the performance of the proposed approach on a numerical simulator and analyse the effect of changing various parameters. It further shows the simulation results on a more realistic Gazebo simulator with a larger number of agents. Finally, I conclude this work in chapter 5.

1.2 Literature Survey

Motion planning techniques for robotic teams have been studied in the past, nevertheless some important challenges remain in collision avoidance for multi-robot systems, as reviewed in Huang et al. (2019). Authors in Luis et al. (2020) develop a distributed MPC scheme for multiple robots to generate trajectories in real-time. Though the approach has been shown to give excellent experimental results, other agents' current state (and planned states) have been assumed to be known to all

agents for avoiding collisions. This information might not be practically possible to communicate amongst each other, making it challenging to implement in a decentralised manner. Moreover, uncertainties in the actual motion of robots or other dynamic obstacles have not been considered.

Reference Park and Kim (2020) develops a decentralised planner and constructs safe corridors using linear constraints. The authors estimate the reachable area by limiting the velocity or acceleration of other bodies, resulting in robust but conservative constraints. Other optimization-based methods such as Zhang et al. (2020) assume static, deterministic, but complex environments to find the least intrusive trajectories by solving MPC with signed-distance constraints formulation. One significant advantage of this method is that all dimensions of robots are accounted for, which is essential for large-sized robotic agents, but has been currently implemented for a single robot in a 2D world. Other works such as Zhu et al. (2021); Cong et al. (2021) use neural networks to predict the motion of neighbouring robots. Both these works use MPC to impose constraints using predicted states and work with decentralised computation. Robust MPC approaches as studied in Kamel et al. (2017) ensure guaranteed safety but lead to a computationally expensive and overly conservative solution.

An alternative to model the uncertainty is using stochastic optimization. Given a probability distribution of possible transitions, probabilistic or chance constraints can be applied to the MPC problem to limit the collision probability Zhu and Alonso-Mora (2019). Building on this, the authors in Castillo-Lopez et al. (2020) propose tighter constraints for a single-agent with uncertain dynamic obstacles using chance constraints. Another work Arul and Manocha (2021) discusses decentralised implementation of MPC with probabilistic OCRA Berg et al. (2011), which basically computes a set of collision-free velocities. Authors in Arul and Manocha (2021) compare both Gaussian and non-Gaussian distribution for evaluating chance constraints. Since chance constraints give probabilistic guarantees over large sample runs, it fails for worst-case scenarios. Hence, a hybrid approach was proposed in Brüdigam et al. (2021) that computes two trajectories: a fail-safe trajectory and chance constrained stochastic MPC and applies the appropriate controller, giving a safety guarantee in worst-case scenarios.

Recently, the risk measure Conditional Value-at-risk (CVaR) has been used in robotics. The CVaR of a random loss is equal to the conditional expectation of the loss within the $(1-\alpha)$ worst-case quantile of the loss distribution Hakobyan et al. (2019). Authors in Hakobyan et al. (2019) propose constraints on the CVaR values, which can assess the worst-case tail events of a probability distribution. This formulation is, so far, one of the most appropriate quantification of risk associated with the motion plan as it also bounds the magnitude of constraint violation. Though Hakobyan et al. (2019) considers Gaussian randomness in obstacle motion, the work Hakobyan and Yang (2021) shows that this formulation works for collected sample data from observation of movement. However, these constraints are mostly derived for the case where the controlled object is a point mass and are studied in a centralized/single-robot setting. In addition, the constraints formulated in Hakobyan and Yang (2021) does not distinguish between solutions when collision does not take place, leading to somewhat risky maneuvers in the computed trajectories. In this work, I formulate the CVaR constraint on the uncertain distance between the ego robot and other moving robots leading to safe trajectories and distributed computation.

Table 1.1 summarises the previous works in this domain and highlights the gap in the literature. It can be observed that no previous work utilises risk-based constraints for decentralised/distributed multi-agent systems. Furthermore, only few works Zhang et al. (2020); Hakobyan and Yang (2021); Gao et al. (2021); Soloperto et al. (2019); Firoozi et al. (2020) consider the polyhedral shapes for obstacles. Authors in Gao et al. (2021); Soloperto et al. (2019) focus primarily on autonomous driving scenarios. Our work aims to bridge this gap by developing risk-sensitive MPC for multi-agent systems with polyhedral obstacle representation.

1.3 Contributions

I have considered a multi-robot system where each robot solves a MPC problem subject to constraints on collision avoidance. At each time, an agent collects samples of the prediction error between the current position of other robots and the position of the other robots predicted in the past in the *decentralized* setting or shared by the other robots in the past in the *distributed* setting. The collision avoidance

constraints are then formulated as distributionally robust CVaR constraints on the distance between the controlled object and a polyhedral obstacle parameterized by the predicted position and the uncertain prediction error. In other words, we require the CVaR of the distance to be bounded for a family of distributions of the uncertain prediction error close (in the sense of Wasserstein metric) to the past samples of prediction errors. While this class of constraints are infinite-dimensional, I derive a tractable finite-dimensional approximations by leveraging convex and minmax duality results for distributionally robust optimization problems. Finally I demonstrate the efficacy of the proposed approach in a multi-drone navigation setting implemented in Gazebo platform with multi-agent MPC being executed in parallel processors with realistic inter-agent communication protocols in place.

Chapter 2

Problem Description

2.1 System Dynamics

While the proposed formulation is general enough for multiple types of multi-robot systems, in order to keep the discussion grounded, I specifically consider drones to evaluate the effectiveness of the proposed approach in this project. As seen in Figure 2.1, two reference frames are considered: (i) inertial frame $\mathcal{A}$ with axes $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$ and (ii) body frame $\mathcal{B}$ with axes $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$. The position $\mathbf{r}$ in $\mathcal{A}$ is represented by $\begin{bmatrix} x & y & z \end{bmatrix}^\top$. The $[Z-X-Y]$ Euler angle notation is used to represent the orientation of the drone. The orientation angles are roll ϕ (along X-axis), pitch θ (along Y-axis), and yaw ψ (along Z-axis). The angular velocity in the body-fixed frame is given by $\boldsymbol{\omega} = p\mathbf{b}_1 + q\mathbf{b}_2 + r\mathbf{b}_3$. The state of the system is represented by the position and velocity, the Euler Angles (in the $[Z-X-Y]$ sequence), and the angular velocities. The control inputs are the thrust and the moments about the three axes.

The state and control inputs are denoted by

$$\mathbf{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & p & q & r \end{bmatrix}^\top,$$

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^\top.$$

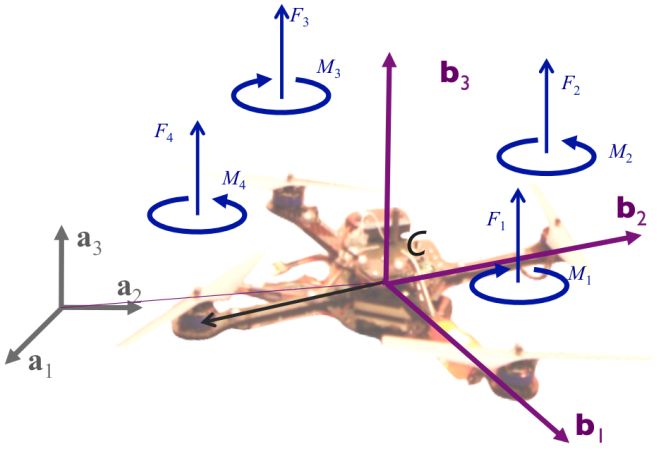

FIGURE 2.1: The inertial and body-fixed reference frames, and forces and moments by each of the rotors. Image obtained from 'Aerial Robotics' on Coursera by Prof. Vijay Kumar.

