COLLISION AVOIDANCE ASSISTANCE IN UAV TELEOPERATION

COLLISION AVOIDANCE ASSISTANCE IN UAV TELEOPERATION

BY

SAHAND GHAFFARI, M.Sc., B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2021)	McMaster University
(Electrical & Computer Engineering)	Hamilton, Ontario, Canada

TITLE:	Collision Avoidance Assistance in UAV Teleoperation
AUTHOR:	Sahand Ghaffari M.Sc. (Mechanical Engineering), B.Sc. (Mechanical Engineering), K. N. Toosi University of Technology, Tehran, Iran
SUPERVISOR:	Prof. Shahin Sirouspour
NUMBER OF PAGES:	xxiii, 169

To my beloved wife Atrin and my dearest parents Nasrin and Touraj

Lay Abstract

Unmanned aerial vehicles (UAVs) are highly maneuverable and agile, and can access spaces that would normally be inaccessible by other means. Their applications often involve complex task scenarios performed in unstructured environments with uncertainties and time constraints that make full autonomous operation impractical and/or ineffective. This thesis presents a novel shared control strategy for human-in-the-loop teleoperation of UAVs. In this strategy, the operator uses a human-to-machine interface to move the UAV in its task-space while an automated collision avoidance algorithm helps prevent collisions with obstacles. The proposed collision avoidance algorithm incorporates UAV operational constraints. Furthermore, by predicting the future trajectory of the UAV, it would proactively prevent the operator from commanding it into a state where collisions with obstacles would become unavoidable. The collision avoidance assistance algorithm is further extended to guarantee collision avoidance in the presence of uncertainty. All collision avoidance strategies have been successfully implemented and evaluated in an indoor laboratory setting.

Abstract

Unmanned aerial vehicles (UAVs) have found an increasing number of applications in recent years. However, the complexity of the task environment and operational requirements in many of these applications render fully autonomous operation rather impractical. This thesis presents a novel shared control strategy for human-in-the-loop teleoperation of UAVs. It integrates user direct teleoperation of the UAV with automatic collision avoidance assistance. In this strategy, the operator utilizes a human-machine-interface (HMI) to provide linear acceleration commands for the UAV in order to navigate it in the task environment. Simultaneously, a collision avoidance assistance algorithm modifies the operator's commands to help avoid potential collisions with obstacles in the environment. These corrective commands are obtained by formulating an optimization problem over a rolling control horizon and solving it in real time, in the so-called model predictive control (MPC) framework. In the optimization formulation, obstacles represent restricted space that must be avoided by the UAV. The obstacle-free space manifests as a set of constraints on the states of the UAV. These obstacle-related constraints are generally non-convex in their original form, which can render the entire optimization problem non-convex. The obstacle free-space may be approximated by a convex region to avoid challenges associated with solving non-convex optimizations. Many of the contributions of this work relate

to obtaining such convex approximations that are not overly conservative to unnecessarily inhibit the UAV ability to move in its task environment.

Reachability analysis provides a powerful tool for identifying obstacles with the chance of collision, and creating approximate safe convex regions that are not too conservative. In this thesis, two new methods based on reachability analysis are proposed to generate such approximate convex obstacle-free space for the use in collision avoidance MPC. The first method uses backward reachability analysis to detect a subset of obstacles with chance of collision with the UAV in the MPC time horizon. Following the detection of these obstacles, the SVM algorithm is employed to construct a safe polyhedral convex region around the UAV. The second method improves on the first one by generating the approximate convex obstacle-free region based on forward reachability analysis. At each time step in the MPC horizon, separating hyperplanes between the UAV and obstacles are found that maximize the volume of the intersection of the UAV reach set and half-space produced by the hyperplanes. The convex safe region is approximated by the intersection of these half-spaces.

The next contribution of the thesis focuses on accounting for uncertainty in the system in the development of the collision avoidance assistance algorithm. A novel ellipsoidal-based robust model predictive control (RMPC) for collision avoidance assistance in UAV teleoperation is presented. The main contribution here is the formation of the collision avoidance assistance in UAV under uncertainties as a convex optimization problem. Ellipsoidal approximation of reachable regions due to feasible inputs and uncertainties are derived using reachability analysis. A new convex generation method is used to approximate the obstacle-free space with a polyhedral. An inner polyhedral approximation of tightened constraints guaranteeing collision avoidance is obtained using geometrical relationship between the ellipsoidal reach set and polyhedral safe region.

The final contribution of this thesis guarantees recursive feasibility of the MPC-based collision avoidance assistance formulation. A fundamental problem in finite-time MPC is the lack of guaranteed recursive feasibility. This could place the UAV into a state where collision with obstacles becomes unavoidable. In this thesis, new MPC-based collision avoidance assistance algorithms with guaranteed recursive feasibility for the UAV teleoperation with/without disturbances are introduced. Terminal velocity constraints are added to the MPC formulation to guarantee its recursive feasibility. A novel ellipsoidal tube-based MPC is introduced that extends collision avoidance assistance with guaranteed recursive feasibility to UAV under disturbances. In this method, an ellipsoidal approximation of the robust positively invariant (RPI) set is derived using a new RPI set approximation based on ellipsoidal techniques. Polyhedral approximation of the obstacle-free space is derived using the SVM algorithm. Ab inner polyhedral approximation of the tightened constraints is obtained using geometrical relation between ellipsoidal RPI set and polyhedral safe region.

The methods proposed in this thesis are evaluated experimentally in an indoor environment using a fully-actuated UAV. The results demonstrate that the MPC-based collision assistance is highly effective in helping the operator navigate the task environment and avoid collisions with obstacles. This is in part due to the fact the methods introduced here generate a realistic convex approximation of the obstacle-free space. This allows the operator to teleoperate the UAV in the task environment without unnecessarily being hindered by a conservative approximation of this space.

Acknowledgements

I would like to express my gratitude toward my supervisor, Dr. Shahin Sirouspour, for his guidance, support, and encouragement throughout my Ph.D. studies. My appreciation to my committee members Dr. Gary Bone and Dr.Prashant Mhaskar for constructive feedback regarding my research. Finally, I would like to give a very special thanks to my family who made it possible for me to excel in my studies with endless love and support.

Contents

List of Figures

Notation, Definitions, Abbreviations, and Symbols

Notation

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Set of real n-vectors
$\mathbb{R}^{m\times n}$	Set of real $m \times n$ matrices
$\mathbb{N},\ \mathbb{N}^+$	Sets of non-negative and positive integers
$conv\mathbb{R}^n$	Closed and convex set in $\mathbb{R}^n$
$\mathbf{0}_{m\times n}$	$m \times n$ matrix with all element is zero
$\mathbf{1}_{m\times n}$	$m \times n$ matrix with all element is one
$\boldsymbol{I}_n$	$n \times n$ identity matrix
$\mathbb{H}$	Set of all quaternions
trX	Trace of matrix $\boldsymbol{X}$

$\|\|\boldsymbol{x}\|\|_2$	Euclidean norm of vector $\boldsymbol{x}$
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	The inner product of vectors $\boldsymbol{x},\ \boldsymbol{y} \in \mathbb{R}^n$, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^T \boldsymbol{y}$
$\langle \boldsymbol{A}, \boldsymbol{B} \rangle$	The inner product of matrices $\boldsymbol{A},\ \boldsymbol{B} \in \mathbb{R}^{n \times m}$, $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = tr(\boldsymbol{A}\boldsymbol{B}^T)$
$diag(a_1, a_2, ..., a_n)$	$n \times n$ diagonal matrix whose entries are $a_1, a_2, ..., a_n$
$\boldsymbol{x} \preceq \boldsymbol{y}$	Componentwise inequality between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$
$det\ \boldsymbol{A}$	The determinant of matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$

Definitions

Polyhedral Set Set of all points inside or on the boundary of the intersection of finite numbers of half-spaces. This set is convex and can be open or closed.

Polytopic Set A bounded polyhedral set is called a polytopic set. This set is closed and convex.

Abbreviations

HMI	Human machine interface
MPC	Model predictive control
BROD	Backward reachability-based obstacle detection
RMPC	Robust model predictive control

RPI	Robust positively invariant
SVM	Support vector machine

Symbols

$\boldsymbol{p}_{cg}$	Position of the UAV
$\boldsymbol{v}_{cg}$	Linear velocity of the UAV
$\boldsymbol{\omega}$	Angular velocity of the UAV
$\boldsymbol{q}$	Unit quaternion representing the attitude of the UAV
$\boldsymbol{g}$	Gravitational constant
$\boldsymbol{m}$	Mass of the UAV
$\boldsymbol{J}$	Moment of inertia of the UAV
$\boldsymbol{f}_b$	Force acting on the body frame of the UAV
$\boldsymbol{\tau}_b$	Torque acting on the body frame of the UAV
$\boldsymbol{R}$	Rotation Matrix
$\boldsymbol{a}_h$	Linear acceleration command provided by human operator
$\boldsymbol{a}_d$	Desired linear acceleration command
$\boldsymbol{a}_c$	Corrective linear acceleration command
$\boldsymbol{q}_d$	Unit quaternion representing the desired attitude of the UAV

$\boldsymbol{\omega}_d$	Desired angular velocity of the UAV
$\boldsymbol{x}$	Sates of the UAV
$\boldsymbol{x}_0, \bar{\boldsymbol{x}}_0, \boldsymbol{x}_{e_0}$	Initial states, nominal initial states, and initial states deviation of the UAV
$\boldsymbol{x}_{c_0}, \boldsymbol{x}_c$	Initial states and states of the obstacle
$\boldsymbol{G}(t, s)$	State transition matrix of linear time-invariant dynamic system at time t starting from time s
$\boldsymbol{\phi}(t, s)$	State transition matrix of linear dynamic system at time t starting from time s
$p_d(t + \tau_n)$	Desired near-viewpoint in human control model
$p_d(t + \tau_f)$	Desired far-viewpoint in human control model
G_m	Mapping transfer function
G_p	Plant transfer function
G_n	Transfer function modelling the operator's response to the near-viewpoint
G_f	Transfer function modelling the operator's response to the far-viewpoint
G_e	Transfer function modelling the operator's response to the tracking error

G_{nms}	Transfer function modelling the neuromuscular dynamics of the human operator
$\boldsymbol{u}, \boldsymbol{u}_h, \bar{\boldsymbol{u}}$	Input command, operator's input command, and nominal input command
$\boldsymbol{u}_c$	Input to the obstacle
$\boldsymbol{\omega}$	Disturbance input acting on the UAV
$\mathbb{U}$	Input set
$\mathbb{X}_0$	Initial state set
$\mathbb{X}_f$	Target set
$\mathbb{X}$	Forward reach set
$\bar{\mathbb{X}}$	Forward reach tube
$\mathbb{W}$	Backward reach set
$\bar{\mathbb{W}}$	Backward reach tube
$\mathbb{X}_e$	State deviation set
$\mathbb{X}_s$	Safe state set
$\mathbb{X}_u$	Undesirable state set
$\mathbb{S}$	Robust positively invariant set
$\bar{\mathbb{S}}$	Minimum robust positively invariant set

$\mathbb{X}_{c_0}$	Initial state set of the obstacle
$\mathbb{U}_c$	Input set of the obstacle
$\boldsymbol{A}_u, \boldsymbol{b}_u$	Parameters of the polytopic input set
$\boldsymbol{A}_v, \boldsymbol{b}_v$	Parameters of the polytopic velocity set
$\boldsymbol{A}_p, \boldsymbol{b}_p$	Parameters of the polyhedral safe convex region
$\boldsymbol{A}_x, \boldsymbol{b}_x$	Parameters of the polyhedral state constraint
$\boldsymbol{A}_o, \boldsymbol{b}_o$	Parameters of the polytopic obstacle
$\boldsymbol{A}_{x_u}, \boldsymbol{b}_{x_u}$	Parameters of the polytopic undesirable set
$\boldsymbol{q}_u, \boldsymbol{Q}_u$	Parameters of the ellipsoidal input set
$\boldsymbol{q}_\omega, \boldsymbol{Q}_\omega$	Parameters of the ellipsoidal disturbance set
$\boldsymbol{q}_f, \boldsymbol{Q}_f$	Parameters of the ellipsoidal target set (undesirable set)
$\boldsymbol{q}_w, \boldsymbol{Q}_w$	Parameters of the ellipsoidal backward reach set
$\boldsymbol{q}_{p_f}, \boldsymbol{Q}_{p_f}$	Parameters of the minimum volume ellipsoid covering the polytopic obstacle
$\boldsymbol{q}_{v_f}, \boldsymbol{Q}_{v_f}$	Parameters of the minimum volume ellipsoid covering the polytopic velocity set
$\boldsymbol{q}_{x_0}, \boldsymbol{Q}_{x_0}$	Parameters of the ellipsoidal initial state set
$\boldsymbol{q}_x, \boldsymbol{Q}_x$	Parameters of the ellipsoidal forward reach set

$\boldsymbol{q}_{x_e}, \boldsymbol{Q}_{x_e}$	Parameters of the ellipsoidal state deviation reach set
$\boldsymbol{q}_{x_c}, \boldsymbol{Q}_{x_c}$	Parameters of the ellipsoidal reach set of obstacle
$\boldsymbol{q}_{u_c}, \boldsymbol{Q}_{u_c}$	Parameters of the ellipsoidal input set of obstacle
$\boldsymbol{q}_{c_0}, \boldsymbol{Q}_{c_0}$	Parameters of the ellipsoidal initial sate set of obstacle
$\boldsymbol{\alpha}, \boldsymbol{\eta}$	Parameters of the separating hyperplane
$\boldsymbol{v}_o^j$	j-th vertex of the polytopic obstacle

Chapter 1

Introduction

1.1 Motivation

Unmanned aerial vehicles (UAVs) are highly maneuverable and agile, and can access spaces that would normally be inaccessible by other means. These unique capabilities have given rise to their growing applications in agriculture, defense and law enforcement, inspection and maintenance of infrastructure, mining, search and rescue, surveillance, fire monitoring, and transportation [50, 68, 74, 98]. These applications often involve complex task scenarios performed in unstructured environments with uncertainties and time constraints that make full autonomous operation impractical and/or ineffective. In such circumstances, a shared control strategy provides an effective collaborative framework for human-robot co-execution of the task [47]. The operator can rely on his/her cognitive abilities to assess the task environment, make high-level decisions, and command the robot to execute the task accordingly, while being assisted by automatic control algorithms.

In devising shared control strategies, care must be taken not to overburden the operator with secondary tasks that could distract from primary objectives. Increased workload combined with poor situational awareness especially when executing complicated manoeuvres can lead to operator errors [79, 94]. Avoiding collisions in complex environments can be particularly challenging for a human operator and can lead to fatigue and poor task performance.

Assigning the collision avoidance task to the UAV while entrusting tasks with higher-level cognitive process requirements to the human operator offers great advantages over a fully autonomous system. However, such shared control strategy would introduce new challenges both in theoretical development and in practical implementation. On the UAV side, the main challenge is to devise a collision avoidance algorithm to assist the user to operate the UAV in an unstructured and complex environment in real-time, while incorporating the UAV operational constraints. On the operator side, a suitable platform must be designed for human-UAV interaction to allow the operator effectively command the UAV and achieve the task objectives, while minimizing the operator's workload.

The subject of collision avoidance in UAVs has found increasing interest in recent years. However, there are still significant gaps in research on collision avoidance in unstructured and complex environments with uncertainties in the UAV and its environment. This thesis focuses on addressing some of the shortcomings in the existing research. It seeks to design and implement novel collision avoidance assistance algorithms that would help an operator to safely and and reliably teleoperate a UAV in its task environment.

1.2 Problem Statement

In this thesis, a semi-autonomous shared control strategy is introduced to navigate the UAV in a complex and unstructured environment. In this strategy, the operator tele-operates the UAV by providing linear motion acceleration commands, while a collision avoidance algorithm modifies these commands to ensure collision-free operation. There has been a great deal of research in developing automated collision avoidance algorithms, particularly in the field of mobile robotics. The reader is referred to a number of reviews on the subject matter in [1, 48, 75]. This thesis proposes optimization-based collision avoidance strategies, where collision avoidance is formulated as a constrained optimization problem. This approach can easily incorporate UAV-related operational constraints in collision avoidance. The optimization problems are formulated over a rolling control horizon, in the so-called model predictive control (MPC) framework. By predicting the future trajectory of the vehicle, the collision assistance algorithm would be able to proactively prevent the operator from commanding the UAV into a state where collisions with obstacles would become unavoidable.

One of the main challenges in optimization-based approaches to collision avoidance is related to the formulation of obstacle-free space as a convex constraint. Convexity of constraints and objective are highly desirable as they would lead to convex optimization problem formulations that can be effectively solved to their global optimal solution with standardized routines. However, obstacle-related constraints are generally non-convex in their original form. A natural idea is to use non-convex programming methods to solve the resulting problem. However, in non-convex optimizations, finding and certifying the global optimum point is impossible or computationally prohibitive [17], making it impractical for real-time applications.

An alternative approach to overcome the non-convexity of obstacle-related constraints is to approximate the obstacle-free space with a convex region. However, existing studies on this approach usually consider all obstacles around the robot in safe convex region generation rather than only focusing on obstacles with the chance of collision. This can lead to an over-constrained approximation of the obstacle-free space, limiting the robot motion space. Furthermore, the obstacle-free space is approximated by a convex region without considering the reachable regions of the robot. The intersection among the resulting convex region and the robot reachable region can be empty or small, significantly inhibiting the robot ability to move in space.

Uncertainties in robot motion and the task environment are unavoidable in real-world applications. To ensure collision-free teleoperation of the UAV in an obstacle-rich environment, uncertainties should be considered in designing the collision avoidance system. Reachability analysis can provide a powerful tool to calculate the reachable states of the UAV due to uncertainties, which can then be used in formulating the collision avoidance optimization problem.

The MPC optimization problem must be repeatedly solved solved over a rolling horizon to find corrective commands that would help avoid collisions. A fundamental problem in finite-time MPC is the lack of guaranteed recursive feasibility. This means that the feasibility of the MPC at the initial time does not guarantee its feasibility at future iterations. This could place the UAV into a state where collision with obstacles becomes unavoidable. Tube-based MPC [54] provides an effective solution to guarantee recursive feasibility in UAV under uncertainties. Reachability analysis can provide a powerful tool to calculate the reachable tube.

An optimization-based collision avoidance algorithm that incorporates the reachability analysis benefits from the flexibility of optimization-based techniques in UAV teleoperation. It would address some of the limitations of optimization-based methods in relation to characterizing the obstacle-free space as a convex region.

1.3 Thesis Contributions

This thesis makes a number of important contributions in the area of human-in-the-loop control of UAVs. It presents a comprehensive theoretical treatment of the problem, including teleoperation strategy, dynamic modelling, and collision avoidance algorithm design. The theoretical developments are evaluated experimentally.

In this work, the operator tele-operates the UAV by providing acceleration commands. A model predictive collision avoidance algorithm modifies these commands to ensure collision-free operation of the UAV. The collision avoidance problem is formulated as a convex constrained optimization problem over a rolling control horizon. The objective is to minimize interference with the operator's commands, while keeping the UAV in the obstacle-free space. The reachability analysis is utilized to approximate the obstacle-free space as convex set in the MPC optimization problem.

The reachability analysis of dynamical system can be utilized to generate a convex region approximation of the obstacle-free space in collision avoidance. This approximation is needed for use as constraints in the formulation of the collision avoidance optimization problem. The so-called forward reachability analysis determines the set of system reachable states during the time of interest. The backward reachability analysis produces the set of system initial states that can lead to the undesirable states over the time of interest. Calculating the exact reach set of the system at each time of interest can be computationally prohibitive, making it unattractive for real-time applications, particularly in high-dimensional systems. In linear dynamical systems, the reachable set can be approximated by predefined geometrical shapes. Reachability analysis based on ellipsoidal techniques is popular for such systems. Ellipsoids can effectively approximate geometrical shapes and have simple mathematical formulation.

This thesis employs the reachability analysis in two new methods for approximating the obstacle-free space with a convex region. The first method is a novel backward reachability-based obstacle detection algorithm that identify obstacles that could potentially collide with the UAV within a rolling control time horizon. The computations are significantly reduced by using an ellipsoidal technique and finding analytical solutions to all relevant differential equations. Following the identification of obstacles with the chance of collision, the SVM algorithm [41] is employed to construct a safe polyhedral convex region around the UAV that would keep it away from the obstacles with the chance of collision.

The second method is a new convex region generation algorithm based on reachability analysis. In this algorithm, first an approximate reachable region of the system at each time instance of interest is obtained. Then, a safe-half space with maximum reachable region is created by finding the separating hyperplane between the unsafe undesirable set intersecting with the reachable region and a so-called target point of the system at each time instance of interest.