The dynamic equations for translational motion of the drone are

$$m\ddot{\mathbf{r}} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix}, \tag{2.1}$$

where the components are denoted in the inertial frame along $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$; m and g represent mass and gravitational acceleration respectively, and R is the rotation matrix from $\mathcal{B}$ to $\mathcal{A}$. The input

$$u_1 = F_1 + F_2 + F_3 + F_4,$$

is the combined thrust obtained, where F_i is the thrust produced by i^{th} propeller. The equations for the rotation are:

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{2.2}$$

where the components are along the body-fixed principal axes $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$; I is the inertia matrix and L is the distance between the rotor and the center of mass of

the drone; the rotational control inputs are

$$\begin{aligned} u_2 &= L(F_2 - F_4), \\ u_3 &= L(F_3 - F_1), \\ u_4 &= M_1 - M_2 + M_3 - M_4, \end{aligned}$$

where u_2, u_3, and u_4 are the moments along the body axes and M_i is the moment produced by i^{th} propeller.

The dynamics of the multirotor can be represented in the standard $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ as represented in Sabatino (2015) form as follows

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \frac{1}{m}(c\psi s\theta + c\theta s\phi s\psi)u_1 \\ \frac{1}{m}(s\psi s\theta - c\theta s\phi c\psi)u_1 \\ \frac{1}{m}(c\phi c\theta)u_1 - g \\ p(c\theta) + r(s\theta) \\ p(s\theta s\phi/c\phi) + q - r(c\theta s\phi/c\phi) \\ -p(s\theta/c\phi) + r(c\theta/c\phi) \\ \frac{1}{I_{xx}}(u_2 - (I_{zz} - I_{yy})qr) \\ \frac{1}{I_{yy}}(u_3 - (I_{xx} - I_{zz})pr) \\ \frac{1}{I_{zz}}(u_4 - (I_{yy} - I_{xx})pq) \end{bmatrix}, \tag{2.3}$$

where I has only the diagonal components I_{xx}, I_{yy}, I_{zz} as $\mathbf{b}_1$ along $\mathbf{b}_2$, $\mathbf{b}_3$ principal axes, and I denote $c\phi := \cos(\phi)$ and $s\phi := \sin(\phi)$ and so on, for better readability.

Specific limits on states and inputs are applied for all agent at all times as deterministic constraints $\mathbf{x} \in \mathcal{X}$ and $\mathbf{u} \in \mathcal{U}$. These include environment boundaries, quadrotor orientation bounds, actuators limit, among others. Full state dynamics (2.3) are utilized for better results and greater angular operation range at the cost of higher computational load.

2.2 Baseline MPC formulation

Model Predictive Control (MPC) is a recursive finite-horizon optimal control problem. Figure 2.2 depicts the working principle of MPC where at each discrete time step k, a constrained optimization problem is solved to compute an optimal control sequence and the resulting state trajectory that minimizes deviation from reference trajectory while satisfying constraints. Only the first of the computed control sequence is applied to the plant, and the system transition to the next state and the process is repeated.

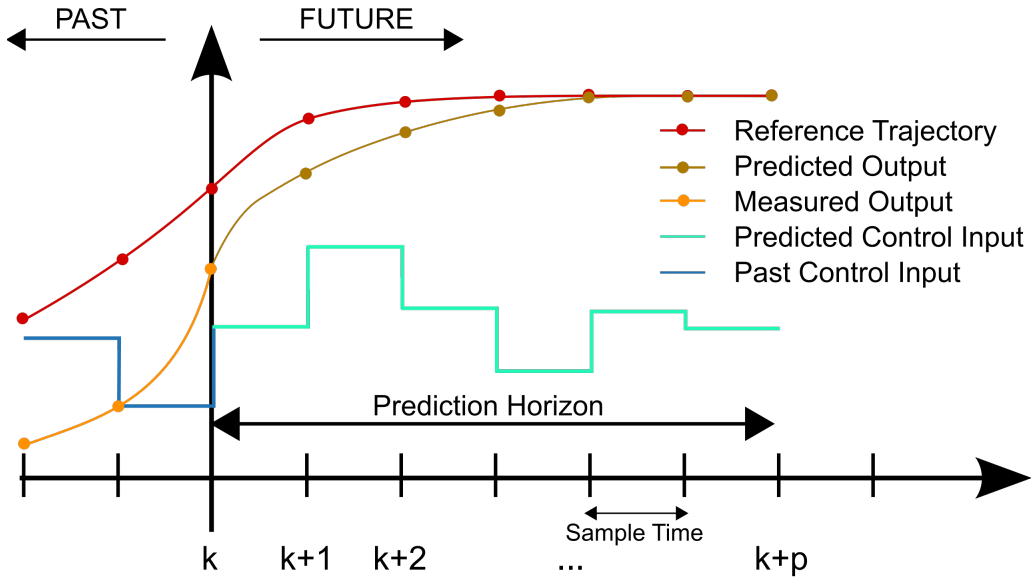

FIGURE 2.2: Visual representation of receding-horizon control or MPC. Image obtained from Wikipedia.

Here, a multi-agent system comprising of N individual mobile robots or agents is considered. The goal of each agent is to reach a (agent-specific) final position or track a desired trajectory. For both objectives, the cost function for an agent i at time $k+l$ computed at time k is given by

$$\begin{aligned} J_i(k+l|k) = \mathbf{x}_{e,i}(k+l|k)^\intercal Q\, \mathbf{x}_{e,i}(k+l|k) \\ + u_i(k+l|k)^\intercal R u_i(k+l|k), \end{aligned} \tag{2.4}$$

where $\mathbf{x}_{e,i}(k+l|k) = \mathbf{x}_{\text{ref},i}(k+l) - \mathbf{x}_i(k+l|k)$ is the difference between the desired state at time $k+l$ and the state at time $k+l$ predicted at k. The second term penalises the control effort with $u_i(k+l|k)$ being the control input for time $k+l$

computed at time k, and the matrices Q and R are assumed to be positive definite. The finite horizon optimal control problem for agent i is given by

$$\begin{aligned} \min_{\mathbf{x}_i, \mathbf{u}_i} \quad & \sum_{l=1}^{T} J_i(k+l|k) \\ \text{s.t.} \quad & x_i(k+l|k) = f_i(x_i(k+l-1|k), u_i(k+l-1|k)), \\ & x_i(k+l|k) \in \mathcal{X}_i,\ u_i(k+l-1|k) \in \mathcal{U}_i, \\ & \mathcal{C}(z_i(k+l|k), z_j(k+l|k)) \leq 0, \quad \forall i \neq j, \\ & \text{for all } l \in [T], j \in [N], \end{aligned} \tag{2.5}$$

where f_i captures the discrete-time dynamics, $\mathcal{X}_i$ and $\mathcal{U}_i$ denote the deterministic constraints on states and control inputs for agent i, and $\mathcal{C}(z_i, z_j) \leq 0$ denote the collision avoidance constraints between two agents i and j with positions z_i and z_j, respectively. The position is assumed to be part of the state vector. The above problem is solved for states $\mathbf{x}_i$, control inputs $\mathbf{u}_i$ for agent i at each time step. For ease of notation, we can define $[N] := \{1, ..., N\}$. Thus, in order to ensure safe navigation of agent i, we need to compute optimal control inputs such that agent i does not collide with any other agent $j \neq i$ over the prediction horizon.

Chapter 3

Collision Avoidance for Polyhedral Objects

3.1 Obstacle Occupancy Modeling

From the perspective of agent i, other agents act as obstacles which occupy some space that is forbidden for it. The occupancy set is modelled as convex polyhedral sets composed as union of multiple half-spaces. In particular, the space occupied by obstacle m is represented as

$$\mathcal{O}^m = \{p \in \mathbb{R}^3 : A^m p \leq b^m\}, \;\; m \in [M], \tag{3.1}$$

where M denotes the total number of obstacles, $A^m \in \mathbb{R}^{n_m \times 3}$ and $b^m \in \mathbb{R}^{n_m}$ are constant matrices and vectors that represent the position and orientation of the obstacle, and n_m is the number of half spaces required to model obstacle m. It is assumed that any non-convex obstacle can be conservatively approximated to an enclosed polyhedron.