Another contribution of this thesis is to account for uncertainties in the UAV and its environment in creating the safe region and associated collision avoidance assistance algorithm. To this end, the reachable states of the UAV and obstacles subject to uncertainties are computed at each time instance of interest. A polyhedral safe convex region is created using the algorithm developed earlier. Collisions with obstacles are avoided by guaranteeing that the UAV would not collide with obstacles under any feasible realization of the uncertainties.

The final contribution of this thesis is in guaranteeing recursive feasibility in MPC-based collision avoidance assistance in the UAV with/without uncertainties. Recursive feasibility is guaranteed in UAV by adding terminal velocity constraints. A novel ellipsoidal tube-based MPC is presented to extend the collision avoidance assistance with recursive feasibility for UAV under uncertainties. In this method, an ellipsoidal approximation of the RPI set is calculated. Polyhedral approximation of the obstacle-free space is derived using the SVM algorithm. The inner polyhedral approximation of the tightened constraints is obtained using geometrical relation between the ellipsoidal RPI set and polyhedral safe region.

In summary, the main contributions of this thesis are:

- A shared control strategy for human-in-the-loop teleoperation of UAVs with automated collision avoidance assistance. The collision avoidance is formulated as a convex optimization problem that can be solved in real-time to find a globally optimal solution.

- An obstacle detection algorithm based on backward reachability analysis. The reachability analysis helps identify obstacles with chance of collision with the UAV over the control time horizon. These are then considered in generating a convex approximate obstacle-free space. This approach generally yields a less conservative approximation of the obstacle-free space compared to a case where all obstacles are considered.

- A convex region generation method based on reachability analysis. This algorithm attempts to maximize the intersection of the reachable region of the UAV and the approximated convex obstacle-free space. This generally yields a less restrictive approximation of the obstacle-free space allowing the UAV to more freely move in its environment.

- The proposed convex region generation algorithm is general and not limited to three dimensional spaces. It can be used to create the convex region for spaces with arbitrary dimension. Furthermore, the algorithm low computational complexity makes it suitable for real-time applications.

- Ellipsoidal-based RMPC for collision avoidance in UAV subject to uncertainty. The proposed algorithm addresses the collision avoidance problem when there are uncertainties in the UAV and obstacles models.

- New MPC-based collision avoidance assistance algorithms with guaranteed recursive feasibility for UAV teleoperation with/without uncertainties.

- Experimental evaluation of the proposed shared control strategy with different collision avoidance assistance algorithms.

1.4 Thesis Organization

The remainder of this thesis is organized as follows. Pertinent literature in the UAV teleoperation, collision avoidance, and reachability analysis is reviewed in Chapter 2. A new optimization-based shared control strategy with automated collision avoidance is introduced in Chapter 3. A novel backward reachability-based obstacle detection is presented and implemented in Chapter 4. A new convex region generation based on reachability analysis is proposed in Chapter 5. The effectiveness of this algorithm is demonstrated experimentally. In Chapter 6, collision avoidance assistance is extended to account for uncertainties in the UAV and its environment. Collision avoidance assistance with guaranteed recursive feasibility is introduced in Chapter 7. This thesis is concluded in Chapter 8 where some suggestions for future research are also provided.

1.5 Publication

- S. Ghaffari and S. Sirouspour, Convex Optimization for Collision Avoidance Assistance in Teleoperation of Mobile Robots with Linear Dynamics, Submitted to IEEE/ASME Transactions on Mechatronics, 2021
- S. Ghaffari and S. Sirouspour, Collision Avoidance Assistance in UAV Teleoperation using Model Predictive Control and Reachability Analysis, Submitted to IEEE Transactions on Systems, Man and Cybernetics, 2021 (revised).
- S. Ghaffari and S. Sirouspour, Collision Avoidance Assistance in UAV Teleoperaiton under Uncertainty using Ellipsoidal Tube based MPC, Submitted to IEEE International Conference on Robotics and Automation (ICRA), 2021.

Chapter 2

Literature Review

The literature review in this thesis is organized under the three categories of human manual control, collision avoidance, and reachability analysis. The first group of work reviewed concerns modelling of the manual control behaviour of a human operator. These theories behind these modelling methods inform decisions made about the mapping of the human commands to the UAV motion in order to facilitate its teleoperation.

Next, some of the most relevant work in the field of mobile robotics as it pertains to obstacle and collision avoidance are surveyed. In this context, various existing formulations of optimization-based collision avoidance including convex and non-convex formulations of the problem are explored. Methods that approximate the obstacle-free space with convex region for use in convex optimization are briefly reviewed and their advantages and disadvantages are discussed.

The thesis main contributions revolve around the use of reachability analysis for dynamical systems in producing more realistic and less restrictive safe convex region approximation of the obstacle-free space. The review here covers different formulations of the reachability analysis and explores the pros and cons of each of these formulations for real-time collision avoidance.

2.1 Human Manual Control

Studying the modelling of the human manual control behavior can yield valuable insight into the design of effective human-machine interaction strategies. A significant portion of the research in this area comes from studying the human control models in piloted aircraft systems [95].

The control behavior of the human operator can be categorized into the three stages of compensatory, pursuit, and precognitive control [49]. In the compensatory stage, the human controller only acts on the error between the reference and system output [65]. Most of the work in this area is based on the research of McRuer and his team [49, 65, 66]. They extensively investigated the human pilot control behavior and proposed the so-called *crossover* model. This model states that the operator controller adapts to the plant dynamics by providing lead and lag equalization such that the combined operator-plant is rendered to an integrator with time delay.

In the pursuit stage, the human controller uses the combination of at least two of the following strategies: a) a feedforward response of the target b) a compensatory feedback response of the error, and c) a feedback response of the system output [70]. In [32–34], the behavior of the feedforward term of the human controller in pursuit tracking is investigated and modelled by a transfer function. In [91, 92], it is shown that that previewing future task information to the operator can improve him/her performance. The work in [93] concludes that increasing the preview look-ahead time enhances the human controller performance.

In the precognitive stage, the human operator is assumed to have complete knowledge of the target and may develop a fully open-loop control strategy. A systematic understanding for this control stage is still lacking [70].

2.2 Collision Avoidance

Due to their agility and maneuverability, UAVs have found increasingly applications in recent years. Having the ability to reliably avoid collisions is crucial to achieving autonomy in UAV operations. There has been a great deal of research in developing automated collision avoidance algorithms particularly in the field of mobile robotics. The reader is referred to a number of reviews on the subject matter in [1, 48, 97]. In general, collision avoidance algorithms can be divided into reactive or deliberative planning [97]. In deliberative planning, a map of the environment is generated and an optimal collision-free path is planned. The need for an accurate and updated map of the environment makes this approach computationally expensive and rather unsuitable for use in dynamic environments. In reactive methods, the robot reacts to the information from local sensors. These approaches are generally computationally efficient but they can lead to a local minimum and may need another navigation techniques to resolve the issue.

Existing collision avoidance algorithms can be categorized into one of geometric methods, potential-field methods, or optimization-based methods [48, 97]. Geometric methods utilize geometrical attributes and velocity of the UAV and obstacles to ensure guaranteed separating distance between the UAV and the obstacles. In [85], a collision avoidance algorithm based on line-of-sight and relative velocity vectors is presented.

In this algorithm, safe directions for preventing collisions are calculated. A collision avoidance algorithm based on geometrical relations is introduced in [42]. In this paper, obstacles with a high risk of collision are identified, boundary spheres/cylinders are generated for these obstacles, and the fastest direction to avoid the obstacles is determined. In [57], an algorithm is introduced for collision avoidance algorithm in fixed-wing UAVs. Collisions are avoided based on obstacle avoidance geometry and critical avoidance start time.

In potential field methods, attractive and repulsive force fields push the robot towards its intended target and away from obstacles in the environment. In [89], a potential field method for obstacle avoidance and path planning of mobile robots in a static environment is presented. In addition to the repulsive forces of the obstacles, the method introduces a gravity chain between the starting position and target points, which would act as a guide for the robot. In [88], an optimized potential field for multi-UAV operation in 3D space is presented. By considering the interaction between the UAVs, the classical potential field method is extended for multi-UAV scenarios. The enhanced curl-free vector field algorithm is proposed in [25]. The curl-free vector field is used instead of the conventional repulsive potential field to avoid the obstacles. The direction of the vector field is determined based on the velocity vectors of obstacles.

2.2.1 Optimization-based Collision Avoidance

Optimization-based methods cast the collision avoidance problem as an optimization with a relvant objective function and the robot operational constraints. The optimal or near-optimal solution to such optimization problem is then sought. The existing literature covers a broad range of optimization problem formulations from convex optimization to heuristic methods [14]. In [16], a predictive controller for decentralized cooperative control of UAVs is presented. In this work, the best set of future commands is selected such that there is no intersection between the UAV future trajectory and obstacles. A stochastic collision avoidance method for a tele-operated UAV is presented in [11]. In this algorithm, the future trajectory of the UAV based on the operator's command is calculated. The operator's commands are modified based on the probability of collision with obstacles. A particle swarm optimization-based path planning method for operation in unknown environments is proposed in [14]. The optimal path is determined based on terrain traversability. Heuristic methods are computationally expensive making them undesirable for real-time applications. These algorithms are usually used as deliberative planning in conjunction with rapid reactive collision avoidance algorithms.

Obstacle-free space in its original form generally translates into non-convex constraints for the optimization problem. Naturally, many of the existing formulations of collision avoidance are non-convex optimizations. In [8, 77, 82], a sequential convex program is used to solve the collision avoidance optimization problem. This method sequentially approximates the non-convex problem by local convex problems, which can then be solved by standardized convex solvers [8]. Finding and certifying the global optimum point for non-convex optimizations is impossible and/or computationally prohibitive [17], making them unsuitable for use in real-time applications.

Alternatively, the non-convex obstacle-free space can be approximated by a convex region to be used as constraints in the formulation of convex optimization. A common approach is based on cell decomposition [15, 20, 39] where the configuration space is discretized to smaller cells and the obstacle-free space is approximated by a combination of free cells. In [20], the obstacle-free space is approximated using the octree-based environment. Topomap is introduced in [15]. It derives a topological map of the environment and approximates the obstacle-free space by growing and merging voxel clusters. In [39], an occupancy grid map is used to discretize the environment and a polytopic convex region is generated from the cluster of free voxels. These methods preprocess data to build and discretize the map; modifying the map is usually time demanding. Therefore, adoption of such algorithms for large spaces in real-time can be challenging.

When obstacles are modeled as polytopes, keeping the robot outside of these polytopic obstacles is equivalent keeping the robot at least outside one of the faces of each obstacle. The collision avoidance problem can be formulated as a mixed integer programming [45, 67, 81]. An integer variable is needed for each obstacle face making the problem quickly intractable when more than a few obstacles are present in the environment [28].

Another approach is to create obstacle-free convex region around selected points in the environment [27, 46, 58, 99]. The IRIS algorithm is introduced in [27] to approximate obstacle-free space by polytopic and ellipsoidal regions. Given an initial point, obstacle-free space is obtained by solving quadratic and semidefinite programs iteratively. A safe flight corridor (SFC) to approximate the obstacle-free space by series of polytopes is presented in [58]. Similar to IRIS, the SFC algorithm uses ellipsoidal and polytopic regions to approximate the obstacle-free space, but it is computationally less demanding. In [99], the obstacle-free space is approximated based on modified star convex polytope. In [46], a safe convex region is approximated by the intersection of half-spaces created by the support vector machine (SVM). In these studies, instead of just considering obstacles with the chance of collision, all obstacles around the robot are considered in safe convex region generation. This can lead to an over-constrained approximation of obstacle-free space limiting the robot motion. Furthermore, the obstacle-free space is approximated by a convex region without considering the reachable regions of the robot. The main drawback of such approach is the possibility of small or empty intersection region between the generated convex region and reachable region leading to a significant limitation in robot motion.

While significant progress has been made in convex optimization-based collision avoidance, most existing work ignores uncertainties in the system. In real-world scenarios, uncertainties are unavoidable and can significantly degrade the performance of collision avoidance algorithm. In the presence of uncertainties, the optimization problem may become infeasible resulting in collisions. Robust model predictive controllers (RMPC) have been developed to deal with uncertainties predictive control of dynamical systems. Min-Max MPC is a popular approach in this category [12, 31, 83]. It finds a conservative solution by considering the worst-case scenario for the uncertainties. The method is ill-suited for use with unstable dynamic systems [80] due to fast-growing uncertainty in the system states, which can lead to infeasibility of optimization-based collision avoidance.

An alternative method to address uncertainty is tube-based MPC introduced in [54] for a linear dynamic system under bounded disturbances. By generating control laws instead of control actions, this method separates the robustness problem from the MPC problem. In [60], tube-based MPC is used in an autopilot system to handle atmospheric disturbances for fixed-wing UAVs. A tube-based MPC is designed in [19] for time-varying attitude tracking of a rigid body spacecraft with additive disturbances. In [44], tube-based MPC is utilized in an active safety controller to deal with disturbance and modelling errors. Tube-based MPC involves complex set-based operations including Minkowski sum and Minkowski difference to obtain the invariant sets, exact reach sets, and tighten constraints. These operations are computationally prohibitive [80] making them unsuitable for real-time applications. To overcome this issue, the invariant and reach sets can be approximated by predefined geometrical shapes which would simplify the set operations by geometrical relations between these geometrical shapes.

2.3 Reachability Analysis

Reachability analysis computes the reachable states of the system at an arbitrary time by knowing the set of initial states and feasible inputs. This approach provides a powerful tool in the safety analysis of dynamical systems. There has been a great deal of research inreachability analysis of dynamical systems. The reader is referred to reviews on the subject matter in [4, 21].

Reachability analysis using the Hamilton-Jacobi (HJ) partial differential equation is one of the well-known methods in this area. In [62, 69], game theory-based HJ formulations of reachability analysis for the safety-critical situation are investigated. These formulations incorporate disturbance effects and are applicable to general non-linear dynamical systems. However, their high computational cost makes them unattractive for real-time applications, particularly for high-dimensional systems [21].

An alternative approach is based on set propagation techniques. In this approach, the reach set is calculated by the propagation of the initial set considering the feasible system inputs. This approach can be interpreted as an extension of the differential equations of the system dynamics to set-value initial condition and input. Although the set propagation approach can be used in reachability analysis of nonlinear systems [2, 7, 22, 78, 84], it is more popular for use in linear systems. In such systems, starting from convex sets of initial states and inputs, the approximated reach set in future times remain convex [4]. Due to the computational efficiency and geometrical simplicity of a convex set, the set propagation approach provides a powerful tool in the reachability analysis of linear dynamical systems. In these systems, the reach set can be approximated by different geometrical shapes including ellipsoids ([23, 52]), polytopes ([6, 26]), zonotopes ([3, 40]), and support functions ([38, 56]).

Reachability analysis based on ellipsoidal techniques is one of the well-known approaches for approximating the reach set in the linear dynamic system due to its efficiency in approximating more complex geometrical shapes and simplicity of its equation. In [23, 71], parameters of external ellipsoid covering the reachable region of the system based on different criteria are derived using a differential equation. In [52, 53], the concept of tight ellipsoids are introduced and the reach set is approximated via the union of internal tight ellipsoids or intersection of external tight ellipsoids. In [72, 73], minimum volume ellipsoid covering the reach set is chosen as the design criterion, and explicit formulas for parameters of the ellipsoid are derived.

2.4 Summary

In this chapter, first, a number of different models of the human manual control behaviour were reviewed. These models inform decisions made about the mapping of human commands to the UAV motion in order to facilitate its teleoperation. Then, some of the most relevant work in the field of collision avoidance in mobile robots was studied. In this context, various formulations of MPC-based collision avoidance including convex and non-convex formulation were investigated. Moreover, RMPC methods that account for modelling uncertainty were studied. Finally, the most relevant methods in reachability analysis of the dynamical system with safety assurance application were briefly.

Chapter 3

Shared Control Strategy

In this chapter, a new shared control strategy for the human-in-the-loop control of UAVs is proposed. In this approach, the operator tele-operates the UAV by providing motion commands. A collision avoidance algorithm modifies these commands to ensure collision-free operation. The mapping between the operator's input and the UAV motion is designed based on the results of numerous studies on human manual control behaviour [33, 92, 93].

The collision avoidance problem is formulated as a convex constrained optimization problem over a rolling control horizon. The objective is to minimize interference with the operator's commands while keeping the UAV in an obstacle-free space. This flexible collision avoidance strategy can easily incorporate the UAV operational constraints. Furthermore, by predicting the future trajectory of the vehicle, it would proactively prevent the operator from commanding it into a state where collisions with obstacles would become unavoidable.

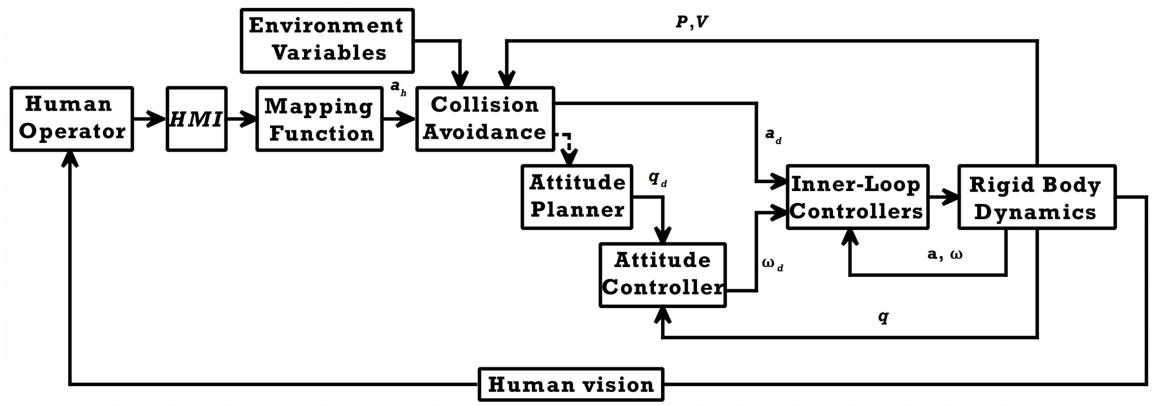

Figure 3.1: A schematic view of the shared control strategy.

3.1 Shared Control Architecture

A schematic view of the shared control strategy to navigate the UAV in an obstacle-rich environment is demonstrated in Fig. 3.1. The operator provides linear accelration commands to the UAV, $\boldsymbol{a}_h \in \mathbb{R}^3$. The collision avoidance algorithm modifies the operator's commands and produces the net desired acceleration commands $\boldsymbol{a}_d \in \mathbb{R}^3$. An attitude planner and controller determines the reference attitudes $\boldsymbol{q}_d \in \mathbb{H}$ (where $\mathbb{H}$ denotes the set of all quaternion) and angular velocities $\boldsymbol{\omega}_d \in \mathbb{R}^3$. The desired accelerations and angular velocities are sent to the inner loop controllers, which produce the actuator commands for the UAV. A mapping function is introduced between the HMI unit and the UAV acceleration commands to help the operator navigate the UAV effectively and to reduce the workload on him/her. To calculate the mapping function, first, the dynamic equations governing the translational motion of the UAV should be determined.

Remark. *The proposed shared control strategy (Fig. 3.1) can be implemented for fully-actuated or under-actuated UAVs. In a fully-actuated UAV, the translational* *motion is independent of angular motion and the attitude planer can be completely independent of the operator's commands. In this case, the dotted line can be removed in Fig. 3.1.*

3.2 Translational Dynamic Equations

Using the proposed shared-control strategy not only assists the operator to teleoperate the UAV without collision in an obstacle-rich environment but also facilitates the teleoperation task by simplifying the translational dynamics of the UAV using the hierarchical control structure. Fast inner-loop linear acceleration with gravity compensation and angular velocity controllers effectively render the dynamics of the UAV (A.28)-(A.33) into

$$\dot{\boldsymbol{p}}_{cg} = \boldsymbol{v}_{cg} \tag{3.1}$$

$$m\dot{\boldsymbol{v}}_{cg} = \boldsymbol{R}^T \boldsymbol{f}_b \tag{3.2}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{W}^T(\boldsymbol{q})\boldsymbol{\omega} \tag{3.3}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{\tau}_b \tag{3.4}$$

In this work, it is assumed that the UAV attitude is controlled within a hierarchical control architecture to decouple its translational and rotational dynamics. The attitude controller is tuned such that its closed-loop dynamics are fast enough to be ignored in the translational dynamics. This is reasonable as the translational dynamics would be subject to the operator teleoperation and automated collision avoidance commands. Using this assumption in (3.1) and (3.2),the translational motion of the

UAV is represented by the following linear dynamics:

$$\dot{\boldsymbol{p}}_{cg} = \boldsymbol{v}_{cg} \tag{3.5}$$

$$\dot{\boldsymbol{v}}_{cg} = \boldsymbol{a}_d \tag{3.6}$$

Here $\boldsymbol{a}_d = \boldsymbol{R}^T \boldsymbol{f}^b / m$ represents the commanded acceleration. It should be noted that in the case of an under-actuated UAV such as a quad-rotor, the attitude needs to be controlled to align the direction of the body-frame acceleration $\boldsymbol{a}_b$ with the desired input acceleration ($\boldsymbol{a}_d$). The attitude can be independently controlled in a fully-actuated UAV such as the one used in this thesis.

The linear translational dynamics in (3.5) and (3.6) can be written in a state-space form.

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{3.7}$$

where $\boldsymbol{x} = [\boldsymbol{p}_{cg}^T, \boldsymbol{v}_{cg}^T]^T$ represents the states of the system and $\boldsymbol{u} = \boldsymbol{a}_d$ is the desired acceleration. The matrices $\boldsymbol{A} \in \mathbb{R}^{6\times 6}$ and $\boldsymbol{B} \in \mathbb{R}^{6\times 3}$ are defined as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & \boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0}_{3\times 3} \\ \boldsymbol{I}_3 \end{bmatrix} \tag{3.8}$$

The solution to the dynamic system (3.7) is given by

$$\boldsymbol{x}(t) = \boldsymbol{G}(t, t_0)\boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{G}(t, s)\boldsymbol{B}\boldsymbol{u}(s)ds$$

where $\boldsymbol{x}_0 \in \mathbb{R}^6$ represents the initial states and $\boldsymbol{G}(t, s) \in \mathbb{R}^{6\times 6}$ is the state transition

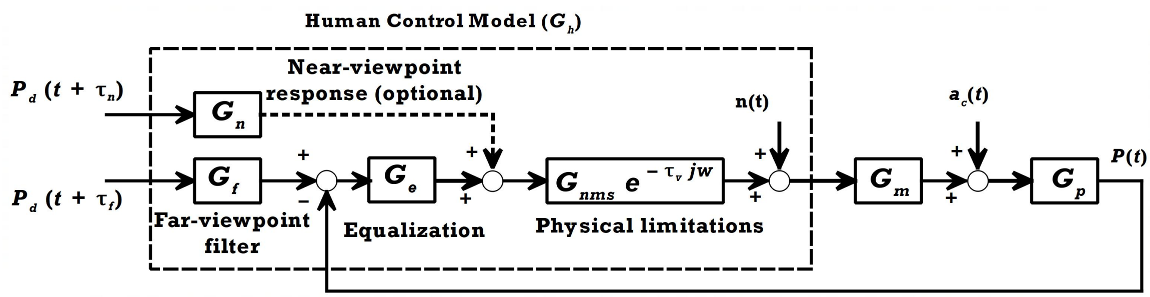

Figure 3.2: A schematic view of the human control model with the mapping function.

matrix

$$\boldsymbol{G}(t, s) = \begin{bmatrix} \boldsymbol{I}_3 & (t-s)\boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{I}_3 \end{bmatrix} \tag{3.9}$$

Using this hierarchical controller allows the operator to simply control a double-integrator plant dynamics. This should significantly facilitate the teleoperation task.

3.3 Mapping Function

A mapping function is introduced between the human machine interface (HMI) and the UAV acceleration commands to help the operator effectively navigate the UAV and reduce his/her workload. The design of the mapping function is informed by the existing research on human manual control behavior, the workspace constraints, and measurement limitations.

Fig. 3.2 shows a schematic view of the control system involving the human operator. This is built based on the models in [33, 92, 93] and the linear dynamics in (3.1) and (3.2). This model represents the pursuit and preview model of the human controller. In this model, the operator acts based on the desired near-viewpoint $p_d(t+\tau_n)$ and far-viewpoint $p_d(t+\tau_f)$. $G_m(j\omega)$ and $G_p(j\omega)$ represent the mapping function and the system dynamics, respectively. Moreover, the disturbance $a_c(t)$ represents the corrective acceleration command produced by the collision avoidance algorithm and any imperfection in inner-loop controllers. $n(t)$ is a colored noise which models non-linearities and noise in human control activities. The transfer function $G_n(j\omega)$ models the operator's response to near-viewpoint. The low-pass filter $G_f(j\omega)$ models the operator's response to the far-view-point. $G_e(j\omega)$ is an equalization term that models the operator's response to the tracking errors. This term models the human controller adaptation to the plant dynamics. $G_{nms}(j\omega)$ represents the neuromuscular dynamics of the human operator. The frequency response model of these transfer functions are given below [92, 93]

$$G_n(j\omega) = k_n j\omega, \qquad G_f(j\omega) = k_f \frac{1}{1+T_f j\omega}, \tag{3.10}$$

$$G_e(j\omega) = k_e \frac{1+T_1 j\omega}{1+T_2 j\omega} \qquad G_{nms}(j\omega) = \frac{\omega_{nms}^2}{(j\omega)^2 + 2\zeta_{nms}\omega_{nms} j\omega + \omega_{nms}^2} \tag{3.11}$$

where k_n is the near-viewpoint gain, k_f and T_f are the far-viewpoint gain and lag time constant; k_e is the error response gain, T_1 and T_2 are lead and lag equalization time constants, ω_{nms} and ζ_{nms} are natural frequency and damping ratio, respectively.

Using the crossover model [66] and considering that the combination of the equalization term, mapping function, and system dynamics behaves approximately as a single integrator around the crossover frequency, the mapping function can be selected as

$$G_m = k_m \frac{(j\omega)^2 + T_{m_1} j\omega}{j\omega + T_{m_2}} \tag{3.12}$$

where k_m is a gain, T_{m_1} and T_{m_2} are time constants, respectively.

Other mapping functions that do not adhere to this structure could force the operator to provide a second-order lead or lag equalization in order to stabilize the closed-loop system. This is a rather difficult or impossible task for the human operator [65]. In addition, G_m needs to be selected such that the human operator is not forced to provide lead compensation in the equalization terms as this could significantly increase the workload [66]. Therefore, the mapping function is simplified as

$$G_m = k_m((j\omega)^2 + T_{m_1} j\omega) \tag{3.13}$$

The HMI used in this research only measures the user input displacements. Numerical computation of the acceleration from these measurements can generate significant noise. To avoid the need for user acceleration measurement, the mapping function is revised as follows

$$G_m = k_m(j\omega) \tag{3.14}$$

This maps the user input velocity to acceleration command for the UAV.

3.4 Control Design

A number of low-level controllers are developed for the UAV as shown in Fig 3.1. The UAV attitude is independently controlled. PID controllers are used for the angular velocities and accelerations. A schematic view of the acceleration controller is shown in Fig. 3.3. The operator's acceleration commands $\boldsymbol{a}_d(t) \in \mathbb{R}^3$ transform to the desired body acceleration commands $\boldsymbol{a}_d^b(t) \in \mathbb{R}^3$) via the rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ and is sent to the PID controller. The acceleration dynamics are given by (A.30)

$$m\dot{\boldsymbol{a}}_{cg}^b = \boldsymbol{c}_1 m(\boldsymbol{a}_{cg}^b - \boldsymbol{R}\boldsymbol{g}) + \boldsymbol{c}_2\boldsymbol{u}_a \tag{3.15}$$

The acceleration controller is designed as

$$\boldsymbol{u}_a(t) = \boldsymbol{K}_a^p(\boldsymbol{a}_d^b(t) - \boldsymbol{a}^b(t)) + \boldsymbol{K}_a^d(\dot{\boldsymbol{a}}_d^b(t) - \dot{\boldsymbol{a}}^b(t)) + \boldsymbol{K}_a^i\int_0^t(\boldsymbol{a}_d^b(\tau) - \boldsymbol{a}^b(\tau))d\tau + \boldsymbol{u}_{comp}(t) \tag{3.16}$$

where $\boldsymbol{u}_{comp}(t) \in \mathbb{R}^3$ is the gravity compensation voltage obtained by (A.23) and (A.24). $\boldsymbol{K}_p^a$ is the proportional gain, $\boldsymbol{K}_d^a$ is the derivative gain, and $\boldsymbol{K}_i^a$ is the integral gain of the PID controller. Due to decoupling of the acceleration dynamics (3.15), the gains of the PID controller are selected as diagonal matrices.

$$\boldsymbol{K}_p^a = diag(k_{p_x}^a, k_{p_y}^a, k_{p_z}^a),\ \boldsymbol{K}_d^a = diag(k_{d_x}^a, k_{d_y}^a, k_{d_z}^a),\ \boldsymbol{K}_i^a = diag(k_{i_x}^a, k_{i_y}^a, k_{i_z}^a) \tag{3.17}$$

A schematic view of the cascade-style attitude and angular velocity controllers are shown in Fig. 3.15. The quaternion controller is designed as an outer-loop controller where the desired angular velocity $\boldsymbol{\omega}_d \in \mathbb{R}^3$ is computed from the quaternion error

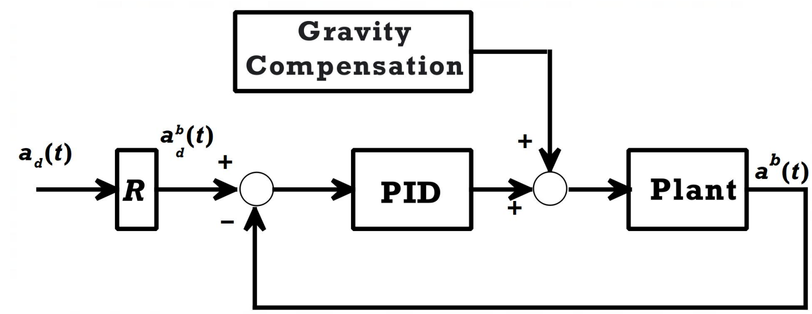

Figure 3.3: A schematic view of the acceleration controller.

$\boldsymbol{q}_e \in \mathbb{H}$. This desired angular velocity is sent to the inner-loop angular velocity controller to determine the desired voltage.

The UAV (omnicopter) attitude dynamics are given by (A.31)-(A.33).

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{W}^T(\boldsymbol{q})\boldsymbol{\omega} \tag{3.18}$$

$$\boldsymbol{J}\ddot{\boldsymbol{\omega}} = c_1\boldsymbol{J}\dot{\boldsymbol{\omega}} + c_2\boldsymbol{u}_\alpha \tag{3.19}$$

An inner-loop PID controller is used to control the angular velocity of the UAV

$$\boldsymbol{u}_\omega(t) = \boldsymbol{K}_\omega^p(\boldsymbol{\omega}_d(t) - \boldsymbol{\omega}(t)) + \boldsymbol{K}_\omega^d(\dot{\boldsymbol{\omega}}_d(t) - \dot{\boldsymbol{\omega}}(t)) + \boldsymbol{K}_\omega^i \int_0^t (\boldsymbol{\omega}_d(\tau) - \boldsymbol{\omega}(\tau))d\tau \tag{3.20}$$

where $\boldsymbol{K}_\omega^p, \boldsymbol{K}_\omega^d, \boldsymbol{K}^i p_\omega \in \mathbb{R}^{3\times 3}$ are diagonal gain matrices. $\boldsymbol{\omega}_d$ is determined by the outer-loop proportional controller in [35]

$$\boldsymbol{\omega}_d(t) = \boldsymbol{K}_q^p f(\boldsymbol{q}_e(t)) \tag{3.21}$$

where $\boldsymbol{K}_q^p \in \mathbb{R}^{3\times 3}$ is a control gain matrix and $f(q_e(t))$ is a function of the quaternion error

$$f(\boldsymbol{q_e}(t)) = \frac{2\arccos q_e^0(t)}{\sqrt{1-(q_e^0(t))^2}}\boldsymbol{q}_e^{1:3}(t) \tag{3.22}$$

Here $\boldsymbol{q}_e^i(t),\ i = 0, ..., 3$ are the elements of the quaternion error $\boldsymbol{q}_e(t)$

$$\boldsymbol{q}_e(t) = \boldsymbol{q}^{-1}(t) \circ \boldsymbol{q}_d(t) \tag{3.23}$$

where $\circ$ denotes the Hamilton product and $\boldsymbol{q}^{-1}(t)$ is the quaternion inverse [30]

$$\boldsymbol{q}^{-1}(t) = \frac{\tilde{\boldsymbol{q}}(t)}{||\boldsymbol{q}(t)||} \tag{3.24}$$

Moreover, $\tilde{\boldsymbol{q}}(t)$ and $||\boldsymbol{q}(t)||$ are calculated as

$$\tilde{\boldsymbol{q}}(t) = \begin{bmatrix} q_0(t) \\ \boldsymbol{q}_{1:3}(t) \end{bmatrix} \tag{3.25}$$

$$||\boldsymbol{q}(t)|| = \sqrt{q_0^2(t) + q_1^2(t) + q_2^2(t) + q_3^2(t)} \tag{3.26}$$

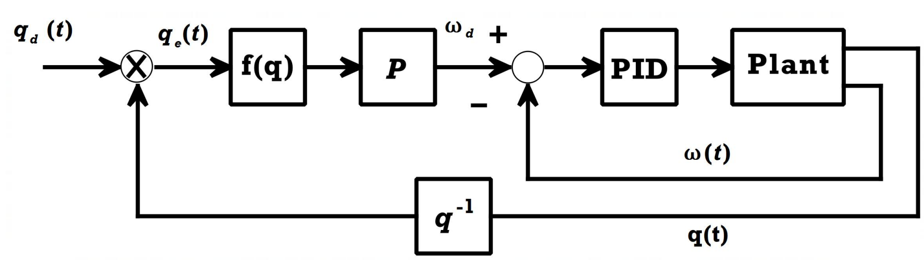

Figure 3.4: A schematic view of the attitude and angular velocity controllers.

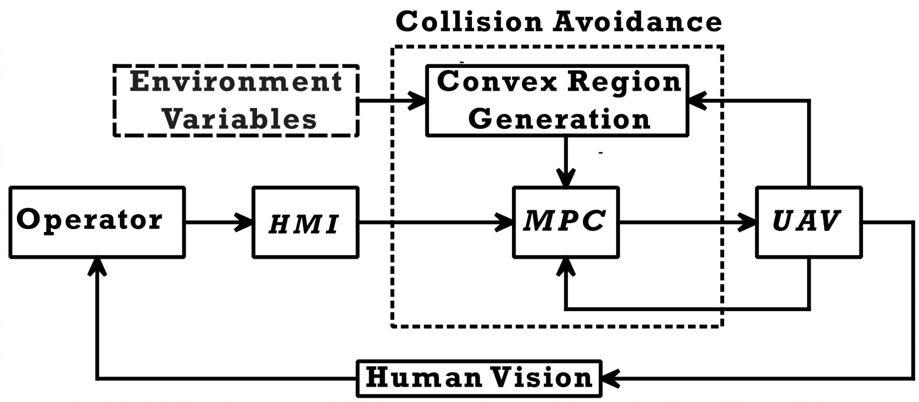

Figure 3.5: A schematic view of the collision avoidance algorithm.

3.5 Collision Avoidance

A model predictive collision avoidance routine modifies the operator's commands to ensure collision-free operation. The collision avoidance problem is formulated as a convex constrained optimization problem over a rolling control horizon with the objective of minimizing interference with operator commands while keeping the UAV in an obstacle-free space.

A schematic view of the collision avoidance algorithm is presented in Fig. 3.5. In this approach, obstacle-free space is identified and estimated by safe convex region and formulated as convex constraints in MPC-based collision avoidance algorithm. The MPC algorithm is formulated to minimize interference with the operator's commands while ensuring collision-free operation. Constraints related to the system dynamics, actuators limitation, and safety considerations are included in the problem formulation.

The collision avoidance algorithm runs at a slower rate than other controllers in the hierarchical control structure. This allows for a discrete-time formulation of the MPC-based collision avoidance in this thesis. To this end, the continuous-time system dynamics in (3.7) are discretized as follows

$$\boldsymbol{x}[k+1] = \boldsymbol{A}_d\boldsymbol{x}[k] + \boldsymbol{B}_d\boldsymbol{u}[k] \tag{3.27}$$

Matrices $\boldsymbol{A}_d \in \mathbb{R}^{6\times 6}$ and $\boldsymbol{B}_d \in \mathbb{R}^{6\times 3}$ are given by

$$\boldsymbol{A}_d = \begin{bmatrix} \boldsymbol{I}_3 & T_s\boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{I}_3 \end{bmatrix}, \; \boldsymbol{B}_d = \begin{bmatrix} T_s^2/2\boldsymbol{I}_3 \\ T_s\boldsymbol{I}_3 \end{bmatrix} \tag{3.28}$$

where T_s is the sampling time. The discrete-time MPC is formulated as follows

$$\min \sum_{k=0}^{N-1} (\boldsymbol{u}[k] - \boldsymbol{u}_h[k])^T \boldsymbol{P}[k](\boldsymbol{u}[k] - \boldsymbol{u}_h[k]) \tag{3.29a}$$

$$\text{s. t.} : \boldsymbol{x}[k+1] = \boldsymbol{A}_d\boldsymbol{x}[k] + \boldsymbol{B}_d\boldsymbol{u}[k] \tag{3.29b}$$

$$\boldsymbol{A}_u\boldsymbol{u}[k] \preceq \boldsymbol{b}_u \tag{3.29c}$$

$$\boldsymbol{A}_x[k]\boldsymbol{x}[k] \preceq \boldsymbol{b}_x[k] \tag{3.29d}$$

where $\boldsymbol{u}[\cdot] \in \mathbb{R}^3$ and $\boldsymbol{x}[\cdot] \in \mathbb{R}^6$ are the UAV's modified inputs and states, and $\boldsymbol{u}_h[\cdot] \in \mathbb{R}^3$ represents the operator's acceleration commands. $\boldsymbol{P}[\cdot] \in \mathbb{R}^3$ and N is the number of samples in the MPC control time horizon. Moreover, (3.29b) are constraints related to the system dynamics. The input constraints (3.29c) represent actuators limits. The constraints in (3.29d) define the approximated safe convex region and velocity related safety constraints imposed by the operator; $\boldsymbol{A}[k] \in \mathbb{R}^{m_d\times 6}$ and $\boldsymbol{b}_x[k] \in \mathbb{R}^{m_d}$ are the parameters of the safe convex region and m_d is the number of constraints.

Chapters 4-5 introduce two new methods for generating approximate convex safe region based on the reachability analysis. Chapter 6 presents a novel reachability-based collision avoidance algorithm that accounts for uncertainty in the system model and dynamics. Chapter 7 introduces a new collision avoidance algorithm with guaranteed recursive feasibility.

Chapter 4

Backward Reachability-based Collision Avoidance

In this chapter, collision avoidance assistance is formulated as a convex optimization problem that can be solved in real-time to find a globally optimal solution. The new collision avoidance method introduced in this chapter employs a novel backward reachability-based obstacle detection algorithm to identify obstacles with a chance of collision within a rolling control time horizon. The computational cost of this step is significantly reduced by using an ellipsoidal technique and finding analytical solutions to all relevant differential equations.

Following the identification of potentially dangerous obstacles, the SVM algorithm is employed to construct a safe polyhedral convex region around the UAV that would keep it away from obstacles with the chance of collision. Finally, collision assistance is formulated as a convex optimization problem over the rolling control horizon, with the objective of minimizing interference with the operator acceleration commands while ensuring that the UAV remains in the safe collision-free space. The approximation to the collision-free space is repeatedly updated as the UAV moves through the environment so only relevant obstacles in the control time horizon are considered.

4.1 Optimization-based Collision Avoidance

In the proposed shared control strategy, a human operator teleoperates the UAV by providing translational acceleration commands, while an optimization-based algorithm modifies the operator's commands to prevent potential collisions with obstacles. The objective is to automate the task of avoiding collisions to reduce the operator's workload. The operator can then focus on tasks that benefit most from the human cognitive and decision-making capabilities.

A schematic view of this shared control strategy is presented in Fig. 4.1. The collision avoidance algorithm consists of three parts. First, the Backward Reachability-based Obstacle Detection (BROD) algorithm employs a reachability analysis to determine a subset of obstacles that could potentially collide with the UAV within the control horizon of a model-predictive controller. Then, the safe convex region formation (SCRF) algorithm approximates the obstacle-free space with a convex set. Finally, an optimization-based model predictive controller (MPC) algorithm uses this convex set to compute corrective commands that would ensure the UAV would not collide with the obstacles within its control time horizon.

The translational motion dynamics of the UAV are formulated as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{4.1}$$

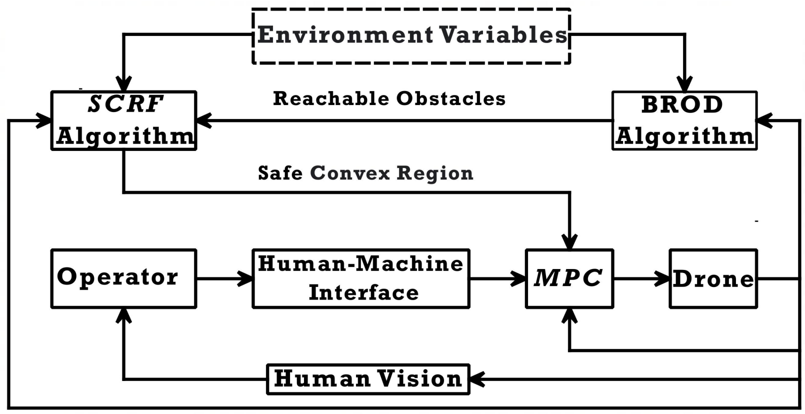

Figure 4.1: A schematic view of the proposed shared control strategy with automatic collision avoidance assistance.