As the agents are considered to be dynamic obstacles, the polyhedral representation of each obstacle is also a function of time. As a result, agent i needs to know the predicted occupancy sets of all other agents T steps into the future. This information is not readily available. There are two main paradigms in the literature based on how this information is accessed. In the *distributed* approach, agents exchange their

optimal solution (future trajectory) with others. Thus, at each time step k, agent i receives the presently computed MPC solution which includes future position and orientation information of other agents. However, other agents may not follow the current optimal trajectory and as a result, this approach is not robust to this future deviation by other agents. In the *decentralized* approach, there is no inter-agent communication, and most prior works assume that agent i predicts the future position of other agents assuming that other agents will continue to move with their present velocity. Thus, this assumption is rather naive and often leads to incorrect predictions and collisions.

In this work, I propose a principled approach to robustify against such prediction inaccuracies in both distributed and decentralized schemes. I begin by assuming that at time k, the controlled agent i has access to anticipated future position of other agents over the prediction horizon, i.e., it is aware of $z_j(k+l|k)$ for $l \in [T], j \in [N]$. In the distributed case, this information is shared by other agents and corresponds to the solution of their MPC problem at the previous time step. In the decentralized case, this information is predicted by the agent under the constant velocity assumption. With regard to orientation, it is assumed that the present occupancy set of an obstacle agent j, denoted by $\mathcal{O}^j_{k|k}$ and characterized by $A^j(k)$ and $b^j(k)$, is known to agent i, and the future orientation of agent j remains unchanged from its present orientation. Under the above assumption, the uncertain obstacle space of agent j predicted at time k for time step $k+l$ is formally stated as

$$\mathcal{O}^j_{k+l|k} = \mathcal{O}^j_{k|k} + z_j(k+l|k) + w^{(l)}_j, \tag{3.2}$$

where $w^{(l)}_j \in \mathbb{R}^3$ is the difference between the true position and the anticipated position of this agent l steps into the future. A similar set up with linearly perturbed uncertainty sets was considered in Hakobyan et al. (2019) in the single-agent case.

While the probability distribution of $w^{(l)}_j$ is unknown, the controlled agent has access to samples of $w^{(l)}_j$ from past trajectory as follows: at each time step k, when position of an agent j is observed, it is compared with the predictions of the position of agent j obtained in previous T time steps and compute the difference as samples of the the uncertain parameter $w^{(l)}_j, l \in [T]$. In other words, $z_j(k|k) - z_j(k|k-q)$ is treated as a sample of $w^{(q)}_j$ which is the q-step prediction error. Thus, at time k, a set of

samples of $w_j^{(l)}, \quad l \in [T]$ is collected as described above. Now, the formulation of data-driven distributionally robust collision avoidance constraints is described.

Given the uncertain occupancy set defined in (3.2), the collision avoidance constraint is now stated as

$$\begin{aligned} F\big(z_i(k+l|k), z_j(k+l|k), w_j^{(l)}\big) := \\ d_{\min} - \text{dist}\big(z_i(k+l|k),\ \mathcal{O}^j_{k+l|k}\big) \leq 0, \end{aligned} \tag{3.3}$$

where the *dist* function is the distance between the agent position $z_i(k+l|k)$ and obstacle space $\mathcal{O}^j_{k+l|k}$. The above constraint is required to hold for all neighbors $j \in [N]$ and time $l \in [T]$ in the MPC problem of agent i at time k.

Reformulation in the deterministic setting

Before introducing the distributionally robust risk sensitive version of the above constraints, I first present the reformulation of the above in the deterministic regime. Consider the occupancy set $\mathcal{O}^m$ defined in (3.1). The distance between an agent at position z_i and $\mathcal{O}^m$ is given by

$$\begin{aligned} \text{dist}(z_i, \mathcal{O}^m) &:= \min_{r \in \mathcal{O}^m} ||z_i - r|| \\ &= \min_d (||d|| : A^m(z_i + d) \leq b^m). \end{aligned} \tag{3.4}$$

It is evident that the constraint (3.3) with the above definition of distance is non-trivial to impose on the optimization problem since the distance function itself involves solving an optimization problem. The following result from Zhang et al. (2020) proposes an equivalent tractable form for these constraints by leveraging convex duality.

Proposition 3.1 (Zhang et al. (2020))**.** *For an obstacle set* $\mathcal{O} = \{p \in \mathbb{R}^3 : A^m p \leq b^m\}$*, we have*

$$\begin{aligned} &dist(z_i, \mathcal{O}^m) \geq 0 \iff \\ &\quad \exists \lambda \geq 0 : (A^m z_i - b^m)^\intercal \lambda \geq 0,\ ||(A^m)^\intercal \lambda||_2 \leq 1. \end{aligned} \tag{3.5}$$

Thus, if there exists λ satisfying the above constraints, then the collision constraint is satisfied (the distance between the controlled agent and the obstacle is non negative). As a result, these conditions can be encoded as constraints with λ being an additional decision variable in the MPC formulation. I now introduce the distributionally robust framework to appropriately handle the uncertain constraint (3.3).

3.2 Data-Driven Distributionally Robust Constraint Formulation

Note that the constraint function $F(z_i, z_j, w_j)$ defined in (3.3) is uncertain with the distribution of w_j not being known. However, a collection of N_s samples of w_j is available with MPC controller of agent i denoted by $\{\hat{w}_{j,n}\}_{n\in[N_s]}$. I leverage these available samples to define data-driven distributionally-robust conditional value-at-risk (CVaR) constraints on the function F as

$$\sup_{\mathbb{P}\in\mathcal{M}_{N_s}^{\theta}} \mathrm{CVaR}_{1-\alpha}^{\mathbb{P}}\left[F\left(z_i, z_j, w_j\right)\right] \leq 0, \tag{3.6}$$

where

- the CVaR of a random loss X with distribution $\mathbb{P}$, is equal to the conditional expectation of the loss within the α worst case quantile of the loss distribution, i.e.,
$$\mathrm{CVaR}_{1-\alpha}^{\mathbb{P}}(X) := \inf_{z\in\mathbb{R}} \left[\alpha^{-1}\mathbb{E}[(X+z)^{+}] - z\right], \tag{3.7}$$
where $(x)^{+} = \max\{x, 0\}$. Consequently, CVaR constraint aims to constrain the value at the tail distribution.

- the set $\mathcal{M}_{N_s}^{\theta}$ is a family of probability distributions that are within a Wasserstein distance θ from the empirical distribution induced by the N_s samples $\{\hat{w}_{j,n}\}_{n\in[N_s]}$; the formal definition of the ambiguity set is omitted in the interest of space and can be found in Hota et al. (2019); Hakobyan and Yang (2021).

Following the definition of CVaR, the constraint (3.6) assumes the form:

$$\sup_{\mathbb{P}\in\mathcal{M}_{N_s}^{\theta}} \inf_{t\in\mathbb{R}} \mathbb{E}^{\mathbb{P}}\left[(F\left(z_i, z_j, w_j\right)+t)^{+}-t\alpha\right] \leq 0. \tag{3.8}$$

The above constraint is infinite-dimensional due to the supremum being over a family of probability distributions. In the remainder of this subsection, it is shown how the above constraint can be approximated and reformulated into a finite-dimensional constraint which can be solved via off-the-shelf solvers.