The matrices $\boldsymbol{A} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{B} \in \mathbb{R}^{6\times3}$ are defined in (3.8)

Obstacles are modelled as convex polytopes

$$\boldsymbol{A}_o^i \boldsymbol{x} \preceq \boldsymbol{b_o^i} \quad i = 1, 2, .., n_o \tag{4.2}$$

where $\boldsymbol{A}_o^i \in \mathbb{R}^{m_p^i \times 3}$ and $\boldsymbol{b_o^i} \in \mathbb{R}^{m_o^i}$ are the parameters of i-th (out of n_o) polytopic shape obstacle, where m_o^i is the number of faces of i-th obstacle. Furthermore, feasible input and velocity sets due to the actuators limitation and safety consideration are modelled as polytopes,

$$\boldsymbol{A}_u \boldsymbol{u} + \preceq \boldsymbol{b}_u \tag{4.3}$$

$$\boldsymbol{A}_v \boldsymbol{v} + \preceq \boldsymbol{b}_v \tag{4.4}$$

where $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^3$ represent feasible input and velocity vectors, respectively. $\boldsymbol{A}_u \in \mathbb{R}^{m_u \times 3}$ and $\boldsymbol{b}_u \in \mathbb{R}^{m_u}$ are the parameters of polytopic input set, where m_u is the number of polytopic input faces. $\boldsymbol{A}_v \in \mathbb{R}^{m_v \times 3}$ and $\boldsymbol{b}_v \in \mathbb{R}^{m_v}$ are the parameters of polytopic velocity set, where m_v represents the number of faces of the polytopic velocity set. Moreover, the UAV is modelled as a sphere with center $\boldsymbol{p}_{cg}$ and radius R_d.

4.2 Reachability-based Obstacle Detection

The BROD algorithm detects a subset of obstacles that can potentially collide with the UAV within the time horizon of the MPC-based controller. By eliminating obstacles with no chance of collision, the algorithm should in principle help create a less conservative convex approximation of the obstacle-free space for the UAV. To this end, All the states where the UAV collides with the obstacle are modelled as an undesirable set. A backward reach tube for each undesirable set corresponding to each obstacle is calculated within the MPC time horizon. Any obstacle whose reach tube intersect with the UAV can potentially collide with it within the control time horizon and is included in convex safe region approximation.

Detecting intersection between geometrical shapes especially in higher dimension can be challenging or impossible for real-time applications. In this thesis, the UAV is represented by a point and the obstacles are enlarged to overestimation of the Minkowski sum of polytope and sphere computed as [100]

$$\boldsymbol{A}_o^i \boldsymbol{p} \preceq \boldsymbol{b_o^i} + R_d \|\boldsymbol{A}_o^i\|_\bullet \quad i = 1, 2, .., n_o \tag{4.5}$$

where $\|\boldsymbol{A}_o^i\|_\bullet = [\|\boldsymbol{A}_{o_{1,*}}^i\|_2,\ \|\boldsymbol{A}_{o_{2,*}}^i\|_2,\ ...,\ \|\boldsymbol{A}_{o_{m_i,*}}^i\|_2]^T$ and $\boldsymbol{A}_{o_{j,*}}^i$ is the normal vector of the j-th face of the i-th obstacle and m_i is the number of faces of i-th obstacle. Consequently, each undesirable set is defined as all the states where the UAV collides with each enlarged obstacle; then, the backward reach tube corresponding to each undesirable set is calculated. Finally, detecting intersection between the UAV and the reach tube is simplified to that of a point (UAV's current state) and the reach tube.

The backward reach tube can be approximated by the union of the backward reach sets computed at each time step of the MPC within its control time horizon (B.10). In this work, each reach set is approximated by the intersection of the external tight ellipsoids introduced in Appendix B. Therefore, the UAV can potentially collide with an obstacle if its current state falls within the intersection of all external ellipsoids corresponding to the obstacle computed at each time step of the MPC within the control horizon. A schematic view of the BROD algorithm is depicted in Fig.4.2. In this figure, the UAV does not intersect with any reach set at time t_1. Therefore, none of the obstacles are reachable for the UAV at the final time starting from t_1. However, as demonstrated in Fig. 4.2(b), the reach set related to obstacle c intersects with the UAV at time t_2. Therefore, the backward reach tube related to obstacle c intersects with the UAV and this obstacle is identified as an obstacle with the chance of collision.

To estimate the reach tube corresponding to each obstacle, the parameters of external ellipsoids need to be calculated. Assuming the ellipsoidal input set with constant parameters over the control time horizon, i.e., $\mathbb{E}(\boldsymbol{q}_u(t), \boldsymbol{Q}_u(t)) = \mathbb{E}(\boldsymbol{q}_u, \boldsymbol{Q}_u))$, and by substituting (B.32) and (B.30) into (B.28), the center of ellipsoids for the k-th time

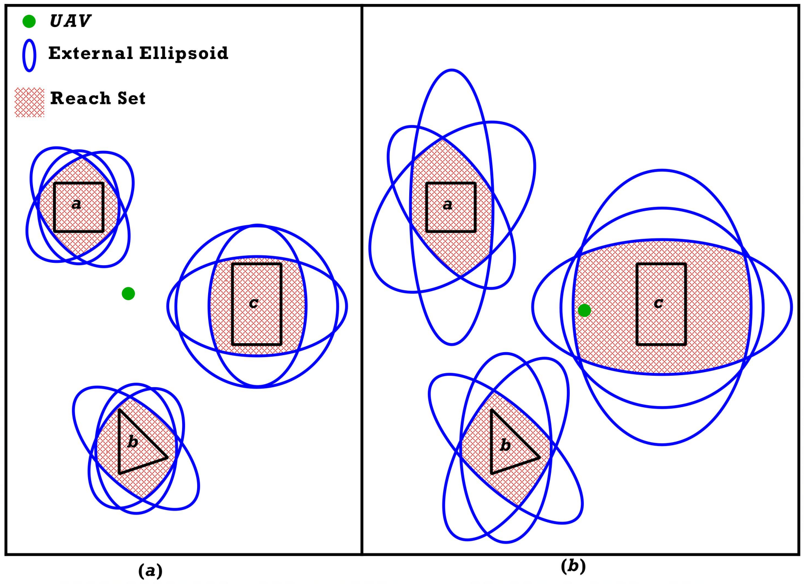

Figure 4.2: A schematic view of the collision detection based on the BROD algorithm at two time instances t_1 and t_2 whitin the the MPC time horizon. a) Reach sets do not intersect with the UAV at time t_1. b) Reach set corresponding to obstacle c intersects with the UAV at time t_2.

instance of interest is calculated by

$$\boldsymbol{q}_w^i(t_k) = \begin{bmatrix} \frac{1}{2}\boldsymbol{q}_u(t_f - t_k)^2 - \boldsymbol{q}_{v_f}^i(t_f - t_k) + \boldsymbol{q}_{p_f}^i \\ -\boldsymbol{q}_u(t_f - t_k) + \boldsymbol{q}_{v_f}^i \end{bmatrix} \tag{4.6}$$

Here $\boldsymbol{q}_{p_f}^i, \boldsymbol{q}_{v_f} \in \mathbb{R}^3$ are the position and velocity vectors of the center of ellipsoidal undesirable set corresponding to the i-th obstacle ($\boldsymbol{q}_f^i = [\boldsymbol{q}_{p_f}^{i\ T}, \boldsymbol{q}_{v_f}^T]^T$), $\boldsymbol{q}_u(t) \in \mathbb{R}^3$ is the center of ellipsoidal input set, and n_o is the number of obstacles.

Using (B.33) and (B.31), the shape matrix of each ellipsoid in each direction ($\boldsymbol{l} \in \mathbb{R}^6$)

is determined as

$$\boldsymbol{Q}^i_{w_l}(t_k) = \boldsymbol{G}(t_k, t_f)\Big(\sqrt{\boldsymbol{l}^T\boldsymbol{Q}^i_f\boldsymbol{l}} + [h(\tau)]_0^{t_f-t_k}\Big) \times \Big(\frac{\boldsymbol{Q}^i_f}{\sqrt{\boldsymbol{l}^T\boldsymbol{Q}^i_f\boldsymbol{l}}} + [\boldsymbol{H}(\tau)]_0^{t_f-t_k}\Big)\boldsymbol{G}(t_k, t_f)^T \quad (4.7)$$

where $h(\tau)$ and $\boldsymbol{H}(\tau)$ are the solutions of the integral terms in (B.33) (see appendix C.1).

The parameters of the ellipsoidal input set $\mathbb{E}(\boldsymbol{q}_u, \boldsymbol{Q}_u)$ and ellipsoidal undesirable set $\mathbb{E}(\boldsymbol{q}^i_f, \boldsymbol{Q}^i_f)$ must be approximated due to polytopic shape of input and undesirable sets. These are needed for the computation of the external ellipsoids. The polytopic input set is approximated by the minimum volume ellipsoid covering it. This so-called Lowner-John ellipsoid [17] is obtained by solving the following convex optimization problem.

$$\min \quad \log\det \boldsymbol{L}_{\boldsymbol{u}}^{-1} \quad (4.8)$$

$$\text{s.t.} \quad ||\boldsymbol{L}_u\boldsymbol{v}_{u_i} + \boldsymbol{c}_u||_2 \leq 1, \qquad i = 1, ..., n_u \quad (4.9)$$

where $\boldsymbol{L}_u \in \mathbb{R}^{3\times3}$ and $\boldsymbol{c}_u \in \mathbb{R}^3$ are the optimization variables, and $\boldsymbol{v}_{u_i}$ and n_u are the i-th vertex of polyhedron and the number of polyhedron vertices, respectively. The ellipsoidal input set is defined as

$$\boldsymbol{q}_u = -\boldsymbol{L}_u^{-1}\boldsymbol{c}_u \quad (4.10)$$

$$\boldsymbol{Q}_u = (\boldsymbol{L}_u^T\boldsymbol{L}_u)^{-1} \quad (4.11)$$

Ellipsoidal undesirable set corresponding to each obstacle must cover the related enlarged obstacle (7.12) for an arbitrary reachable velocity (5.16c). Therefore, the parameters of the ellipsoidal undesirable set are defined as (proof in C.2)

$$\boldsymbol{q}_f^i = \begin{bmatrix} \boldsymbol{q}_{p_f}^i \\ \boldsymbol{q}_{v_f} \end{bmatrix} \tag{4.12}$$

$$\boldsymbol{Q}_f^i = \begin{bmatrix} \frac{1}{1-\epsilon}\boldsymbol{Q}_{p_f}^i & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \frac{1}{\epsilon}\boldsymbol{Q}_{v_f} \end{bmatrix} \quad i = 1, ..., n_o \tag{4.13}$$

where $\boldsymbol{q}_{p_f}^i \in \mathbb{R}^3$ and $\boldsymbol{Q}_{p_f}^i \in \mathbb{R}^{3\times 3}$ are the parameters of minimum volume ellipsoid covering the i-th enlarged polytopic obstacle, $\boldsymbol{q}_{v_f} \in \mathbb{R}^3$ and $\boldsymbol{Q}_{v_f} \in \mathbb{R}^{3\times 3}$ are the parameters of minimum volume ellipsoid covering the polytopic velocity set. $0 < \epsilon < 1$ is the scaling factor. By choosing ϵ sufficiently small ($\epsilon \to 0$), the ellipsoidal undesirable set approaches to minimum volume ellipsoid covering the obstacle for any arbitrary velocity.

Parameters of the minimum volume ellipsoid covering the i-th enlarged obstacle are calculated as

$$\boldsymbol{q}_{p_f}^i = -{\boldsymbol{L}_o^i}^{-1}\boldsymbol{c}_o^i \tag{4.14}$$

$$\boldsymbol{Q}_{p_f}^i = ({\boldsymbol{L}_o^i}^T\boldsymbol{L}_o^i)^{-1} \tag{4.15}$$

where $\boldsymbol{c}_o^i \in \mathbb{R}^3$ and $\boldsymbol{L}_o^i \in \mathbb{R}^{3\times 3}$ are optimization variables.

$$\textit{minimize} \qquad \log\det(\boldsymbol{L}_o^i)^{-1} \tag{4.16}$$

$$\textit{subject to} \qquad ||\boldsymbol{L}_o^i\boldsymbol{v}_{o_j}^i + \boldsymbol{c}_o^i||_2 \le 1, \qquad j = 1, ..., n_i \tag{4.17}$$

Here $\boldsymbol{v}^i_{o_j}$ is the j-th vertex of the i-th polytope and n_i is the number vertices of the i-th polytope.

The parameters of the minimum volume ellipsoid covering the velocity set are obtained as

$$\boldsymbol{q}_{v_f} = -\boldsymbol{L}_v{}^{-1}\boldsymbol{c}_v \tag{4.18}$$

$$\boldsymbol{Q}_{v_f} = (\boldsymbol{L}_v{}^T\boldsymbol{L}_v)^{-1} \tag{4.19}$$

where $\boldsymbol{c}_v \in \mathbb{R}^3$ and $\boldsymbol{L}_v \in \mathbb{R}^{3\times 3}$ are optimization variables.

$$\min \qquad \log\det(\boldsymbol{L}_v)^{-1} \tag{4.20}$$

$$\text{s.t.} \qquad ||\boldsymbol{L}_v\boldsymbol{v}_{v_i} + \boldsymbol{c}_v||_2 \leq 1, \qquad i = 1, ..., n_v \tag{4.21}$$

Here $\boldsymbol{v}_{v_i}$ and n_v are the i-th vertex of polytope and the number of polytope vertices.

4.3 Safe Convex Region Formation Based on SVM

Once obstacles with potential for collision within the MPC time horizon are identified, a safe obstacle-free convex region is created around the UAV and is added as constraints to the optimal control problem in MPC. Constraints in the form of half-spaces defined by linear inequalities are used here to ensure the resulting MPC optimization problem can be solved in real-time. To this end, the SVM algorithm [41] is used to create separating planes between the UAV and the obstacles, defining the half-spaces that form the safe obstacle-free region.

In hard-margin SVM, the goal is to find the separating plane between two disjoint sets Φ and Ψ in which the closest members of each set to the plane create the maximum distance to it. This goal is achieved by solving the following optimization [41]

$$\max_{\boldsymbol{\alpha},\ \eta} (\min_{\boldsymbol{x}_k \in \Phi} \frac{|\boldsymbol{\alpha}^T \boldsymbol{x}_k + \eta|}{\|\boldsymbol{\alpha}\|_2} + \min_{\boldsymbol{y}_l \in \Psi} \frac{|\boldsymbol{\alpha}^T \boldsymbol{y}_l + \eta|}{\|\boldsymbol{\alpha}\|_2} \tag{4.22}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^3$ and $\eta \in \mathbb{R}$ are the parameters of separating plane, $\boldsymbol{x}_k \in \mathbb{R}^3$ and $\boldsymbol{y}_l \in \mathbb{R}^3$ are the k-th and l-th members of sets Φ and Ψ, respectively. $|.|$ denotes the absolute value.

In utilizing SVM algorithm for finding the separating plane between the UAV and each obstacle, Ψ is a singleton set representing the UAV's position ($\Psi = \{\boldsymbol{p}_{cg}\}$) and Φ_i denotes the vertices of the i-th enlarged obstacle ($\Phi_i = \{\boldsymbol{v}^i_{o_1},\ \boldsymbol{v}^i_{o_2}, \dots\ ,\ \boldsymbol{v}^i_{o_{n_i}}\}$). The separating plane corresponding to each enlarged obstacle intersects with its closest vertex, i.e.,

$$\min_{\boldsymbol{v}^i_{o_j} \in \Phi_i} |\boldsymbol{\alpha}_i^T \boldsymbol{v^i_{o_j}} + \eta_i| = 0, \qquad j = 1, ..., n_i \tag{4.23}$$

Here, $\boldsymbol{\alpha}_i \in \mathbb{R}^3$ and $\eta_i \in \mathbb{R}$ are the parameters of i-th separating plane. Without loss of generality, it is assumed $min|\boldsymbol{\alpha_i^T} \boldsymbol{p}_{cg} + \eta_i| = 1$ [41]. The SVM algorithm for finding the i-th separating plane can be formulated as the following optimization problem

$$\begin{aligned} \min \quad & \boldsymbol{\alpha_i^T \alpha_i} \\ \text{s.t.} \quad & \boldsymbol{\alpha_i^T} \boldsymbol{p}_{cg} + \eta_i \geq 1 \\ & \boldsymbol{\alpha_i^T} \boldsymbol{v}^i_{o_j} + \eta_i \leq 0 \qquad j = 1, ..., n_i \end{aligned} \tag{4.24}$$

4.4 Collision Avoidance Based on MPC

The MPC-based collision avoidance assistance is formulated in a way to minimize interference with the operator's commands while ensuring collision-free operation. Constraints related to the system dynamics, actuator, and the collision-free safe convex region are considered in the problem formulation. The MPC is formulated in discrete time as follows

$$\min \sum_{k=0}^{N-1} (\boldsymbol{u}[k] - \boldsymbol{u}_h[k])^T \boldsymbol{P}[k] (\boldsymbol{u}[k] - \boldsymbol{u}_h[k]) \tag{4.25a}$$

$$\text{s.t.} : \boldsymbol{x}[k+1] = \boldsymbol{A}_d \boldsymbol{x}[k] + \boldsymbol{B}_d \boldsymbol{u}[k] \tag{4.25b}$$

$$\boldsymbol{A}_u \boldsymbol{u}[k] \preceq \boldsymbol{b}_u \tag{4.25c}$$

$$\boldsymbol{A}_v \boldsymbol{v}_{cg}[k] \preceq \boldsymbol{b}_v \tag{4.25d}$$

$$\boldsymbol{A}_p \boldsymbol{p}_{cg}[k] \preceq \boldsymbol{b}_p \tag{4.25e}$$

where $\boldsymbol{u}[\cdot] \in \mathbb{R}^3$ and $\boldsymbol{x}[\cdot] \in \mathbb{R}^6$ are the UAV's modified inputs and states, and $\boldsymbol{u}_h[\cdot] \in \mathbb{R}^3$ represents the operator acceleration command. Here $\boldsymbol{P}[\cdot] = \boldsymbol{I}_3$ and N is the number of samples in the MPC control time horizon. Moreover, (4.25b) are constraints related to the system dynamics where $\boldsymbol{A}_d \in \mathbb{R}^{6\times 6}$ and $\boldsymbol{B}_d \in \mathbb{R}^{6\times 3}$ are defined in (3.28). The input and velocity constraints (4.25c) and (4.25d) are identical to those in (4.3) and (4.4) and represent actuators limits and safety considerations defined by the operator. The constraint (4.25e) defines the approximated polyhedral safe convex region created by the SCRF algorithm in the previous section, where $\boldsymbol{A}_p \in \mathbb{R}^{m_d \times 3}$ and $\boldsymbol{b}_p \in \mathbb{R}^{m_d}$ are the parameters of the safe convex region and m_d is the number of obstacles with chance of collision.

4.5 Experiment

A schematic view of the experimental setup is shown in Fig. 4.3. The operator commands UAV's desired acceleration via the human-machine interface (HMI). An OptiTrack Flex 13 motion capture system measures UAV's position and orientation. The MPC-based collision assistance algorithm computes the actual UAV's acceleration command. The reference commands for the UAV's angular velocities are obtained by assuming zero reference attitude and using the attitude controller in [36]. The UAV's linear acceleration angular velocity commands are sent to the on-board flight controller PID loops. The overall update rate for the MPC-based collision avoidance algorithm is $f_s = 40$ Hz.

A fully-actuated omni-copter [36] is used for the experiments in this work, i.e., see Fig. 4.3. This UAV can generate force and torque in all possible directions. The Phantom Premium 1.5 haptic interface is the human-machine interface in the experiments.

Several optimization problems need to be solved in implementing the collision assistance algorithm. The Lowner-John theorem [17] is used to find the minimum volume ellipsoids covering the polyhedral input, velocity and obstacle sets. This results in semidefinite programs [13] which are solved off-line by the Mosek solver [5]. The approximated safe convex region around the UAV is obtained in real time by the SVM algorithm. This problem is a standard quadratic program and is solved by the OSQP [86]. The real-time MPC optimization is also formulated as standard quadratic program and solved by the OSQP.

The time horizon for the MPC algorithm in (4.25a) is set to 1.5 seconds. The operator's command is assumed constant over the controller prediction horizon. Moreover, it is assumed the obstacles positions are known. Two-dimensional representation of the obstacles are shown in Fig. 4.4 and Fig. 4.7 where the height of each obstacle is one meter. Finally, the parameters of input and velocity constraints in (4.25c) and (4.25d) are selected as

$$\boldsymbol{A}_u = \boldsymbol{A}_v = [-\boldsymbol{I}_3,\ \boldsymbol{I}_3]^T \tag{4.26}$$

$$\boldsymbol{b}_u = [2,\ 0.3,\ 2,\ 2,\ 0.3,\ 2]^T \tag{4.27}$$

$$\boldsymbol{b}_v = [1.5,\ 0.3,\ 1.5,\ 1.5,\ 0.3,\ 1.5]^T \tag{4.28}$$

The reach set is approximated by intersecting 1120 ellipsoids calculated along 1120 different directions in (4.7). Computing these ellipsoids in real time would be prohibitive. However, owing to the linear time-invariant model of the UAV and the fixed shape matrices of ellipsoidal input and undesirable sets, the parameters of external ellipsoids are time-invariant and can be computed off-line. Finally, the scaling factor is set to $\epsilon = 4 \times 10^{-2}$ in (4.13).

4.5.1 Flight Test

Two experiments are designed to demonstrate the performance of the proposed collision avoidance algorithm. In the first experiment, the importance of reachability analysis in collision avoidance is investigated. In the second experiment, the proposed collision avoidance algorithm which exploits features of MPC and reachability analysis is implemented on the UAV.

Two scenarios are considered in the first experiment. The first scenario involves collision avoidance assistance without reachability analysis (SCRF+MPC). In the

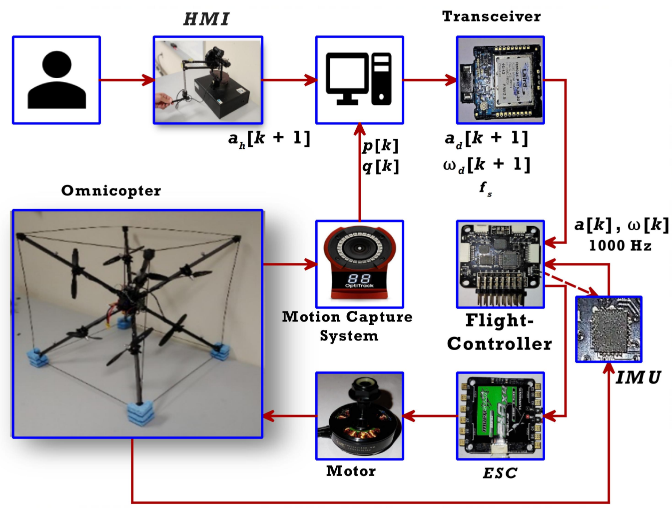

Figure 4.3: Experimental setup including the Omnicopter, motion capture system, HMI and Transceiver

second scenario (BROD+SCRF+MPC), the reachability analysis is added to the collision avoidance assistance algorithm. The trajectories of the UAV in the first scenario are shown in Fig. 4.4(a). In this scenario, both obstacles are considered for convex region formation in the SCRF algorithm and separating planes are created for each of them. These planes act as constraints in the MPC algorithm and erroneously block the path in front of the UAV. This leads the MPC to produce undesired corrective commands as depicted in Fig. 4.5(a).

The trajectory of the UAV in the second scenario of the first experiment is presented in Fig. 4.4(b). Due to the use of the reachability analysis, both obstacles are identified safe as they would not be reachable by the UAV during the MPC time horizon. The undesirable corrective commands are absent in this scenario, i.e., see Fig. 4.5(a). The operator's commands in both scenarios are shown in Fig. 4.5(b) and Fig. 4.5(c), respectively.

Fig. 4.6 underscores the positive role of reachability analysis in obstacle detection and avoidance. Here $\Delta t = t_f - t$ where t_f is the MPC time horizon and t is the time instance of interest at which the reach tube is calculated. As is evident in this figure, the UAV is outside the reach tubes corresponding to the obstacles which render them safe. The estimated reach sets at the beginning and end of the MPC time horizon are depicted in Fig. 4.6(b) and Fig. 4.6(c), respectively. The UAV is outside of each reach set. Since the UAV moving in X-direction (the velocity is positive in X-direction), the center of the reach set shifts in negative X-direction (Fig. 4.6(c)), i.e., given the positive velocity in X-direction, the possibility of collision increases when the UAV is in front of the obstacles rather than when it has passed the obstacles.

In the second experiment, three obstacles are placed in the UAV operating environment. The obstacles are placed on the sides and in the front of the UAV. As it is evident in Fig. 4.7, no separating planes is generated for the first two obstacles as they are deemed safe outside the reachable space of the UAV. The third obstacle in the front of UAV is identified as potentially dangerous and a separating plane is created for it. This results in the corrective commands shown in Fig. 4.8(a), which modify the operator's commands in Fig. 4.8(b) in order to prevent the UAV from colliding with the obstacle.

Fig. 4.9 demonstrates the utility of the BROD algorithm. It can be seen in Fig. 4.9(a) that the UAV only intersects with the reach tube associated with the third obstacle, meaning that this is the only obstacle that could be reached over the MPC time horizon. Note that estimated reach sets at the beginning and end of the MPC time horizon are depicted in Fig. 4.9(b) and Fig. 4.9(c).

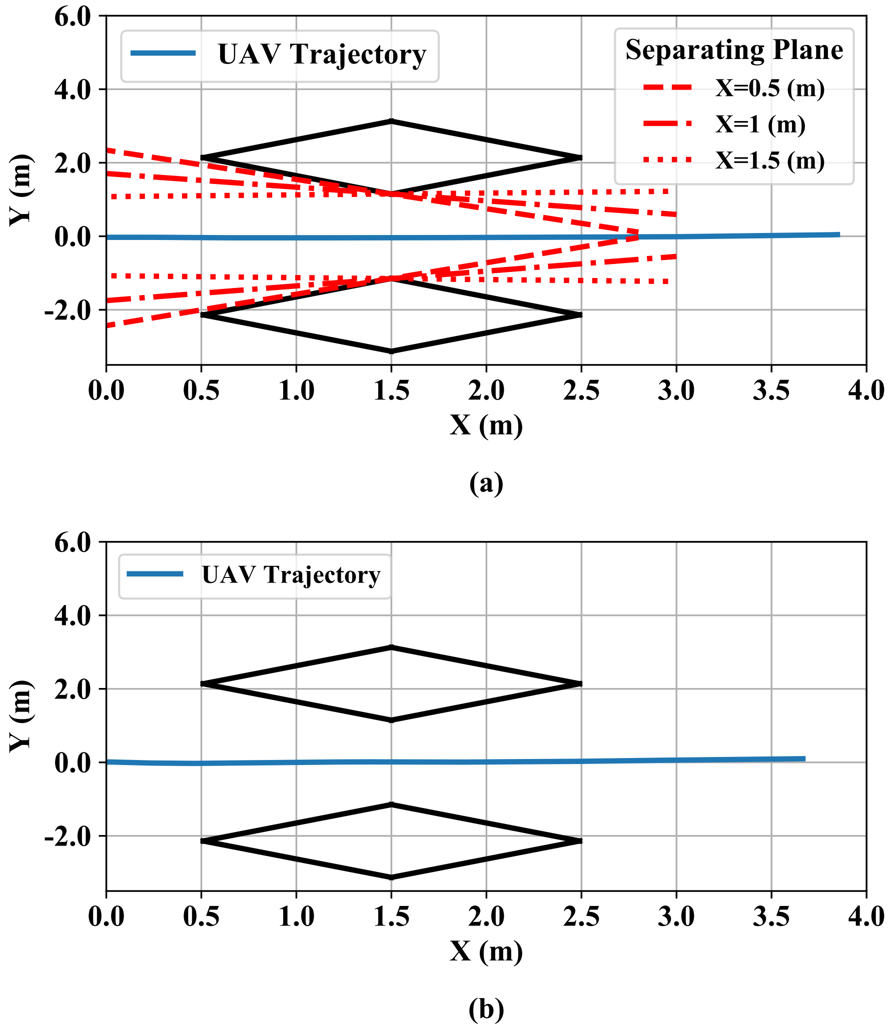

Figure 4.4: Two dimensional representation of the UAV trajectory using the collision avoidance algorithm a) without reachability analysis b) with reachability analysis.

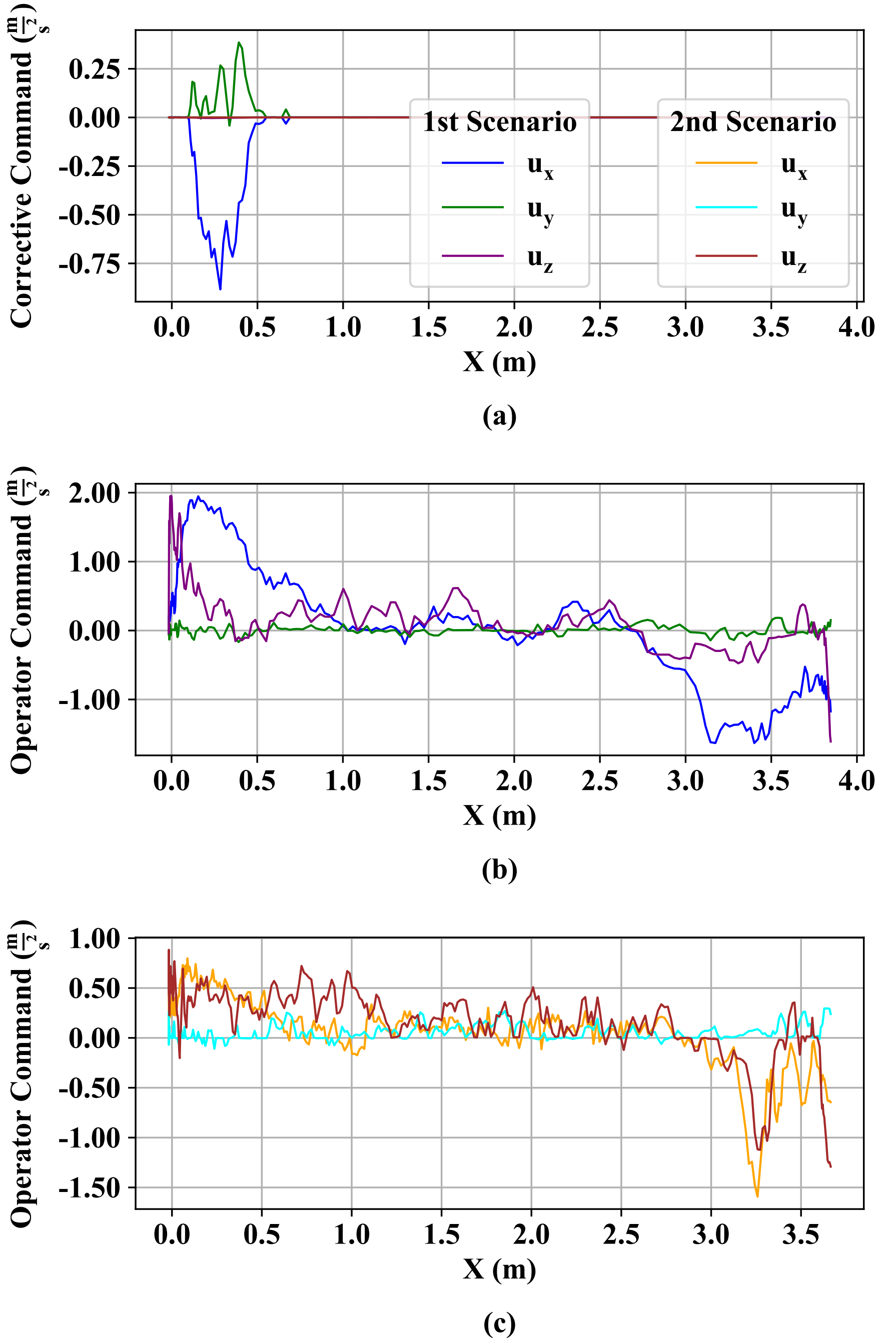

Figure 4.5: a) Corrective commands created in the first scenario (without reachability analysis) and in the second scenario (with reachability analysis). b) Operator's commands in the first scenario. c) Operator's commands in the second scenario.

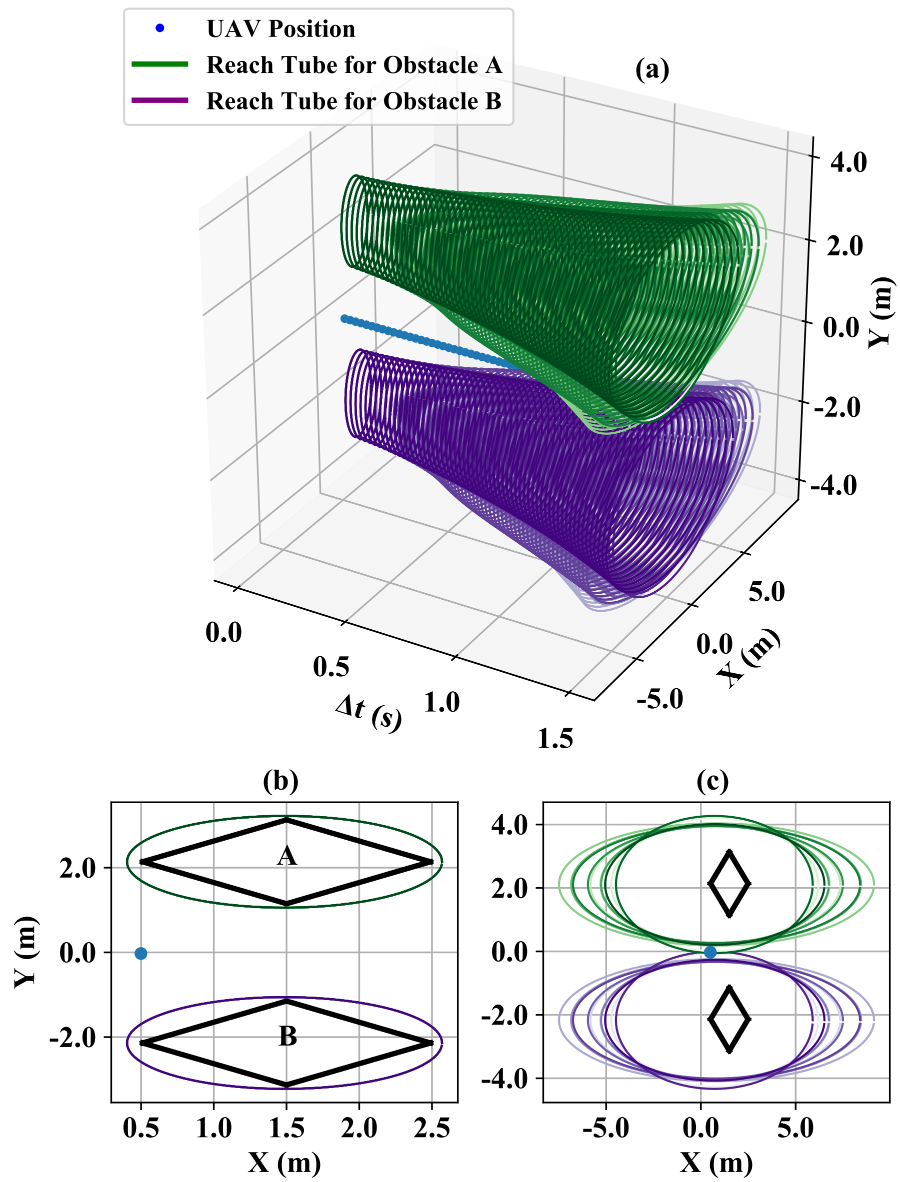

Figure 4.6: Intersection of six-dimensional estimated backward reach tubes with velocities and altitude of the UAV for each obstacle when the UAV reaches to $x = 0.5m$ a) within the MPC time horizon, b) at the beginning of MPC time horizon, c) at the end of MPC time horizon.

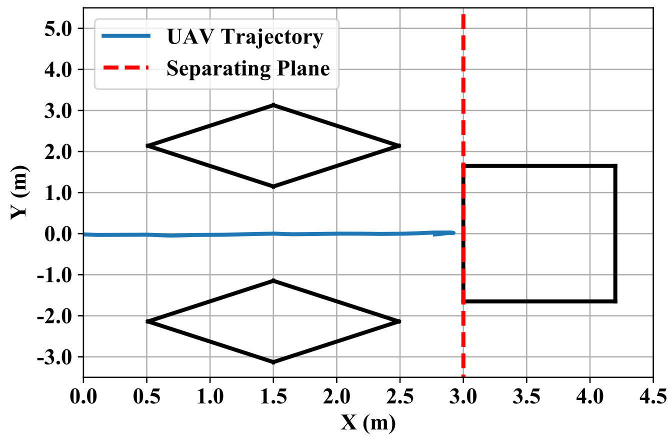

Figure 4.7: Two dimensional representation of the UAV trajectory using the proposed collision avoidance algorithm.

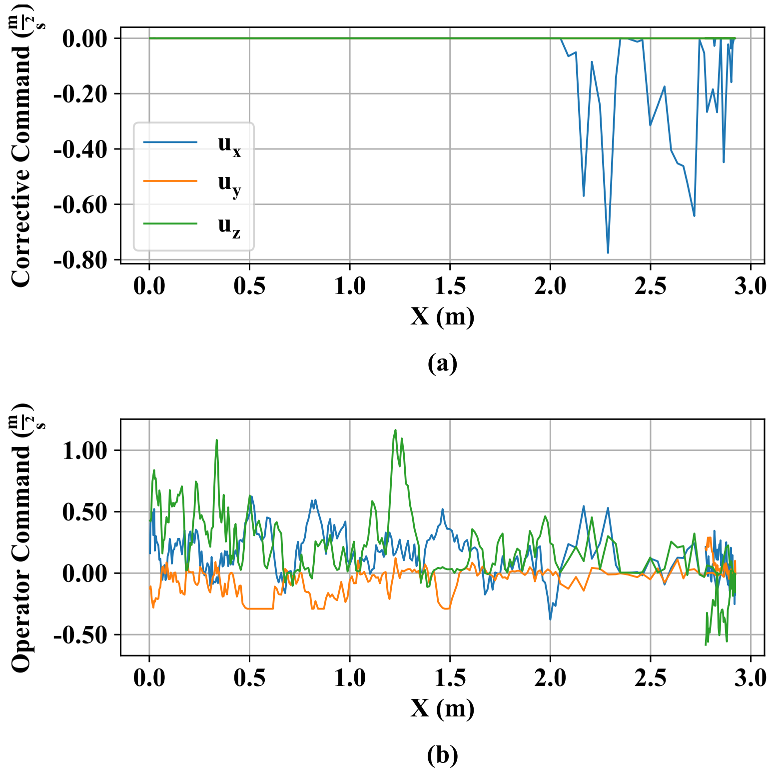

Figure 4.8: a) Corrective commands created by the proposed collision avoidance algorithm. b) Operator's commands.

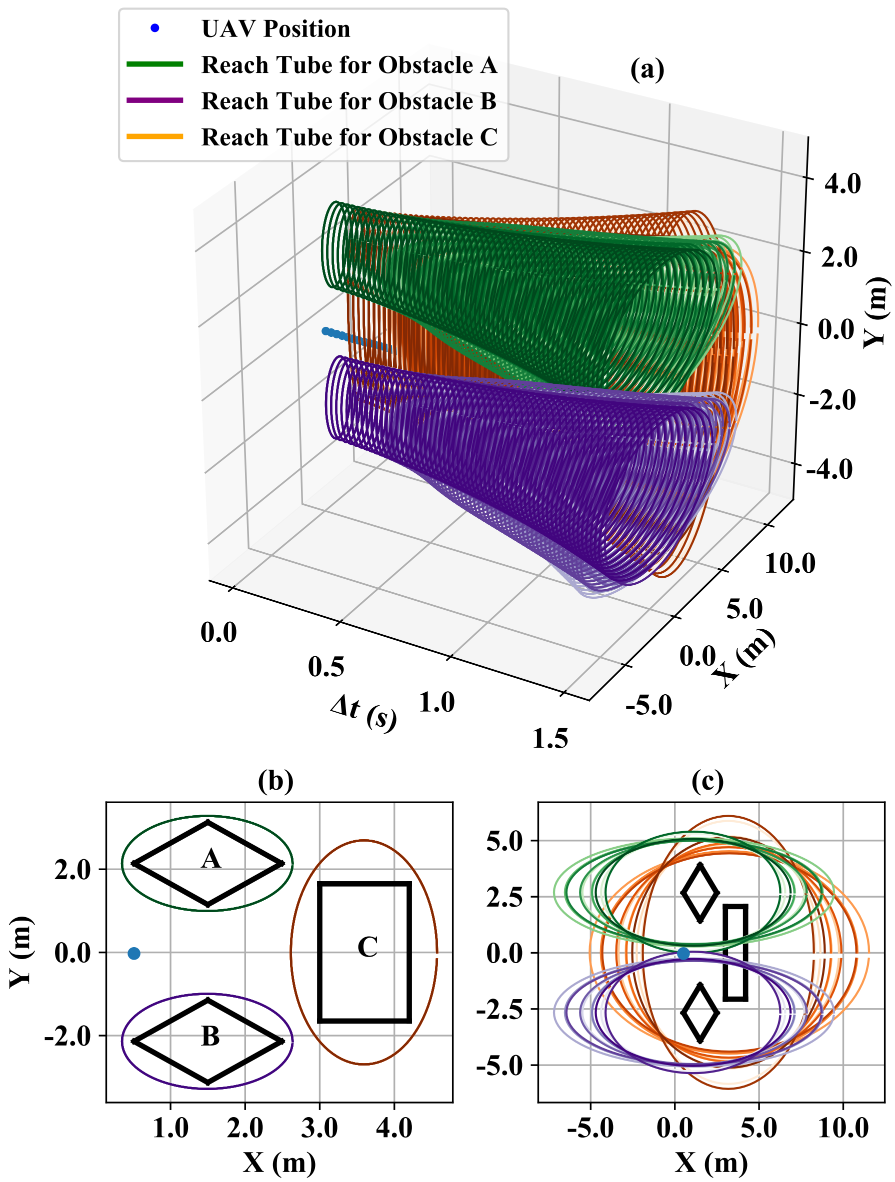

Figure 4.9: Intersection of six-dimensional estimated backward reach tubes with velocities and altitude of the UAV for each obstacle when the UAV is at $x = 0.5m$ a) within MPC time horizon, b) at the beginning of MPC time horizon, c) at the end of MPC time horizon.