First we observe that since (sup inf [ ]) $\leq$ (inf sup [ ]), the constraint

$$\inf_{t\in\mathbb{R}} \sup_{\mathbb{P}\in\mathcal{M}_{N_s}^{\theta}} \mathbb{E}^{\mathbb{P}}\left[(F\left(z_i, z_j, w_j\right)+t)^{+}-t\alpha\right] \leq 0. \tag{3.9}$$

is sufficient for (3.8) to hold true. Now, the inner supremum problem in the above equation can be reformulated as shown in Hota et al. (2019) to an infimum problem, which then combined with the the infimum over t yields the following set of constraints that are sufficient for (3.8) to hold true:

$$\lambda_\theta\theta - t\alpha + \frac{1}{N_s}\sum_{n=1}^{N_s} s_n \leq 0, \tag{3.10a}$$

$$s_n \geq \sup_{w_j\in\Omega_j} [F\left(z_i, z_j, w_j\right) + t - \lambda_\theta||w_j - \hat{w}_{j,n}||_2], n \in [N_s], \tag{3.10b}$$

$$s_n \geq 0, \quad t \in \mathbb{R}, \quad \lambda_\theta \geq 0,$$

where θ is the radius of the ambiguity set. I now focus on reformulating the semi-infinite constraint (3.10b) which involves an optimization problem over the support of w_j denoted by Ω_j in the following two major steps.

Step 1: Reformulation of (3.10b)**.**

From the definition of the constraint function F in (3.3), equation (3.10b) is expressed for sample n as

$$\begin{aligned} s_n &\geq \sup_{w_j\in\Omega_j}\Big[d_{\min} - \operatorname{dist}\left(z_i, \mathcal{O}^j\right) + t - \lambda_\theta||w_j - \hat{w}_{j,n}||_2\Big] \\ &= d_{\min} + t - \inf_{w_j\in\Omega_j}\Big[\operatorname{dist}\left(z_i, \mathcal{O}^j\right) + \lambda_\theta||w_j - \hat{w}_{j,n}||_2\Big]. \end{aligned}$$

Based on the representation (3.1), when $\mathcal{O}^j_{k|k} = \{p \in \mathbb{R}^3 | Ap \leq b\}$, then the distance function in the above equation is given by

$$\begin{aligned} \min \quad & ||t|| \\ \text{s.t.} \quad & A\left(z_i + t - z_j - w_j\right) \leq b, \end{aligned} \tag{3.11}$$

and following the strong duality result in Proposition 3.1, it can be stated equivalently as

$$\begin{aligned} \max_{\lambda \geq 0} \ & \left[A(z_i - z_j - w_j) - b\right]^\intercal \lambda \\ \text{s.t.} \ & ||A^\intercal \lambda||_2 \leq 1. \end{aligned} \tag{3.12}$$

Substituting the above in the inequality involving s_n yields

$$\begin{aligned} s_n \geq & d_{\min} + t - \inf_{w_j \in \Omega_j} \Big[\max_{\lambda \geq 0,\ ||A^\intercal \lambda||_2 \leq 1} \big\{ [A(z_i - z_j - w_j) \\ & - b]^\intercal \lambda \big\} + \lambda_\theta ||w_j - \hat{w}_{j,n}||_2 \Big]. \end{aligned} \tag{3.13}$$

Once again, note that since $(\sup \ \inf \ [\ ]) \leq (\inf \ \sup \ [\ ])$, the inequality

$$\begin{aligned} s_n \geq & d_{\min} + t - \max_{\lambda \geq 0,\ ||A^\intercal \lambda||_2 \leq 1} \Big[\inf_{w_j \in \Omega_j} \big\{ \big[A(z_i - z_j - w_j) \\ & - b\big]^\intercal \lambda + \lambda_\theta ||w_j - \hat{w}_{j,n}||_2 \big\} \Big], \end{aligned} \tag{3.14}$$

is sufficient for (3.13) to hold.

Rearranging the equations, we obtain

$$\begin{aligned} s_n \geq & d_{\min} + t - \max_{\lambda \geq 0,\ ||A^\intercal \lambda||_2 \leq 1} \Big([A(z_i - z_j) - b]^\intercal \lambda \\ & + \inf_{w_j \in \Omega_j} \big[\lambda_\theta ||w_j - \hat{w}_{j,n}||_2 - w_j^\intercal (A^\intercal \lambda) \big] \Big). \end{aligned} \tag{3.15}$$

Step 2: Reformulation of the infimum with respect to w_j.

The infimum term with respect to w_j can be written as

$$\inf_{w_j \in \Omega_j} \big[-(\lambda^\intercal A w_j - \lambda_\theta ||w_j - \hat{w}_{j,n}||_2)\big], \tag{3.16}$$

$$\iff -\sup_{w_j \in \Omega_j} \big[\lambda^\intercal A w_j - \lambda_\theta ||w_j - \hat{w}_{j,n}||_2\big]. \tag{3.17}$$

When the support of w_j is a polyhendron, i.e., $\Omega_j = \{w \in \mathbb{R}^3 | C_j w \leq h_j\}$, the authors in Hota et al. (2019) showed that the supremum term above is equivalent to

$$\begin{aligned} \min_{\eta_{j,n} \geq 0} \quad & (A^\intercal \lambda - C_j^\intercal \eta_{j,n})^\intercal \hat{w}_{j,n} + \eta_{j,n}^\intercal h_j \\ \text{s.t.} \quad & ||A^\intercal \lambda - C_j^\intercal \eta_{j,n}||_2 \leq \lambda_\theta. \end{aligned} \tag{3.18}$$

As a result, (3.17) can be stated equivalently as

$$\begin{aligned} \max_{\eta_{j,n} \geq 0} \quad & -\big[(A^\intercal \lambda - C_j^\intercal \eta_{j,n})^\intercal \hat{w}_{j,n} + \eta_{j,n}^\intercal h_j\big] \\ \text{s.t.} \quad & ||A^\intercal \lambda - C_j^\intercal \eta_{j,n}||_2 \leq \lambda_\theta. \end{aligned} \tag{3.19}$$

Consequently, (3.15) can be stated equivalently as

$$\begin{aligned} s_n \geq & d_{\min} + t - \max_{\lambda \geq 0,\, ||A^\intercal \lambda||_2 \leq 1} \Big([A(z_i - z_j) - b]^\intercal \lambda \\ & + \max_{\substack{\eta_{j,n} \geq 0, \\ ||A^\intercal \lambda - C_j^\intercal \eta_{j,n}||_2 \leq \lambda_\theta}} -[(A^\intercal \lambda - C_j^\intercal \eta_{j,n})^\intercal \hat{w}_{j,n} + \eta_{j,n}^\intercal h_j] \Big). \end{aligned} \tag{3.20}$$

Since the maximum terms on the R.H.S are preceded by a negative sign, the following constraints are sufficient to guarantee that the constraint in (3.20) holds:

$$\begin{aligned} s_n \geq & \, d_{\min} + t - \Big([A(z_i - z_j) - b]^\intercal \lambda \\ & - [(A^\intercal \lambda - C_j^\intercal \eta_{j,n})^\intercal \hat{w}_{j,n} + \eta_{j,n}^\intercal h_j] \Big), \end{aligned} \tag{3.21a}$$

$$\lambda \geq 0, \quad ||A^\intercal \lambda||_2 \leq 1, \tag{3.21b}$$

$$\eta_{j,n} \geq 0, \quad ||A^\intercal \lambda - C_j^\intercal \eta_{j,n}||_2 \leq \lambda_\theta. \tag{3.21c}$$

When the support Ω_j is not known and is assumed to be $\mathbb{R}^3$, we do not need the multipliers $\eta_{j,n}$, and consequently, the following set of constraints

$$s_n \geq d_{\min} + t - \Big([A(z_i - z_j) - b]^\intercal \lambda - [(A^\intercal \lambda)^\intercal \hat{w}_{j,n}] \Big), \tag{3.22a}$$

$$\lambda \geq 0, \quad ||A^\intercal \lambda||_2 \leq 1, \quad ||A^\intercal \lambda||_2 \leq \lambda_\theta. \tag{3.22b}$$

are sufficient to guarantee that (3.20) holds.

To summarize, the original distributionally CVaR collision avoidance constraint (3.6) can be approximated as

$$\lambda_\theta \theta - t\alpha + \frac{1}{N_s} \sum_{n=1}^{N_s} s_n \leq 0, \tag{3.23a}$$

$$d_{\min} + t - s_n \leq [A(z_i - z_j) - b]^\intercal \lambda - [\lambda^\intercal A \hat{\omega}_{j,n}] \tag{3.23b}$$

$$\lambda \geq 0, \ ||A^\intercal \lambda||_2 \leq \min(1, \lambda_\theta), \tag{3.23c}$$

$$\lambda_\theta \geq 0, \qquad t \in \mathbb{R}, \qquad s_n \geq 0 \quad \forall n \in [N_s]. \tag{3.23d}$$

Thus, the MPC problem for agent i has the above set of constraints for each neighbor j with z_i, z_j, A, b being replaced by $z_i(k+l|k), z_j(k+l|k), A^j(k), b^j(k)$ for all time steps over the horizon $l \in [T]$.

Chapter 4

Simulation Results

First, I use a numerical simulator with nonlinear dynamics discretized with sampling time 1 ms. The sampling time for MPC is chosen to be 0.1 s. In this setup, all agents solve the MPC problem synchronously and the numerical simulator gives the next state of the drone. All these independent processes run on a workstation with AMD Ryzen 5800H chipset and 16GB RAM. In particular, Python `multiprocessing` module is used to launch parallel nodes for each agent interacting with the common simulation. The odometry data as well as predictions are communicated among each other using Robot Operating System (ROS). The MPC problem is solved using a nonlinear programming solver IPOPT Wächter and Biegler (2006). This implementation uses the MA27 as the linear solver for IPOPT and the "do-mpc" interface Lucia et al. (2017) for the solver, that also uses CasADi package Andersson et al. (2019).

Subsequently, for more realistic dynamics, Gazebo physics simulations are used for the AsTec Firefly hexacopter model using RotorS for low-level control Furrer et al. (2016). Gazebo runs on the PC interacting with each agents for actuator commands and returning the odometry data.

4.1 Distributed Setting

In this subsection, I demonstrate my proposed formulation on numerical simulations in a distributed setting. To simplify the analysis, I start with a case with two agents, trying to cross each other on a straight path. Each agent considers itself to be a point mass, whereas the surrounding obstacles are assumed to be polyhedral. Each obstacle/agent is assumed to be a rectangular prism with a square cross-section of 2 m. The length of prism is assumed very high to visualise planar collision avoidance. Unless stated otherwise, all subsequent results are obtained with a sample size of $N_s = 10$ and the prediction horizon of $T = 20$ steps.

Figure 4.1 shows the trajectory of both agents when the risk tolerance parameter (of the CVaR function) $\alpha = 0.1$ and the Wasserstein radius $\theta = 0.001$. The mean and standard deviation in errors in the predictions, i.e, the actual position of an obstacle agent and the optimal MPC solution of that agent l steps before, is illustrated in Figure 4.2. The figure shows that deviations increases across the time horizon and does not necessarily have zero mean. Therefore, it is necessary to robustify the trajectories against these errors.

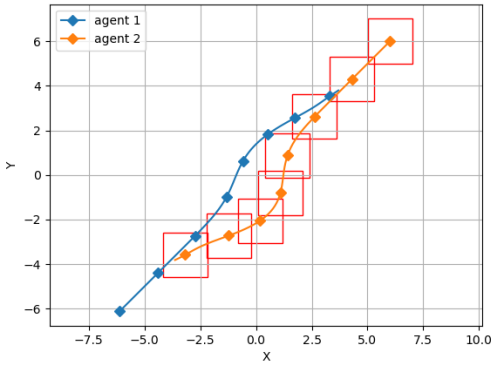

FIGURE 4.1: Trajectory as seen by agent 1 for $\alpha = 0.1$ and $\theta = 0.001$.

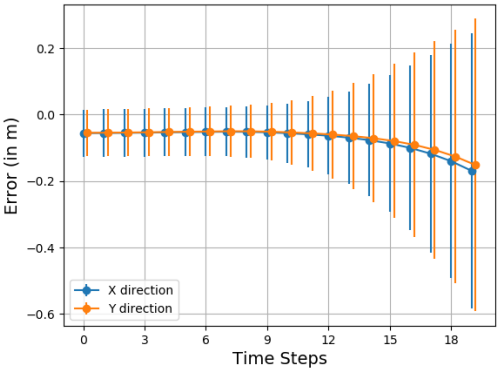

FIGURE 4.2: Prediction errors across time horizon.

Figure 4.3 depicts the average value of the minimum distance between the agents over 50 runs. Higher values of θ result in larger ambiguity sets around the collected samples which leads to more robust trajectories; this is observed from the figure which shows that average minimum distance is larger when θ is larger. As α increases, agents are more tolerant towards risk of collision, and as a result opt for risky

trajectories which lead to reduced value of average minimum distance. The behaviour is more sensitive to θ for smaller values of α. The integration of distance from goal, in figure 4.4 shows weakly increasing trend, denoting that the agents move closer to each other but take a slightly longer route.

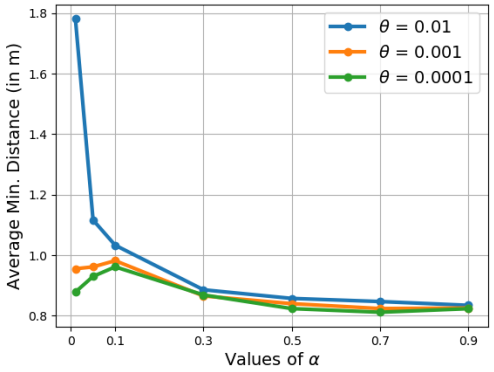

FIGURE 4.3: Average minimum distance with increasing α values.

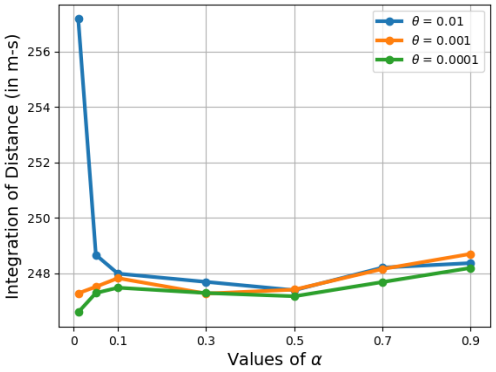

FIGURE 4.4: Integration of distance with increasing α values.

4.2 Variations with added noise

Both, observed states and the predictions now contain a random noise based on Gaussian distribution. The standard deviation of the distribution is varied to compare the effect of changing noise values. The standard deviation for the prediction noise is fixed to be three times of that for the state observations. Hence, for case with std. dev. of 1m in state observation, the predictions contain noise with std. dev. of 3m. For this case, we have $\theta = 10^{-3}$, with a time horizon of 20, and sample size of 10. Two values of $\alpha = 0.05, 0.5$ are used.