Chapter 5

Forward Reachability-based Collision Avoidance

In this chapter, a novel method is introduced to generate a convex approximation of the obstacle-free region in the taskspace of a mobile robot with linear dynamics. The main contribution of this work is in utilizing the reachability analysis of linear dynamical systems to produce a less conservative convex approximation of the obstacle-free space than what otherwise would be possible. First, an approximate reachable region of the robot is obtained for each time step in the control horizon. Next, hyperplanes are created to separate the so-called target point of the robot, i.e., its predicted state from the undesirable sets characterizing the obstacles that can potentially collide with the robot. These are obtained by maximizing the intersection of the reachable set with the halfspace defined by the hyperplane. The intersection of these halfspaces constitutes the safe obstacle-free space at a given time step in the control time horizon.

The proposed algorithm is general and can be applied to systems other than those in mobile robotic applications. In collision avoidance for mobile robots, this algorithm should improve the system performance for two principal reasons: (i) Among all the obstacles in the environment, only those with the chance of collision are considered in convex region generation. These obstacles are identified through a reachability analysis, (ii) The reachability analysis is also employed in creating the approximated safe convex region of the robot.

5.1 Safe Convex Region Generation Based on Reachability Analysis

Forward reach tube, as defined in Appendix B, is the set of all states where the system can reach during the time interval of interest. Undesirable set is defined as a set of states which should be avoided during the time interval of interest. If the reach tube intersects with the undesirable set, the undesirable set is reachable by the system states; it would be identified as an unsafe undesirable set during this time interval. A convex region must be constructed to designate the safe space, i.e., the space that separates the system states from this unsafe undesirable set. Ideally, this convex safe region should be as close as possible to the intersection of the complement of the undesirable set and the reach tube.

By approximating the reach tube with the union of a finite number of reach sets at time instances of interest in (B.6), the unsafe undesirable set can be determined from the intersection of the undesirable set and reach set at each of these time steps. The safe region can then be approximated by a convex approximation of the intersection of the complement of the unsafe undesirable set and the reach set.

The safe convex region at each time instance is approximated by half-spaces created by separating hyperplanes between the unsafe undesirable set and target point. These hyperplanes are chosen to maximize the intersection of the halfspace with the reachable set. The undesirable states at each time instance are modeled as polytopic set represented by its vertices, i.e., $\mathbb{P}_{x_u}(t_k) = convh\{\boldsymbol{v}_1(t_k), \boldsymbol{v}_2(t_k), ..., \boldsymbol{v}_{n_v}(t_k)\}$, where *convh* denotes the convex hull. The separating hyperplane is derived by solving the following optimization problem

$$\begin{aligned} &\max_{\boldsymbol{\alpha}(t_k),\eta(t_k)} && V_c(t_k) \\ &\text{s.t.} && \boldsymbol{\alpha}^T(t_k)\boldsymbol{x}_t + \eta(t_k) \geq 0 \\ & && \boldsymbol{\alpha}^T(t_k)\boldsymbol{v}_j(t_k) + \eta(t_k) \leq 0 \qquad j = 1,\ ...,\ n_v \end{aligned} \tag{5.1}$$

where $V_c(t_k),\ k = 1,\ ...,\ n_f$ is the inscribed volume between k-th separating hyperplane ($\boldsymbol{\alpha}(t_k) \in \mathbb{R}^{n_s},\ \eta(t_k)$) and the k-th reach set, n_f is the number of time instances of interest, n_s is the number of states of the system. $\boldsymbol{x}_t \in \mathbb{R}^{n_s}$ represents target point of the system and $\boldsymbol{v}_j(t_i) \in \mathbb{R}^{n_s}$ is the j-th (out of n_v) vertex of polytopic unsafe undesirable set at t_i.

A new algorithm is proposed here to solve this optimization problem and find the separating hyperplanes and the corresponding safe convex region, i.e., see Algorithm 1. First, each reach set is approximated by the minimum volume enclosing ellipsoid by giving the set of admissible inputs within the time interval of interest ($\mathbb{T}_e = [t_0, t_k]$) and the set of possible initial states ($\mathbb{P}_{x_0}$). To reduce computations, a conservative intersection detection algorithm is used to identify whether the undesirable set would intersect with each ellipsoidal reach set at the corresponding time instance by using a unique affine transformation. These undesirable sets are denoted

Algorithm 1 Algorithm for generating approximate safe convex region.

$k \leftarrow 0$
while $k \leq n_f$ **do**
  $\mathbb{P}_x(t_k) \leftarrow MVE\left(\mathbb{P}x_0, \mathbb{P}_u(t)_{t\in[t_0,t_k]})\right)$
  $ID,\ \mathbb{P}_y(t_k), \mathbb{P}_{y_u}(t_k) \leftarrow CID(\mathbb{P}_x(t_k), \mathbb{P}_{x_u}(t_k)$
  if $ID\ is\ True$ **then**
    $\boldsymbol{\alpha}(t_k), \eta(t_k) \leftarrow OSH(\mathbb{P}_y(t_k), \mathbb{P}_{y_u}(t_k))$
    return $\boldsymbol{\alpha}(t_k), \eta(t_k)$
  end if
  $k \leftarrow k + 1$
end while

as potentially unsafe undesirable sets. $\mathbb{P}_y(t_k)$ and $\mathbb{P}_{y_u}(t_k)$ represent reach set and undesirable set in transformed coordinates, respectively.

5.2 Reach Set Approximation based on Minimum Volume Enclosing Ellipsoid

A mathematical expression is required for the volume of the reach set to solve the optimization problem in (5.1). This volume is approximated by a minimum volume ellipsoid enclosing the reach set. The formula for calculating the parameters of minimum volume enclosing ellipsoid for a continuous-time linear system is provided in [72, 73]. It is assumed that the initial state set ($\mathbb{P}_{x_0}$) and input set ($\mathbb{P}_u(t)$) can be expressed as ellipsoidal sets

$$\mathbb{P}_{x_0} = \mathbb{E}(\boldsymbol{q}_{x_0}, \boldsymbol{Q}_{x_0}),\ \mathbb{P}_u(t) = \mathbb{E}(\boldsymbol{q}_u(t), \boldsymbol{Q}_u(t))$$

where $\boldsymbol{q}_{x_0} \in \mathbb{R}^{n_s}$ and $\boldsymbol{Q}_{x_0} \in \mathbb{R}^{n_s \times n_s}$ are the center and shape matrix of ellipsoidal initial state set, respectively. $\boldsymbol{q}_u(t) \in \mathbb{R}^{n_u}$ and $\boldsymbol{Q}_u(t) \in \mathbb{R}^{n_u \times n_u}$ are the center and shape matrix of ellipsoidal input set, correspondingly. Here $\mathbb{E}(\boldsymbol{q}, \boldsymbol{Q})$ represents an arbitrary ellipsoidal set.

$$\mathbb{E}(\boldsymbol{q}, \boldsymbol{Q}) = \{\boldsymbol{x} | (\boldsymbol{x} - \boldsymbol{q})^T \boldsymbol{Q}^{-1} (\boldsymbol{x} - \boldsymbol{q}) \leq 1)\}$$

The parameters of the reach set $(\mathbb{P}_x(t_k) = \mathbb{E}(\boldsymbol{q}_x(t_k), \boldsymbol{Q}_x(t_k)))$ at k-th time instance of interest for dynamic system (B.1) are given by [72, 73]

$$\boldsymbol{q}_x(t_k) = \Gamma(t_k) \tag{5.2}$$

$$\boldsymbol{Q}_x(t_k) = \boldsymbol{P}^{-1}(t_k) \tag{5.3}$$

where $\Gamma(t_k)$ is defined as

$$\Gamma(t_k) = \boldsymbol{\Phi}(t_k, t_0)\boldsymbol{q}_{x_0} + \int_{t_0}^{t_k} \boldsymbol{\Phi}(t_k, \tau)\boldsymbol{B}(\tau)\boldsymbol{q}_u(\tau)d\tau$$

$\boldsymbol{P}(t_k) \in \mathbb{R}^{n_s \times n_s}$ is the only solution of recursive equation $\boldsymbol{P}_{n+1} = \Lambda(\boldsymbol{P}_n)$

$$\Lambda(\boldsymbol{P}_n) = \left(\sqrt{Tr\left(\boldsymbol{P}_n \overline{\boldsymbol{Q}}_0^T\right)} + \int_{t_0}^{t_k} \sqrt{Tr\left(\boldsymbol{P}_n \bar{\boldsymbol{Q}}_u^T(\tau)\right)} d\tau \right.$$
$$\left. \frac{\bar{\boldsymbol{Q}}_0}{\sqrt{Tr\left(\boldsymbol{P}_n \bar{\boldsymbol{Q}}_0^T\right)}} + \int_{t_0}^{t_k} \frac{\overline{\boldsymbol{Q}}_u(\tau)}{\sqrt{Tr\left(\boldsymbol{P}_n \bar{\boldsymbol{Q}}_u^T(\tau)\right)}} d\tau \right)^{-1} \tag{5.4}$$

where $\bar{\boldsymbol{Q}}_0 \in \mathbb{R}^{n_s \times n_s}$ and $\bar{\boldsymbol{Q}}_u \in \mathbb{R}^{n_s \times n_s}$ are defined as

$$\bar{\boldsymbol{Q}}_0 = \boldsymbol{\Phi}(t_k, t_0)\boldsymbol{Q}_{x_0}\boldsymbol{\Phi}^T(t_k, t_0)$$
$$\bar{\boldsymbol{Q}}_u(\tau) = \boldsymbol{\Phi}(t_k, \tau)\boldsymbol{B}(\tau)\boldsymbol{Q}_u(\tau)\left(\boldsymbol{\Phi}(t_k, \tau)\boldsymbol{B}(\tau)\right)^T$$

Starting from an initial positive-definite matrix $\boldsymbol{P}_0$, the recursive equation converges to the unique solution [72].

Remark. *[73] Considering the initial state set approaches to a point ($\boldsymbol{Q}_{x_0} = \epsilon \boldsymbol{I}_{n_s}$ with $\epsilon \to 0$), (5.4) converges to*

$$\Lambda(\boldsymbol{P}_n) = \left(\int_{t_0}^{t_k} \sqrt{Tr\left(\boldsymbol{P}_n \bar{\boldsymbol{Q}}_u^T(\tau)\right)} d\tau \int_{t_0}^{t_k} \frac{\bar{\boldsymbol{Q}}_u(\tau)}{\sqrt{Tr\left(\boldsymbol{P}_n \bar{\boldsymbol{Q}}_u^T(\tau)\right)}} d\tau \right)^{-1} \tag{5.5}$$

The algorithm for calculating the parameters of ellipsoidal reach set is given in Algorithm 2. In this algorithm, the center of ellipsoidal reach set is calculated using (5.2). The shape matrix of ellipsoidal reach set is obtained from the recursion in (5.4), starting from $\boldsymbol{P}_0 = \boldsymbol{I}_{n_s}$ and using the termination condition

$$\mu\left(\boldsymbol{P}_{n+1}, \boldsymbol{P}_n\right) = \frac{\left|\left(det\boldsymbol{P}_{n+1}\right)^{-1/2} - \left(det\boldsymbol{P}_n\right)^{-1/2}\right|}{\left(det\boldsymbol{P}_n\right)^{-1/2}}$$

Here det and $|.|$ denote the determinant of matrix and the absolute value of scalar, respectively. The termination condition is satisfied when the normalized rate of change in the volume of ellipsoid is less than the desired tolerance ϵ.

Algorithm 2 Calculating parameters of ellipsoidal reach set

$\boldsymbol{q}_x(t_k) \leftarrow \Gamma(t_k)$
$\boldsymbol{P}_0 \leftarrow \boldsymbol{I}_{n_s}$
$n \leftarrow 0$
repeat
  $\boldsymbol{P}_{n+1} \leftarrow \Lambda(\boldsymbol{P}_n)$
  $n \leftarrow n+1$
until $\mu\left(\boldsymbol{P}_{n+1}, \boldsymbol{P}_n\right) < \epsilon$
$\boldsymbol{Q}_x(t_k) \leftarrow \boldsymbol{P}_{n+1}^{-1}$
return $\boldsymbol{q}_x(t_k), \boldsymbol{Q}_x(t_k)$

5.2.1 Potentially Unsafe Undesirable Set Detection

To find whether the undesirable set is potentially unsafe and can intersect with the reach set at each time instance of interest, the conservative intersection detection algorithm is presented, i.e., see Algorithm 3.

Intersection status between geometrical shapes is invariant under an affine transformation [55]. The intersection detection between each ellipsoidal reach set and polytopic undesirable set is equivalent to intersection detection between unit spherical reach set with center at origin ($\mathbb{P}_y = \{\boldsymbol{y} | \ \|\boldsymbol{y}\| \leq 1\}$) and transformed polytopic set ($\mathbb{P}_{y_u}(t_k) = \{\boldsymbol{y} | \boldsymbol{A}_{y_u}(t_k)\boldsymbol{y} \preceq \boldsymbol{b}_{y_u}(t_k)\}$). The affine transformation is given by

$$F_{t_k}(\boldsymbol{x}) = \boldsymbol{L}_x^{-1}(t_k)(\boldsymbol{x} - \boldsymbol{q}_x(t_k)) \tag{5.6}$$

where $\boldsymbol{L}_x(t_k) \in \mathbb{R}^{n_s \times n_s}$ is the Cholesky factorization of shape matrix of ellipsoid, $\boldsymbol{q}_x(t_k)$ is the center of ellipsoid, $\boldsymbol{A}_{y_u}(t_k) \in \mathbb{R}^{n_h \times n_s}$, $\boldsymbol{b}_{y_u}(t_k) \in \mathbb{R}^{n_h}$ are the parameters of transformed polytopic undesirable set, and n_h is the number of polytopic faces.

The over-approximation of the Minkowski sum of sphere and each polytope is

Algorithm 3 Conservative intersection detection

$ID \leftarrow False$
$\mathbb{P}_y \leftarrow \{F_{t_k}(\boldsymbol{x})|\ \boldsymbol{x} \in \mathbb{P}_x(t_k)\}$
$\mathbb{P}_{y_u}(t_k) \leftarrow \{F_{t_k}(\boldsymbol{x})|\ \boldsymbol{x} \in \mathbb{P}_{x_u}(t_k)\}$
$\mathbb{P}_{y_s}(t_k) \leftarrow F_s\left(\mathbb{P}_y + \mathbb{P}_{y_u}(t_k)\right)$
if $\mathbb{P}_{y_s}(t_k) \cap \{\mathbf{0}_{n_s}\} \neq \varnothing$ **then**
  $ID \leftarrow True$
end if
return $ID, \mathbb{P}_y, \mathbb{P}_{y_u}(t_k)$

defined as $\mathbb{P}_{y_s} = F_s(\mathbb{P}_y + \mathbb{P}_{y_u}(t_k))$ [100] with

$$F_s(\mathbb{P}_y + \mathbb{P}_{y_u}(t_k)) = \{\boldsymbol{y}|\boldsymbol{A}_{y_u}(t_k)\boldsymbol{y} \preceq \boldsymbol{b}_{y_u}(t_k) + \|\boldsymbol{A}_{y_u}(t_k)\|_\bullet\}$$

where $\|\boldsymbol{A}_{y_u}(t_k)\|_\bullet = \left[\|\boldsymbol{A}^{1,*}_{y_u}(t_k)\|_2,\ ...,\ \|\boldsymbol{A}^{n_h,*}_{y_u}(t_k)\|_2\right]^T$, $\boldsymbol{A}^{j,*}_{y_u}(t_k)$ is the normal vector of j-th face of polytopic undesirable set and $\|.\|_2$ denotes the euclidean norm. Now, the intersection detection between each spherical reach set and the polytopic undesirable set can be conservatively approximated by intersection detection between origin and over-approximation of the Minkowski sum of sphere and each polytope

$$\mathbb{P}_{y_s} \cap \mathbf{0}_{n_s} \neq \varnothing \Leftrightarrow \boldsymbol{b}_{\boldsymbol{y_u}}(t_k) + \|\boldsymbol{A}_{y_u}(t_k)\|_\bullet \succeq \mathbf{0}_{n_h}$$

where $\mathbf{0}_{n_s}$ and $\mathbf{0}_{n_h}$ are n_s-dimension and n_h-dimension zero vectors, respectively. Here $\varnothing$ denotes the empty set.

5.2.2 Finding an Optimal Separating Hyperplane

By approximating each reach set with the ellipsoid defined in (5.2) and (5.3), the objective in (5.1) is to maximize the inscribed volume between the ellipsoid and separating hyperplane, $V_c(t_k)$. The volume of geometrical shapes transformed by an affine function is proportional to the original volume [61]. Using this fact and the affine transformation in (5.6), the goal is to maximize the volume of spherical cap produced by the intersection of the unit hypersphere centered at the origin and the transformed separating hyperplane. This volume can be simply maximized by maximizing the distance of the separating hyperplane from the origin, $d_y(t_k)$. Therefore, the optimization problem in (5.1) is re-formulated as

$$\begin{aligned} \max_{\boldsymbol{\alpha}_y(t_k), \eta_y(t_k)} \quad & d_y(t_k) \\ \text{s.t.} \quad & \boldsymbol{\alpha}_y^T(t_k)\boldsymbol{y}_t(t_k) + \eta_y(t_k) \geq 0 \\ & \boldsymbol{\alpha}_y^T(t_k)\boldsymbol{v}_y^j(t_k) + \eta_y(t_k) \leq 0 \qquad j = 1,\ \dots,\ n_v \end{aligned} \tag{5.7}$$

where $d_y(t_k) = |\eta_y(t_k)|/\|\boldsymbol{\alpha}_y(t_k)\|$ is the distance of the separating hyperplane from the origin.

Two scenarios can occur given the position of the target point, origin, and undesirable set, as shown in Fig. 5.1. If a hyperplane could be found to separate the origin and target point form undesirable set (Fig. 5.1(a)), the distance of the separating hyperplane from the origin should be maximized. The parameters of the hyperplane are scaled by $\eta_y(t_k) > 0$, as $(\boldsymbol{\alpha}_y(t_k), 1)$. The objective function in (5.7) is given by

$d_y(t_k) = 1/(\boldsymbol{\alpha}_y(t_k)^T \boldsymbol{\alpha}_y(t_k))$. An equivalent optimization problem is derived as

$$\begin{aligned} \min_{\boldsymbol{\alpha}_y(t_k)} \quad & \boldsymbol{\alpha}_y^T(t_k)\boldsymbol{\alpha}_y(t_k) \\ \text{s.t.} \quad & \boldsymbol{\alpha}_y^T(t_k)\boldsymbol{y}_t(t_k) + 1 \geq 0 \\ & \boldsymbol{\alpha}_y^T(t_k)\boldsymbol{v}_y^j(t_i) + 1 \leq 0 \qquad j = 1,\ ...,\ n_v \end{aligned} \tag{5.8}$$

In this scenario, the inscribed volume is larger than the volume of hemi-sphere ($V_c > V_s/2$, where V_s is the volume of unit sphere).