We can see from figure 4.5, that the percentage of collisions is zero for low α value, whereas we begin to see collisions for higher α value with higher state noise of 0.6m standard deviation. The higher noise levels of 0.8 are unsustainable for $\alpha = 0.5$. The figure 4.6 shows the average of the minimum distance between the agents. It can be seen that with increasing levels of noise, the low α configuration takes more risk-averse paths leading to higher separation. The sudden dip in the plot for $\alpha = 0.5$ is caused due to higher collision rate. In the figure 4.7, the integration of the distance

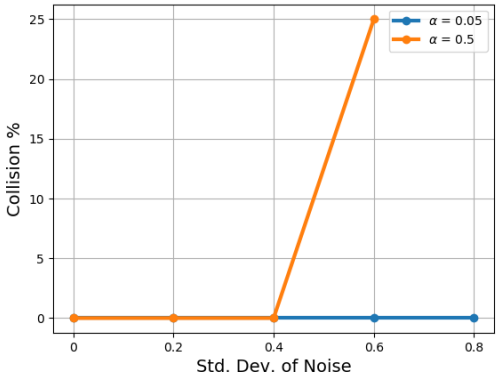

FIGURE 4.5: Collision percentage with increasing noise level.

from the goal position, increases with increasing noise depicting longer paths. Drop in the value for $\alpha = 0.5$ occur as only the non-colliding test cases are considered.

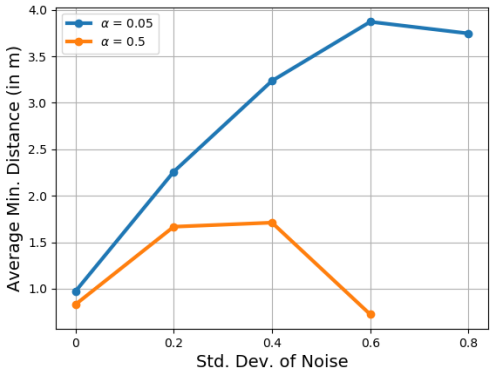

FIGURE 4.6: Avg. min. distance with increasing noise level.

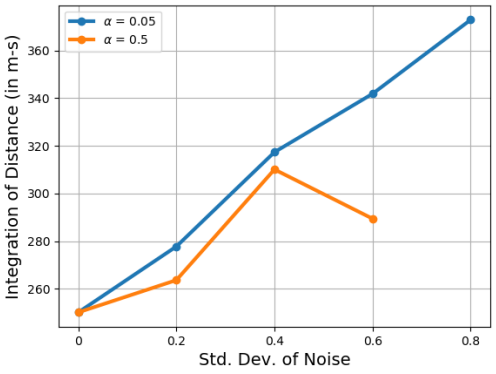

FIGURE 4.7: Integration of distance with increasing noise level.

Figure 4.8, shows the mean of errors across time horizon for increasing levels of noise. As expected, the standard deviation in error increases with higher levels of noise. Also, as seen in figure 4.9, for noise levels of 0.4, we see that the added noise dominated the error in prediction, giving uniform and high standard deviation across the time horizon.

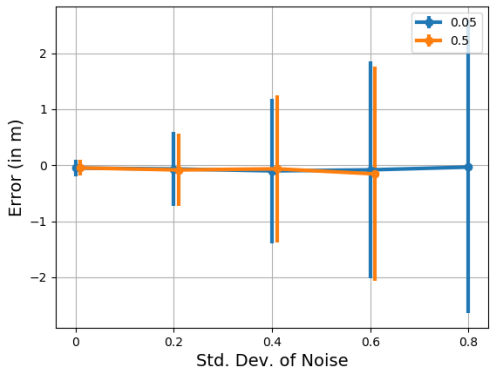

FIGURE 4.8: Prediction errors with increasing noise level.

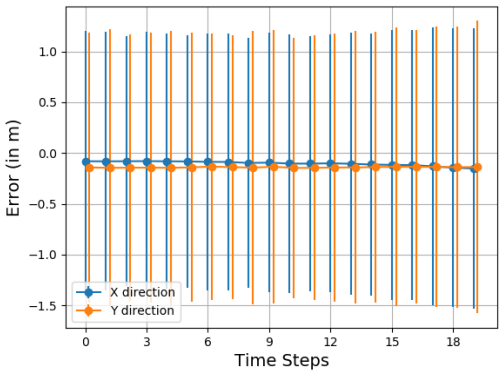

FIGURE 4.9: Prediction errors across time horizon..

4.3 Comparison with the baseline

In this subsection, I compare my formulation with the baseline deterministic MPC solutions in both *distributed* and *decentralised* settings with two agents. In the distributed setting, each agent has the access to the optimal MPC trajectories of other agents and solves a deterministic MPC problem avoiding collision with the predicted trajectories. In the decentralised setting, an agent solves a deterministic MPC problem avoiding collision with the predicted trajectories of others under a constant velocity assumption. These baseline solutions are compared with the data-driven distributionally robust CVaR constrained solutions. To increase the level of uncertainty, I add Gaussian noise to the perceived states and predictions of other agents at each time step. The most recent samples based on user-specified sample size are used, with $\alpha = 0.05$ and $\theta = 0.001$. Fifty simulations with each parameter configuration are conducted.

As evident from Figure 4.10, collisions occur for the baseline MPC as the noise levels increase; with the collision percentage being higher for the decentralised case which incorrectly predicts the future positions of other agents based on a constant velocity assumption. The proposed formulation (CVaR-MPC) does not give collision even for very high noise standard deviation in any of the simulations. In Figure 4.11, average of minimum distance between agents with increasing noise levels is plotted. We can observe that CVaR-MPC takes a more risk-averse approach, causing higher separation as the level of uncertainty increases. In contrast, baseline approaches lead

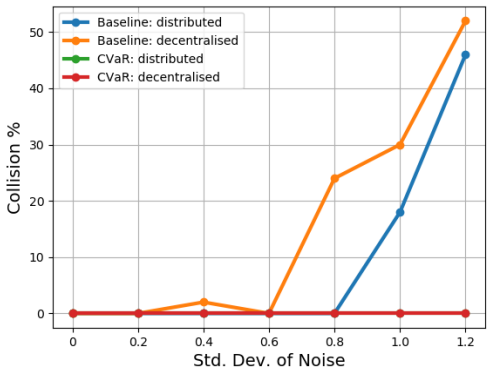

FIGURE 4.10: Percentage of collisions with increasing noise values ($\alpha = 0.05$ and $\theta = 0.001$).

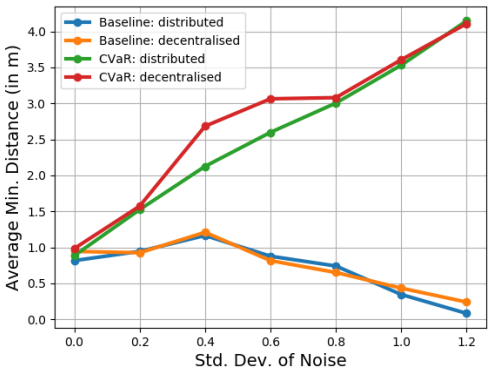

FIGURE 4.11: Avgerage minimum distance with increasing noise values (for $\alpha = 0.05$ and 0.001).

to a higher proportion of collisions leading to a smaller average minimum distance. Thus, the distributionally robust approach enables us to robustify MPC solutions even with a relatively small samples size.

TABLE 4.1: Computation Time (in milliseconds)

Time Horizon Sample Size	10	20	30
5	$19 \pm 2ms$	$40 \pm 13ms$	$63 \pm 18ms$
10	$27 \pm 8ms$	$60 \pm 18ms$	$102 \pm 76ms$
20	$41 \pm 12ms$	$113 \pm 35ms$	$184 \pm 127ms$

4.4 Computation Time

Table 4.1 shows the computation time for different values of sample size for CVaR constraints and prediction horizon of the MPC. This is obtained from the previous numerical simulation setting for the two-agent case. We can observe that most of the configurations have computation the time less than 0.1s or 100ms, which is the step size of MPC optimization. The mean and standard deviation of computation time is higher for increasing time horizon and sample size.