The optimization problem in (5.8) would be infeasible if there is no hyperplane separating the origin and target point from the undesirable set. Given that the target point would not be in the undesirable set, either the origin is in the undesirable set, or the undesirable set is between the origin and target point (Fig. 5.1(b)). In such case, by assuming $\eta(t_k) \leq 0$ in (5.7), the optimization problem is formulated as finding the closest hyperplane to the origin that separates the target point from the origin and undesirable set. This optimization problem is non-convex and can have multiple optimal solutions. One obvious solution is the bounding hyperplane of the polytope with minimum distance to the origin, which is visible to the target point.

$$\boldsymbol{\alpha}_y(t_k), \eta_y(t_k) = \arg\ \min(d^1, d^2, ..., d^p) \tag{5.9}$$

where d^i is the distance of i-th (out of p) visible bounding hyperplane of the polytopic undesirable set

$$d^i = \frac{|b^i_{y_u}(t_k)|}{\|\boldsymbol{A}^{i,*}_{y_u}(t_k)\|}$$

where $\boldsymbol{A}_{y_u}^{i,*}(t_k) \in \mathbb{R}^{n_s}$ and $b_{y_u}^{i}(t_k) \in \mathbb{R}$ are the parameters of the i-th visible bounding hyperplane of polytope to target point.

$$\boldsymbol{A}_{y_u}^{i,*}(t_k)\boldsymbol{y}_t(t_k) + b_{y_u}^{i}(t_k) \geq 0$$

The separating hyperplane calculated using (5.9) would not necessarily yield the largest volume safe convex region, but it would generate the largest volume safe convex region around the visible faces of the polytopic set. The maximum inscribed volume in this scenario is smaller than the volume of hemi-hypersphere, $V_c \leq V_s/2$. Note that (5.9) can have more than one solution any of which could serve as the separating hyperplane.

The steps for finding the optimal separating hyperplane are outlined in Algorithm 4. First, the function OSH_a attempts to solve the optimization problem in (5.8). If $d_y^*(t_k) \leq 1$ where $d_y^*(t_k)$ is the optimal distance, the optimal solution yields the parameters of the separating hyperplane. When $d_y^*(t_k) > 1$, the reach set would not overlap with the undesirable set and hence there is no need for a separating hyperplane. In case the optimization problem is infeasible, then (5.9) is used to determine the parameters of the separating hyperplane. Finally, the parameters of separating hyperplane in the original coordinates are computed from the transformation in (5.6).

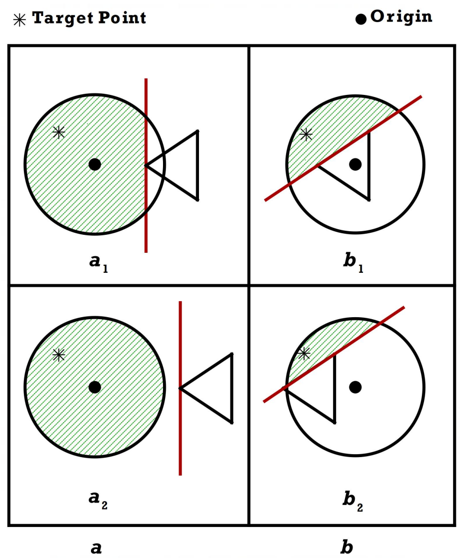

Figure 5.1: Generated separating line when a) the target point and origin (center of reach set) can be separated from undesirable set b) origin is in the undesirable set or the undesirable set is between origin and target point.

Algorithm 4 Steps for finding optimal separating hyperplane

---

if $OSH_a\left(\mathbb{P}_y(t_k), \mathbb{P}_{y_u}(t_k)\right)$ *is feasible* **then**
  $\boldsymbol{\alpha}_y(t_k), \eta_y(t_k) \leftarrow OSH_a\left(\mathbb{P}_y(t_k), \mathbb{P}_{y_u}(t_k)\right)$
  if $|\eta_y(t_k)| > \|\boldsymbol{\alpha}_y(t_k)\|$ **then**
    Collision does not occur
    $\boldsymbol{\alpha}_y(t_k) \leftarrow \mathbf{0}_{n_s \times 1}, \ \eta_y(t_k) = 0$
  end if
else
  $\boldsymbol{\alpha}_y(t_k), \eta_y(t_k) \leftarrow OSH_b\left(\mathbb{P}_y(t_k), \mathbb{P}_{y_u}(t_k)\right)$
end if
$\boldsymbol{\alpha}(t_k) \leftarrow (\boldsymbol{L}_x^T)^{-1}\boldsymbol{\alpha}_{\boldsymbol{y}}(t_k)$
$\eta(t_k) \leftarrow \eta_y(t_k) - \boldsymbol{\alpha}_{\boldsymbol{y}}(t_k)^T \boldsymbol{L}_x^{-1}\boldsymbol{q}_x(t_k)$
return $(\boldsymbol{\alpha}(t_k), \eta(t_k))$

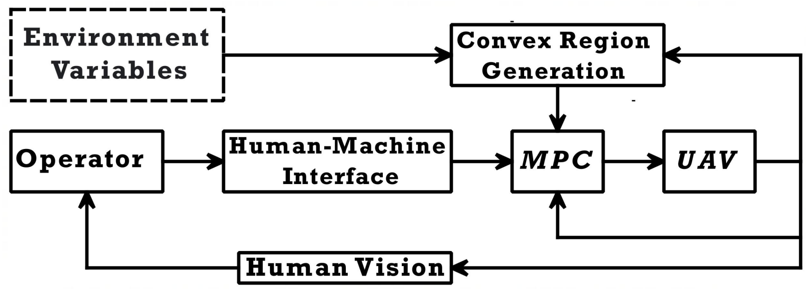

Figure 5.2: UAV teleoperation with collision avoidance assistance.

5.3 Safe Convex Region Formation in UAV Collision Avoidance

The proposed method for generating safe convex regions is employed for formulating a model-predictive controller for collision avoidance assistance in a UAV teleoperation application. A block diagram overview of this system is shown in Fig. 5.2. Here, the operator navigates the UAV by providing translational acceleration commands through a human-machine interface. A model predictive controller (MPC) modifies the operator's commands to prevent potential collisions with obstacles in the UAV task environment. The corrective commands are obtained by solving a convex optimization problem over a receding horizon.

5.3.1 System Dynamics

The dynamics of multi-rotor UAVs were given in Appendix A. Using the hierarchical control structure in Chapter 3, and assuming the closed-loop dynamics of the attitude and inner-loop controllers are fast so they can be ignored in the translational dynamics, the translational dynamics are simplified as (3.5 and 3.6)

$$\boldsymbol{p}_{cg} = \boldsymbol{v}_{cg} \tag{5.10}$$

$$\boldsymbol{v}_{cg} = \boldsymbol{a}_d \tag{5.11}$$

Let $\boldsymbol{x} = [\boldsymbol{p}_{cg}^T,\ \boldsymbol{v}_{cg}^T]^T$ and $\boldsymbol{u} = \boldsymbol{a}_d$. The translational dynamics of the UAV can be formulated in state-space form,

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u},\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{5.12}$$

where $\boldsymbol{x}_0 \in \mathbb{R}^6$ represents the vector of initial states, $\boldsymbol{A} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{B} \in \mathbb{R}^{6\times3}$ are derived as

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0}_{3\times3} & \boldsymbol{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix},\ \boldsymbol{B} = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \boldsymbol{I}_3 \end{bmatrix} \tag{5.13}$$

where $\boldsymbol{I}_3$ is 3×3 identity matrix and $\mathbf{0}_{3\times3}$ is 3×3 zero matrix.

Assuming constant input within the sampling time (T_s), the discrete-time dynamic model of continuous- time system is derived as

$$\boldsymbol{x}[k+1] = \boldsymbol{A}_d\boldsymbol{x}[k] + \boldsymbol{B}_d\boldsymbol{u}[k],\ \boldsymbol{x}[0] = \boldsymbol{x}_0 \tag{5.14}$$

where $\boldsymbol{A}_d \in \mathbb{R}^{6\times6}$ and $\boldsymbol{B}_d \in \mathbb{R}^{6\times3}$ are given by

$$\boldsymbol{A}_d = \begin{bmatrix} \boldsymbol{I}_3 & T_s\boldsymbol{I}_3 \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_3 \end{bmatrix},\ \boldsymbol{B}_d = \begin{bmatrix} T_s^2/2\boldsymbol{I}_3 \\ T_s\boldsymbol{I}_3 \end{bmatrix} \tag{5.15}$$

5.3.2 Collision Avoidance Algorithm

Collision avoidance assistance is provided by formulating and solving an optimization problem over a rolling horizon in a model-predictive control framework. The goal is to find a minimally interfering corrective command to modify the operator's command in order to avoid potential collisions with obstacles in the task space. The optimization problem is formulated as follows:

$$\min \sum_{k=0}^{N-1} (\boldsymbol{u}[k] - \boldsymbol{u}_h[k])^T \boldsymbol{P}[k](\boldsymbol{u}[k] - \boldsymbol{u}_h[k]) \tag{5.16a}$$

$$\text{s. t.} : \boldsymbol{x}[k+1] = \boldsymbol{A}_d \boldsymbol{x}[k] + \boldsymbol{B}_d \boldsymbol{u}[k] \tag{5.16b}$$

$$\boldsymbol{A}_u \boldsymbol{u}[k] \preceq \boldsymbol{b}_u \tag{5.16c}$$

$$\boldsymbol{A}_p[k] \boldsymbol{p}_{cg}[k] \preceq \boldsymbol{b}_p[k] \tag{5.16d}$$

where $\boldsymbol{u}_h \in \mathbb{R}^3$ contains the actual and predicted operator's commands over the control horizon, N is the number of time steps in the control horizon, and $\boldsymbol{P} = \boldsymbol{I}_3$. $\boldsymbol{A}_u \in \mathbb{R}^{n_u \times 3}$ and $\boldsymbol{b_u} \in \mathbb{R}^{n_u}$ are parameters of polytopic input set and n_u is the number of faces of polytope. The obstacle-free space is formulated in (5.16d) where $\boldsymbol{E}[k] \in \mathbb{R}^{n_x \times 6}$ and $\boldsymbol{h}[k] \in \mathbb{R}^{n_x}$ are the parameters of the polyhedral safe region and n_x is the number of hyperplanes creating the polyhedron.

5.3.3 Safe Convex Region Generation

The polyhedral safe convex region is obtained by the intersection of safe half-spaces corresponding to the separating hyperplanes between each undesirable set and the UAV. To find the separating hyperplanes, the input set should be modelled by an ellipsoid. Considering the time-independent polytopic shape of input set (5.16c) over the MPC time horizon, the parameters of the ellipsoidal input set are independent of time and can be approximated by minimum volume ellipsoid covering the polytope. This ellipsoid is called Lowner-John ellipsoid calculated as [17]

$$\begin{aligned} \min \quad & \log\det \boldsymbol{D}_{\boldsymbol{u}}^{-1} \\ \text{s.t.} \quad & ||\boldsymbol{D}_u \boldsymbol{v}_{u_i} + \boldsymbol{d}_u||_2 \leq 1, \qquad i = 1, ..., n_u \end{aligned} \tag{5.17}$$

where $\boldsymbol{D}_{\boldsymbol{u}} \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{d}_u \in \mathbb{R}^3$ are optimization variables, $\boldsymbol{v}_{u_i}$ is the i-th (out of n_u) vertex of polytope. The parameters of ellipsoidal input set is defined as

$$\begin{aligned} \boldsymbol{q}_u &= -\boldsymbol{D}_u^{-1}\boldsymbol{d}_u \\ \boldsymbol{Q}_u &= \left(\boldsymbol{D}_u^T \boldsymbol{D}_u\right)^{-1} \end{aligned}$$

Assuming the initial state set is represented by a point ($\boldsymbol{q}_{x_0} = \boldsymbol{x}_0,\ \boldsymbol{Q}_{x_0} = \epsilon \boldsymbol{I}_6$ with $\epsilon \to 0$) and considering the ellipsoidal input set is time-independent, the shape matrix of reach set at each sample can be calculated off-line using (5.3) where $\boldsymbol{P}$ calculated by (5.5). By defining $\boldsymbol{q}_x(t_{k+1}) = \boldsymbol{q}_x[k+1]$, the center of reach set (5.2) can be represented by discrete-time dynamic model

$$\boldsymbol{q}_x[k+1] = \boldsymbol{A}_d \boldsymbol{q}_x[k] + \boldsymbol{B}_d \boldsymbol{q}_u,\ \ \boldsymbol{q}_x[0] = \boldsymbol{q}_{x_0}$$

In the context of collision avoidance, undesirable sets are 3-dimensional objects. Reaching these sets with any feasible velocity is undesirable. The 6-dimensional ellipsoidal reach set must be projected onto the 3-dimensional position coordinates, which then can be used to generate the approximate convex safe region. This projection is also an ellipsoid calculated as (proof in appendix C.3)

$$\begin{aligned}\boldsymbol{q}_r &= \boldsymbol{q}_p \\ \boldsymbol{Q}_r &= \left(\boldsymbol{U}_p - \boldsymbol{U}_{pv}^T \boldsymbol{U}_v^{-1} \boldsymbol{U}_{pv}\right)^{-1}\end{aligned}$$

where $\boldsymbol{q}_p \in \mathbb{R}^3$ is the position subvector of $\boldsymbol{q}_x$ and $\boldsymbol{U}_p,\ \boldsymbol{U}_{pv},\ \boldsymbol{U}_v \in \mathbb{R}^{3\times 3}$ are submatrices of $\boldsymbol{Q}_x^{-1}$.

$$\begin{bmatrix}\boldsymbol{U}_p & \boldsymbol{U}_{pv}^T \\ \boldsymbol{U}_{pv} & \boldsymbol{U}_v\end{bmatrix} = \boldsymbol{Q}_x^{-1}$$

To calculate potentially unsafe undesirable sets, the intersection of the projected ellipsoid with undesirable sets at each sample is examined using the CID in Algorithm 3. By considering polytopic obstacles, a safe minimum distance of the UAV to the obstacle, and modelling the UAV as a sphere with center at $\boldsymbol{p}_{cg}$ and radius R_u, the undesirable set for each obstacle is obtained as overestimation of Minkowski sum of the inflated polytope and sphere [100]

$$\boldsymbol{A}_{x_u}^i \boldsymbol{x} \preceq \boldsymbol{b}_{x_u}^i$$

where $\boldsymbol{A}_{x_u}^i \in \mathbb{R}^{n_i\times 3}$ and $\boldsymbol{b}_{x_u}^i \in \mathbb{R}^{n_i}$ are the parameters of i-th undesirable set and n_i is the number of faces of i-th polytopic undesirable set.

$$\boldsymbol{A}_{x_u}^i = \boldsymbol{A}_{o_i},\ \boldsymbol{b}_{x_u}^i = \boldsymbol{b}_{o_i} + (R_u + d_s)\|\boldsymbol{A}_{o_i}\|_{\bullet}$$

Here d_s is the safe minimum distance between the UAV and each obstacle, $\boldsymbol{A}_{o_i} \in \mathbb{R}^{n_i\times 3}$

and $\boldsymbol{b}_{o_i} \in \mathbb{R}^{n_i}$ are the parameters of the i-th polytopic obstacle

$$\|\boldsymbol{A}_{o_i}\|_{\bullet} = \left[\|\boldsymbol{A}_{o_i}^{1,*}\|_2,\ \ldots,\ \|\boldsymbol{A}_{o_i}^{n_i,*}\|_2\right]^T$$

where $\boldsymbol{A}_{o_i}^{j,*}$ is the normal vector of the j-th face of the i-th polytopic obstacle.

Algorithm 4 requires a target point at each sample time to compute the separating hyperplane. The target point is normally the predicted position of the UAV. If this predicted position falls inside the undesirable set, the closest point to the predicted position which is inside the reach set and outside the undesirable set is chosen.

5.4 Experiment

A block diagram of the experimental setup is shown in Fig. 4.3. The overall update rate for the MPC-based collision avoidance algorithm is $f_s = 40\ Hz$ and the time horizon for the MPC algorithm is set to $t_f = 1.5\ sec$. The proposed collision avoidance assistance algorithm must solve several optimization problems. In particular, the separating hyperplanes are obtained from solving the optimization problem in (5.8). This optimization is formulated as a quadratic program and is solved in real-time by using the OSQP solver with its code generation [9, 87]. The real-time MPC collision avoidance algorithm is formulated as a quadratic program; this is also solved by the OSQP solver. The minimum volume ellipsoid covering the polytopic input set (5.17) is formulated as semi-definite program [13] and is solved off-line using the Mosek solver [5].

The operator's acceleration command is assumed constant over the MPC time horizon. A two-dimensional representation of obstacles is shown in Fig. 5.3. Each obstacle is assumed to be one meter high. The safe minimum distance of the UAV and obstacles is set to $d = 10\ cm$. The parameters of polytopic input set are set to

$$\boldsymbol{A}_u = [-\boldsymbol{I}_3, \boldsymbol{I}_3]^T, \ \boldsymbol{b}_u = [2, 0.5, 2, 2, 0.5, 2]^T$$

5.4.1 Flight Test

Two experiments are carried out to demonstrate the performance of proposed algorithms in this paper. The first experiment investigates the effectiveness of the safe convex region generation algorithm by comparing two scenarios. In the first scenario, the SVM algorithm [46] is used to generate the separating hyperplane between the undesirable sets and the UAV. To this end, the UAV is modelled as a point and the separating hyperplanes are assumed to touch the boundary of of the undesirable sets, to create the maximum possible volume. Therefore, the SVM is formulated as

$$\begin{aligned}
\min_{\boldsymbol{\alpha}_s^i,\ \eta_s^i} \quad & {\boldsymbol{\alpha}_s^i}^T \boldsymbol{\alpha}_s^i \\
\text{s.t.} \quad & {\boldsymbol{\alpha}_s^i}^T \boldsymbol{p}_{cg} + \eta_s^i \geq 1 \\
& {\boldsymbol{\alpha}_s^i}^T \boldsymbol{v}_j^i + \eta_s^i \leq 0 \qquad j = 1, ..., n_i
\end{aligned}$$

where $\boldsymbol{\alpha}_s^i \in \mathbb{R}^3$ and η_s^i are the parameters of the i-th plane, which separates the i-th undesirable set from the UAV, $\boldsymbol{v}_j^i \in \mathbb{R}^3$ is j-th vertex (out of n_i) of i-th polytopic undesirable set.

The trajectory of the UAV in this scenario is displayed in Fig. 5.3. All obstacles (undesirable sets) are considered in generating the safe convex region, with a corresponding separating hyperplane, i.e., see Fig. 5.3(a-d). These planes serve as constraints in the MPC optimization, effectively blocking the UAV path. This in turn leads to undesirable corrective commands by the collision avoidance assistance algorithm as seen in Fig. 5.5(a).

The second scenario employs the proposed safe convex region generation algorithm. The trajectory of the UAV in this scenario is depicted in Fig. 5.4. Two-dimensional representation of the reach sets and the separating planes at different sample times and UAV positions are presented in Fig. 5.4(a-d). As evident in Fig. 5.5(a), the use of the new algorithm in finding unsafe undesirable sets and creating safe convex region significantly improves the MPC performance, and eliminates some of the undesirable corrective commands seen in the previous scenario. The operator's commands for the first and second scenarios are displayed in Fig. 5.5(b) and Fig. 5.5(c), respectively. In the second experiment, the path in front of the UAV is blocked by moving the second pair of the undesirable sets in Y-direction. As seen in Fig. 5.6, the corrective commands in Fig. 5.7(a) help modify the UAV trajectory and assist the operator in guiding it through the opening between obstacles. The corrective commands achieve this while attempting to minimize interference with the operator's commands as much as possible, i.e. see Fig. 5.7(b). A minimum safe distance is maintained between the UAV trajectory and the obstacles; the actual obstacle is denoted by the dashed trapezoid. The reach sets and separating planes at different sample times and UAV positions are depicted in Fig. 5.6(a-d). Separating planes with the maximum safe region are created for each reachable undesirable set at each sample time. This yields a less restrictive convex region approximation, eliminating some of the unnecessary corrective commands that would have otherwise been generated.