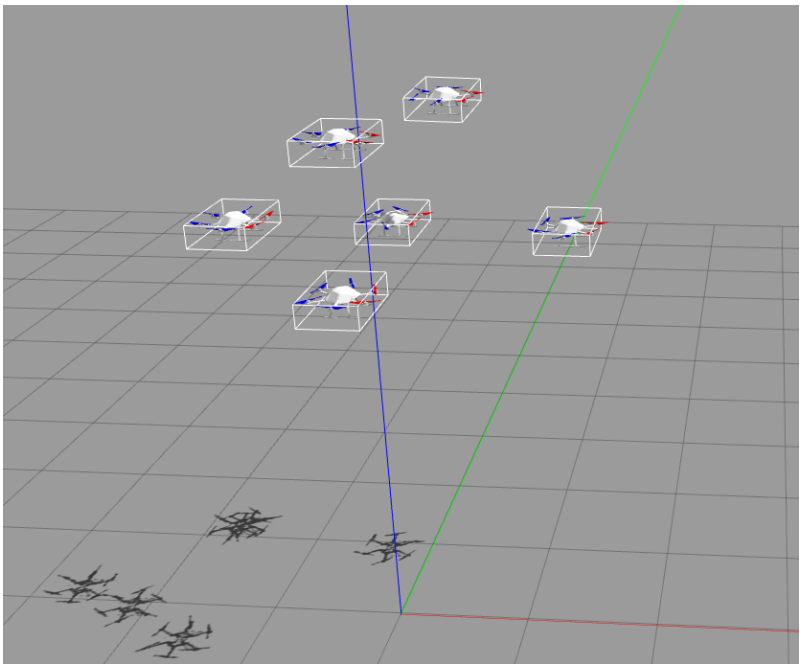

FIGURE 4.12: Snapshot of the Gazebo simulation.

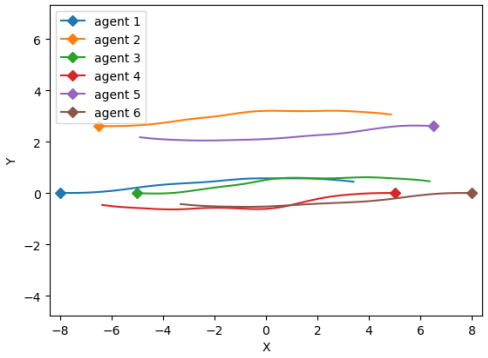

FIGURE 4.13: 2D visualisation of the first mission.

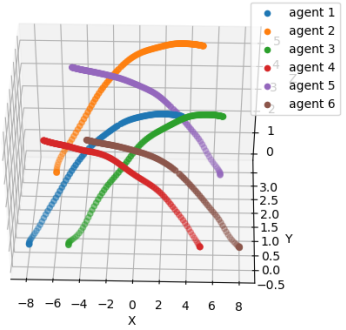

FIGURE 4.14: 3D visualisation of the first mission. Video of the mission can be accessed at `https://bit.ly/3AFp4Uc`

4.5 Gazebo Simulations

For the realistic Gazebo simulations (see fig. 4.12), I validate this approach on similar position exchange mission, as demonstrated in previous reports. The value of α is fixed to 0.1, with a sample size of 10, without adding synthetic noise.

In the first mission, a team of three drones needs to reach the position of other three drones. Figures 4.13 and 4.14 depict the 2D and 3D visualisations of the trajectory. In the second case, six agents in a rectangular formation must reach the opposite side of the rectangle. Similarly, figures 4.15 and 4.16 show the trajectories of all the agents.

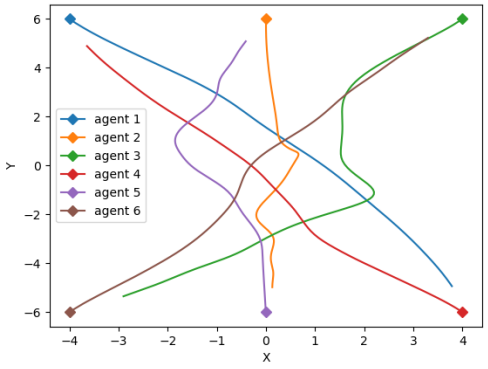

FIGURE 4.15: 2D visualisation of the second mission.

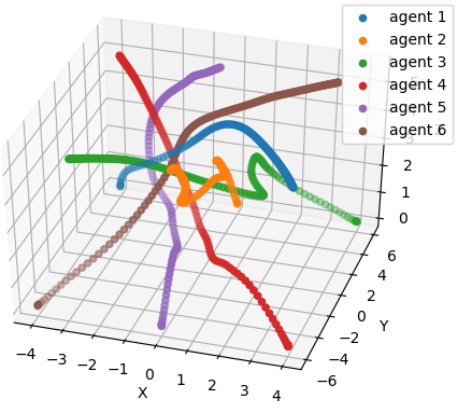

FIGURE 4.16: 3D visualisation of the second mission. Video of the mission can be accessed at `https://bit.ly/3D4o9iW`

4.6 Embedded Implementation

To demonstrate the real-world use case, I implemented the code on commonly used embedded platform, Raspberry Pi. Figure 4.17 shows the workflow of the implementation. Each Raspberry Pi run independently acting as an agent, while the Gazebo simulation runs on the PC. All the communication is handled using ROS messages, which is also used commonly in experimental robotics. The Raspberry Pi is wirelessly connected to the computer by connecting to a shared local network. Figures 4.18 and 4.19 show the 2D and 3D trajectories computed on the Raspberry Pi. I found the computation time for each step to be 0.54s with a standard deviation of 0.18s. Though this is not directly implementable on the experimental platform at this stage, this demonstrates the practical usability of my approach. Optimised C code with sophisticated solvers and more powerful embedded platform would enable using this formulation on real-world robots.

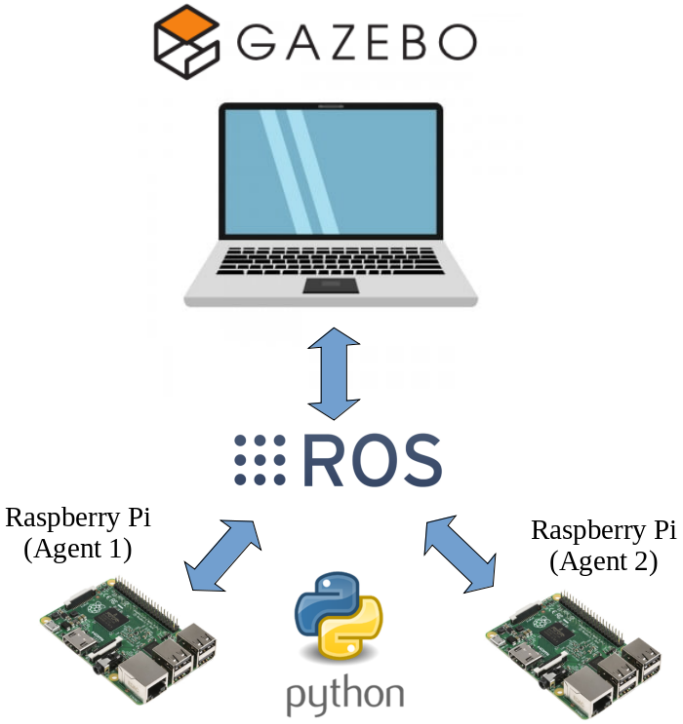

FIGURE 4.17: Workflow for Raspberry Pi implementation.

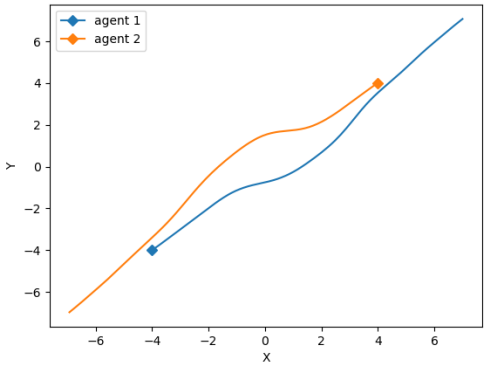

FIGURE 4.18: 2D visualisation of mission using Raspberry Pi.

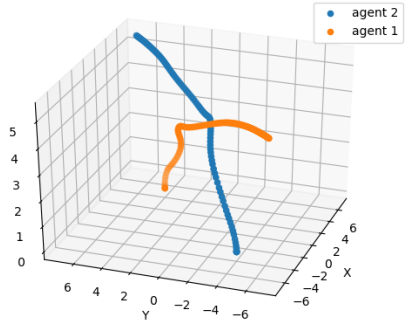

FIGURE 4.19: 3D visualisation of mission using Raspberry Pi.

Chapter 5

Conclusion

In this thesis, I presented a novel data-driven risk sensitive collision avoidance constraint formulation for safe multi-robot navigation in both distributed and decentralised settings. The proposed approach robustifies MPC solutions against errors in the predictions of surrounding objects in terms of distributional robustness guaranteed by the Wasserstein metric by leveraging data that is collected online during execution. CVaR-based risk constraints capture the proximity to the uncertain polyhedral obstacles and provides the ability for the user to dictate the risk-appetite of the robot. The performance of the proposed approach was examined via numerical simulations with multiple aerial robots, the influence of risk tolerance level and size of the ambiguity sets was illustrated in detail, and further validated on realistic Gazebo simulations. The computation time of the resulting MPC problem is sufficiently small to be deployed in practice. In future, I aim to explore other classes of ambiguity sets for this task that may reduce the computational burden while preserving the desired robustness properties.

The code for this work has been hosted on the Github on `https://github.com/anavsalkar/cvar_dist_mpc` repository. This work is currently under review for International Conference on Robotics and Automation (ICRA) 2023. This project also won the "Most Innovative Project" award in the Student Innovation Grant Program by AI & Robotics Park, IISc. Bangalore.

Bibliography

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36.

Arul, S. H. and Manocha, D. (2020). Dcad: Decentralized collision avoidance with dynamics constraints for agile quadrotor swarms. *IEEE Robotics and Automation Letters*, 5(2):1191–1198.

Arul, S. H. and Manocha, D. (2021). Swarmcco: Probabilistic reactive collision avoidance for quadrotor swarms under uncertainty. *IEEE Robotics and Automation Letters*, 6(2):2437–2444.

Batkovic, I., Rosolia, U., Zanon, M., and Falcone, P. (2020). A robust scenario mpc approach for uncertain multi-modal obstacles. *IEEE Control Systems Letters*, 5(3):947–952.

Berg, J. v. d., Guy, S. J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer.

Brüdigam, T., Olbrich, M., Wollherr, D., and Leibold, M. (2021). Stochastic model predictive control with a safety guarantee for automated driving. *IEEE Transactions on Intelligent Vehicles*.

Castillo-Lopez, M., Ludivig, P., Sajadi-Alamdari, S. A., Sanchez-Lopez, J. L., Olivares-Mendez, M. A., and Voos, H. (2020). A real-time approach for chance-constrained motion planning with dynamic obstacles. *IEEE Robotics and Automation Letters*, 5(2):3620–3625.

Cheng, H., Zhu, Q., Liu, Z., Xu, T., and Lin, L. (2017). Decentralized navigation of multiple agents based on orca and model predictive control. In *2017 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pages 3446–3451. IEEE.

Cong, S., Wang, W., Liang, J., Chen, L., and Cai, Y. (2021). An automatic vehicle avoidance control model for dangerous lane-changing behavior. *IEEE Transactions on Intelligent Transportation Systems*.

Dai, L., Hao, Y., Xie, H., Sun, Z., and Xia, Y. (2022). Distributed robust mpc for nonholonomic robots with obstacle and collision avoidance. *Control Theory and Technology*, 20(1):32–45.

de Groot, O., Brito, B., Ferranti, L., Gavrila, D., and Alonso-Mora, J. (2021). Scenario-based trajectory optimization in uncertain dynamic environments. *IEEE Robotics and Automation Letters*, 6(3):5389–5396.

Dixit, A., Ahmadi, M., and Burdick, J. W. (2022). Risk-averse receding horizon motion planning. *arXiv preprint arXiv:2204.09596*.

Firoozi, R., Ferranti, L., Zhang, X., Nejadnik, S., and Borrelli, F. (2020). A distributed multi-robot coordination algorithm for navigation in tight environments. *arXiv preprint arXiv:2006.11492*.

Furrer, F., Burri, M., Achtelik, M., and Siegwart, R. (2016). Rotors—a modular gazebo mav simulator framework. In *Robot operating system (ROS)*, pages 595–625. Springer.

Gao, Y., Jiang, F. J., Xie, L., and Johansson, K. H. (2021). Risk-aware optimal control for automated overtaking with safety guarantees. *IEEE Transactions on Control Systems Technology*.

Gopalakrishnan, B., Singh, A. K., Kaushik, M., Krishna, K. M., and Manocha, D. (2017). Prvo: Probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1089–1096. IEEE.

Hakobyan, A., Kim, G. C., and Yang, I. (2019). Risk-aware motion planning and control using cvar-constrained optimization. *IEEE Robotics and Automation letters*, 4(4):3924–3931.

Hakobyan, A. and Yang, I. (2021). Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk. *IEEE Transactions on Robotics*, 38(2):939–957.

Hota, A. R., Cherukuri, A., and Lygeros, J. (2019). Data-driven chance constrained optimization under wasserstein ambiguity sets. In *2019 American Control Conference (ACC)*, pages 1501–1506. IEEE.

Huang, S., Teo, R. S. H., and Tan, K. K. (2019). Collision avoidance of multi unmanned aerial vehicles: A review. *Annual Reviews in Control*, 48:147–164.

Kamel, M., Alonso-Mora, J., Siegwart, R., and Nieto, J. (2017). Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 236–243. IEEE.

Katriniok, A., Kojchev, S., Lefeber, E., and Nijmeijer, H. (2018). Distributed scenario model predictive control for driver aided intersection crossing. In *2018 European Control Conference (ECC)*, pages 1746–1752. IEEE.

Lucia, S., Tătulea-Codrean, A., Schoppmeyer, C., and Engell, S. (2017). Rapid development of modular and sustainable nonlinear model predictive control solutions. *Control Engineering Practice*, 60:51–62.

Luis, C. E., Vukosavljev, M., and Schoellig, A. P. (2020). Online trajectory generation with distributed model predictive control for multi-robot motion planning. *IEEE Robotics and Automation Letters*, 5(2):604–611.

Park, J. and Kim, H. J. (2020). Online trajectory planning for multiple quadrotors in dynamic environments using relative safe flight corridor. *IEEE Robotics and Automation Letters*, 6(2):659–666.

Sabatino, F. (2015). Quadrotor control: modeling, nonlinearcontrol design, and simulation.

Soloperto, R., Köhler, J., Allgöwer, F., and Müller, M. A. (2019). Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control. In *2019 18th European Control Conference (ECC)*, pages 811–817. IEEE.

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57.

Zhang, X., Liniger, A., and Borrelli, F. (2020). Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29(3):972–983.

Zhang, X., Ma, J., Cheng, Z., Huang, S., and Lee, T. H. (2021). Receding horizon motion planning for multi-agent systems: A velocity obstacle based probabilistic method. *arXiv preprint arXiv:2103.12968*.

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

Zhu, H., Claramunt, F. M., Brito, B., and Alonso-Mora, J. (2021). Learning interaction-aware trajectory predictions for decentralized multi-robot motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 6(2):2256–2263.