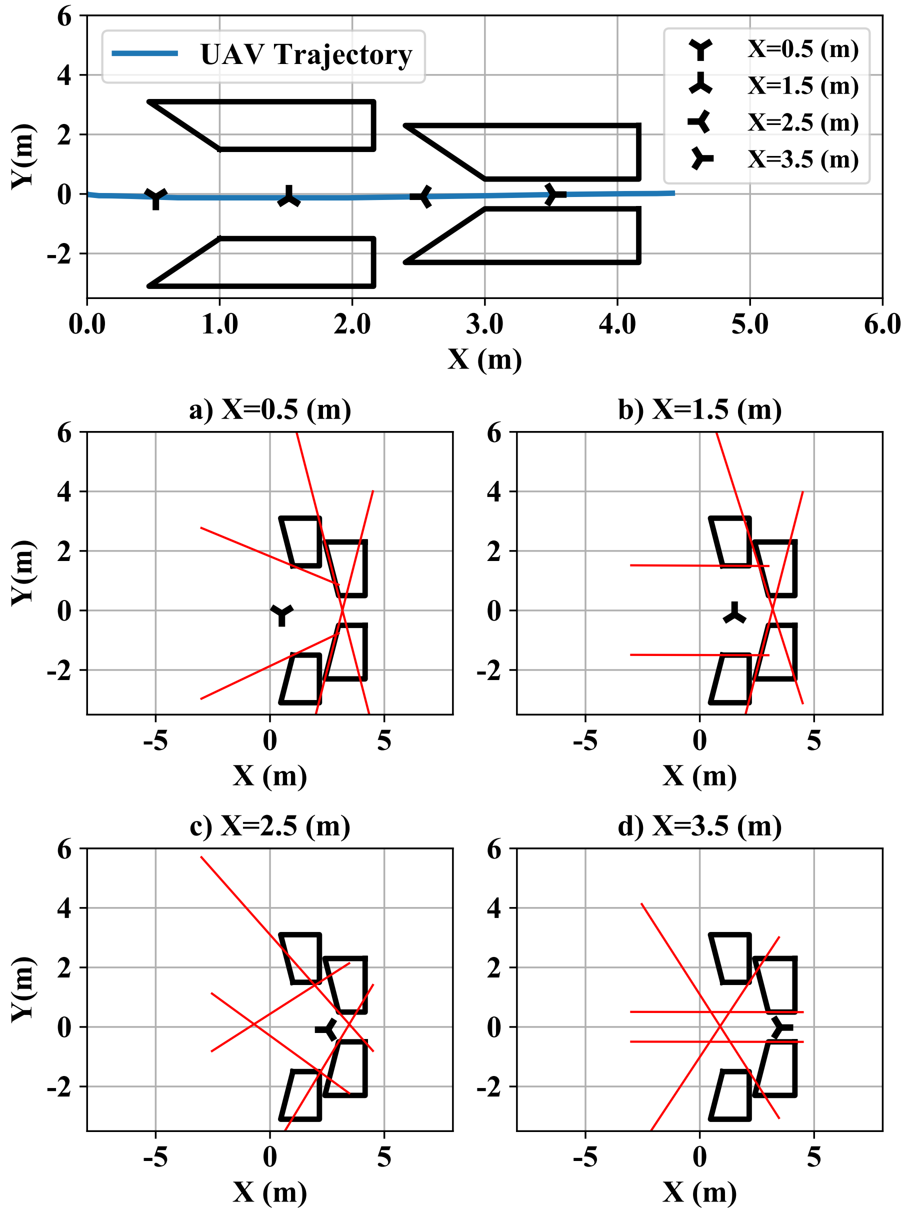

Figure 5.3: Two-dimensional representation of the UAV trajectory in a scenario where SVM algorithm is used for generating separating hyperplanes; two-dimensional representation of separating planes a) at X=0.5 (m), b) at X=1.5 (m), c) at X=2.5 (m), d) at X=3.5 (m).

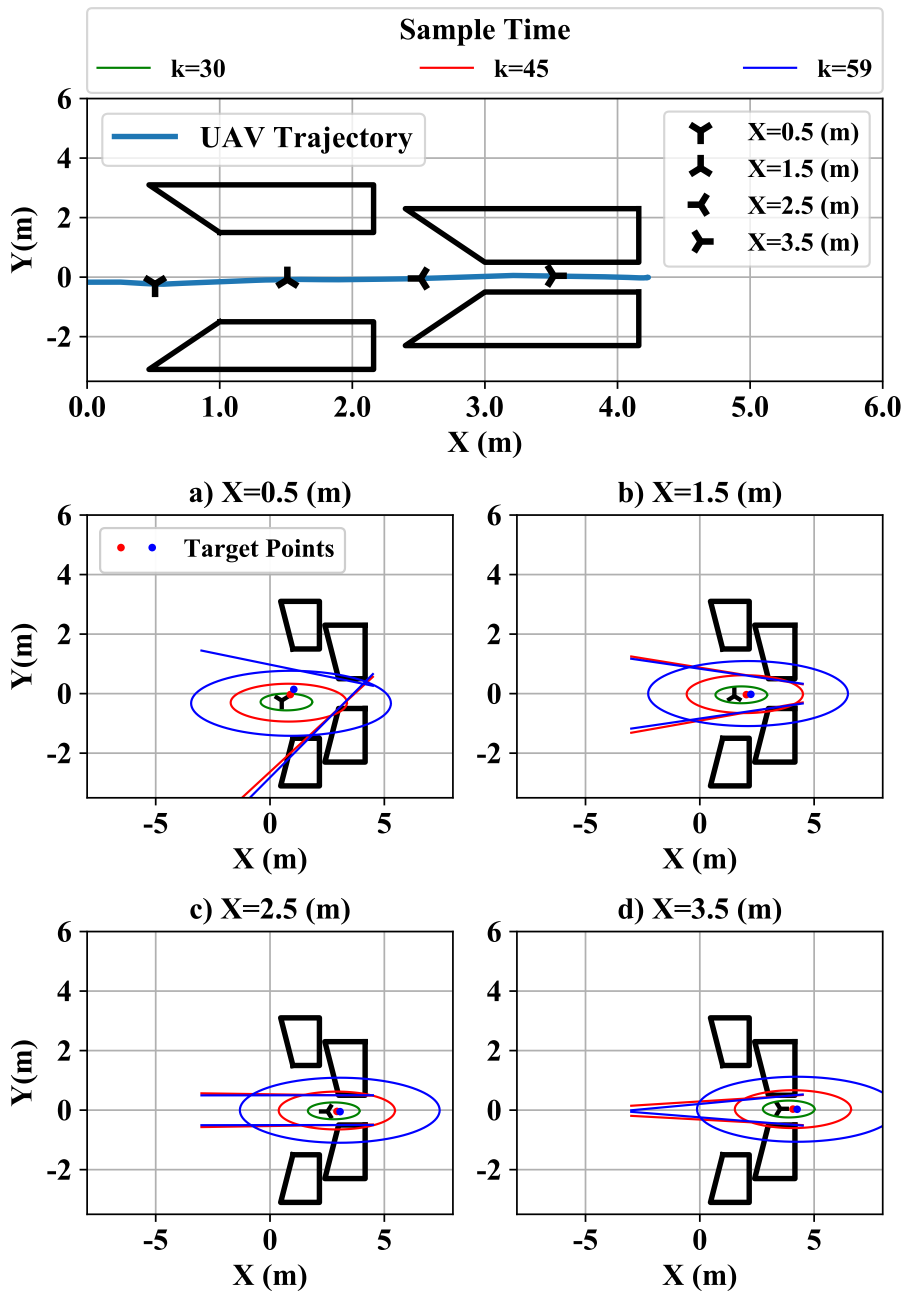

Figure 5.4: Two-dimensional representation of the UAV trajectory using the proposed algorithm for generating safe convex region. Two-dimensional representation of reach sets, target points and separating planes a) at X=0.5 (m), b) at X=1.5 (m), c) at X=2.5 (m), d) at X=3.5 (m).

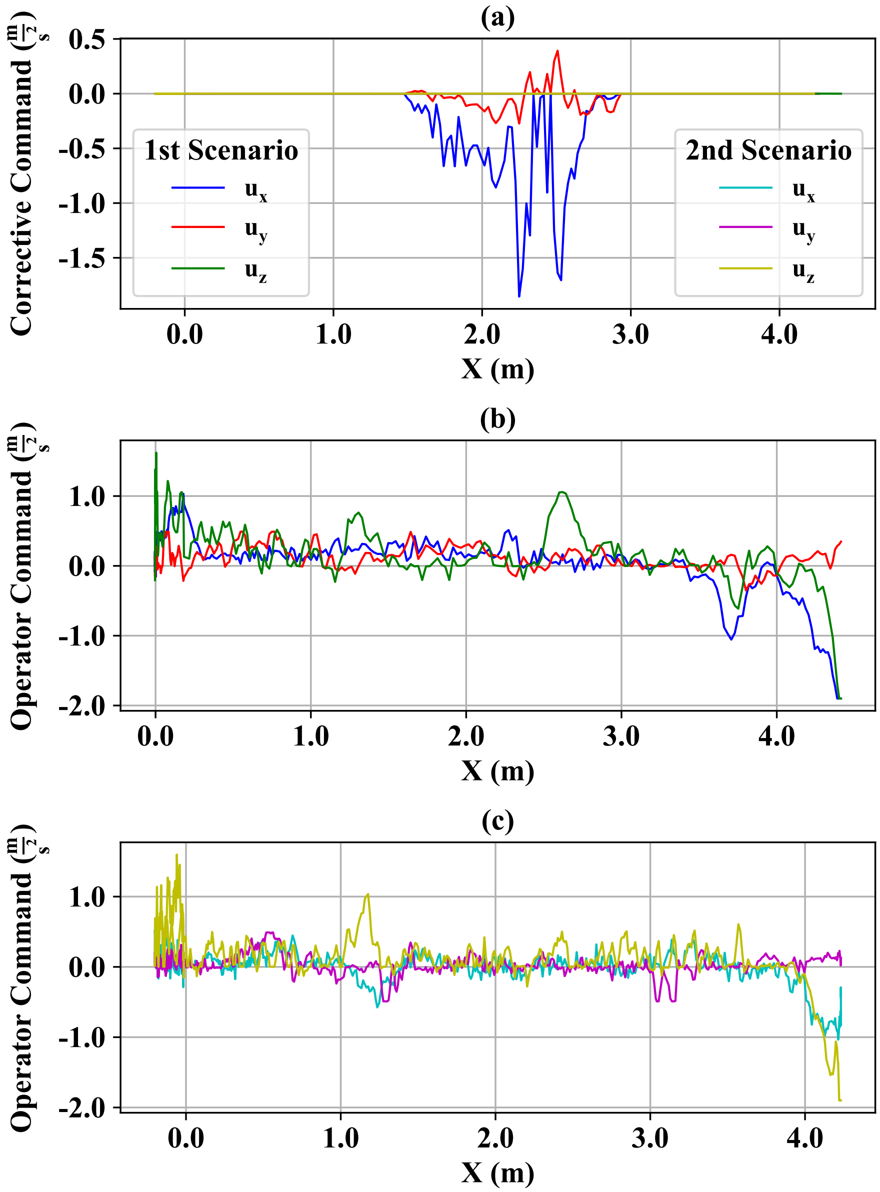

Figure 5.5: a) Corrective commands in the scenarios using SVM and the new algorithm for creating safe convex region. b) Operator's commands in the scenario using SVM. c) Operator's commands in the scenario using the new algorithm.

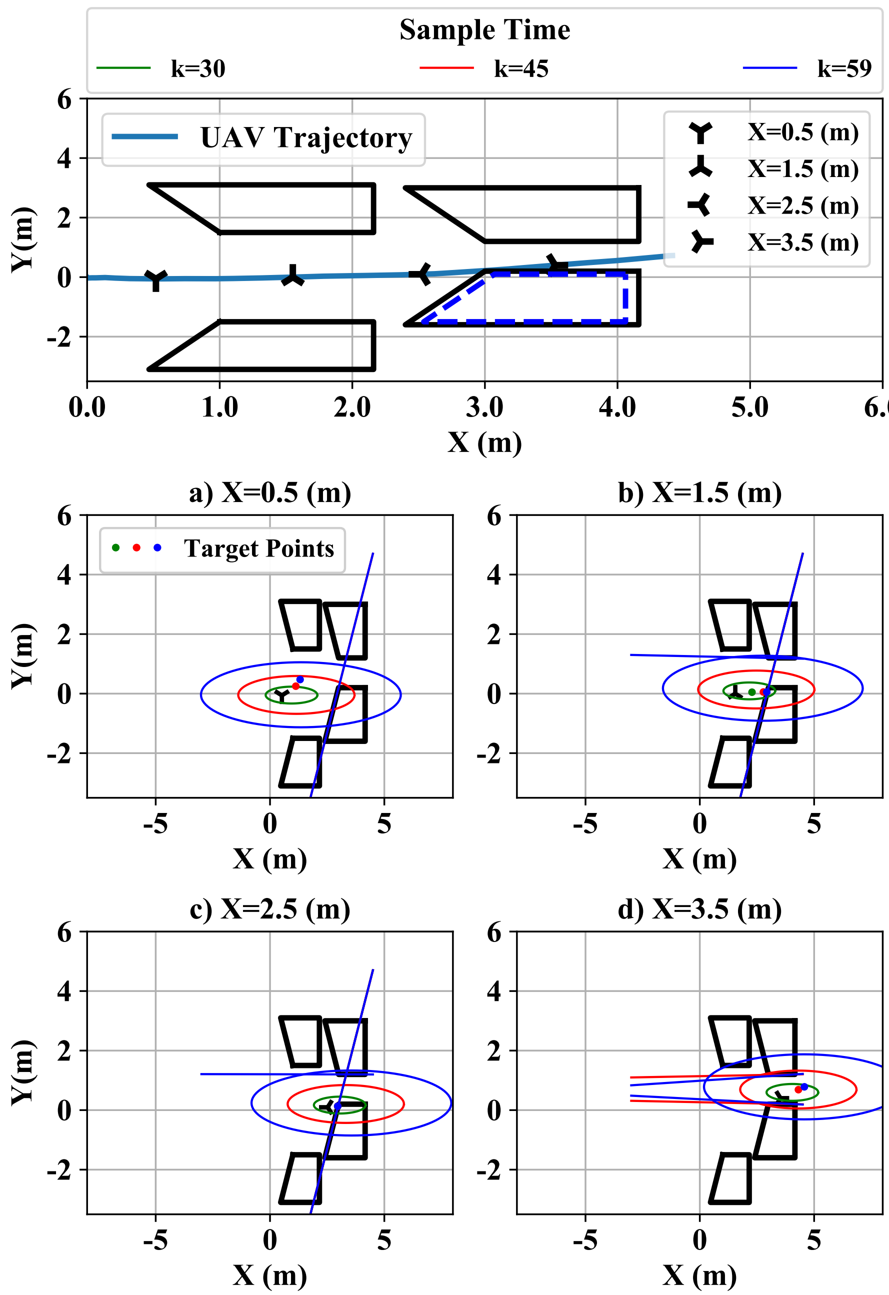

Figure 5.6: Two-dimensional representation of the UAV trajectory using the new algorithm for generating safe convex region. Two-dimensional representation of reach sets, target points and separating planes a) at X=0.5 (m), b) at X=1.5 (m), c) at X=2.5 (m), d) at X=3.5 (m).

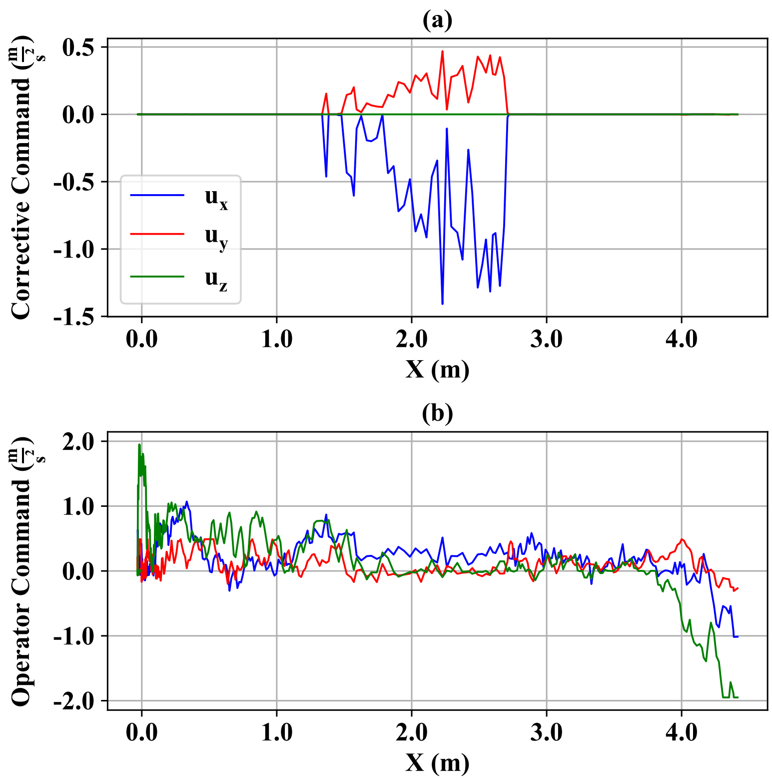

Figure 5.7: a) Corrective commands generated by the proposed collision avoidance assistance algorithm. b) Operator's commands.

Chapter 6

Collision Avoidance in the Presence of Uncertainty

In real-world scenarios, uncertainties are unavoidable and can significantly degrade the performance of collision avoidance algorithms. In the presence of uncertainties, the optimization problem may become infeasible resulting in collisions. Robust model predictive control (RMPC) has been developed to deal with uncertainties in the predictive control of dynamical systems. Worst case MPC is a popular approach in this category. it ensures constraints satisfaction for all possible uncertainties. However, the formulation of this approach as a convex optimization problem can be challenging.

In this chapter, a novel ellipsoidal-based RMPC for collision avoidance assistance in UAV teleoperation is presented. The main contribution of this work is the formulation of the collision avoidance assistance in UAV as a convex optimization problem in the presence of uncertainties. These uncertainties are in the form of disturbances and measurement noises affecting the UAV and obstacles. In the proposed approach, the operator tele-operates the UAV by providing linear acceleration commands while an ellipsoidal-based RMPC modifies these commands to ensure collision-free operation in the presence of uncertainties. Collision avoidance is formulated as a convex optimization problem in three steps. First, ellipsoidal approximation of the reachable regions due to admissible inputs and uncertainties are calculated using reachability analysis. Afterwards, a polyhedral approximation of the obstacle-free space is derived using the safe convex region generation method presented in Chapter 7. Finally, an inner polyhedral approximation of the tightened constraints is obtained using geometrical relation between the ellipsoidal reach set due to uncertainties and polyhedral safe region.

6.1 System Dynamics

The dynamic equations of multi-rotor UAVs are expressed in Appendix A. Using the hierarchical control structure in Chapter 3, it is assumed the attitude and inner-loop controllers are tuned such that their closed-loop dynamics can be ignored in the translational dynamics. Therefore, the linear translational dynamics are simplified as

$$\dot{\boldsymbol{p}}_{cg} = \boldsymbol{v}_{cg} \tag{6.1}$$

$$\dot{\boldsymbol{v}}_{cg} = \boldsymbol{a} + \boldsymbol{a}_d \tag{6.2}$$

where $\boldsymbol{p}_{cg},\ \boldsymbol{v}_{cg} \in \mathbb{R}^3$ denote the position and velocity of the center of gravity of UAV, respectively. $\boldsymbol{a},\ \boldsymbol{a}_d \in \mathbb{R}^3$ are the input acceleration and disturbance, respectively.

Defining $\boldsymbol{x} = [\boldsymbol{p}_{cg}^T,\ \boldsymbol{v}_{cg}^T]^T$, $\boldsymbol{u} = \boldsymbol{a}$, $\boldsymbol{w} = \boldsymbol{a}_d$, the translational dynamics of the UAV is formulated in state-space form.

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{D}\boldsymbol{w}(t),\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{6.3}$$

$\boldsymbol{A} \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B} \in \mathbb{R}^{n_x \times n_u}$, and $\boldsymbol{D} \in \mathbb{R}^{n_x \times n_w}$ are defined as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix},\ \boldsymbol{B} = \boldsymbol{D} = \begin{bmatrix} \boldsymbol{0}_{3\times3} \\ \boldsymbol{I}_3 \end{bmatrix} \tag{6.4}$$

Here $n_x = 6$ and $n_u = n_w = 3$. $\boldsymbol{u}(t) \in \mathbb{U}$ with $\mathbb{U} = \mathbb{P}(\boldsymbol{A}_u, \boldsymbol{b}_u)$ is the polytopic input set containing all admissible inputs, $\boldsymbol{x}_0 \in \mathbb{X}_0$ with $\mathbb{X}_0 = \mathbb{E}(\boldsymbol{q}_{x_0}, \boldsymbol{Q}_{x_0})$ is the ellipsoidal initial state set containing all possible initial states due to measurement noise, and $\boldsymbol{w}(t) \in \mathbb{W}$ with $\mathbb{W} = \mathbb{E}(\boldsymbol{q}_w, \boldsymbol{Q}_w)$ is the ellipsoidal disturbance set containing all possible disturbances. Here $\mathbb{P}(\boldsymbol{A}, \boldsymbol{b})$ and $\mathbb{E}(\boldsymbol{q}, \boldsymbol{Q})$ represent arbitrary polytopic set and ellipsoidal set, respectively.

$$\mathbb{P}(\boldsymbol{A}, \boldsymbol{b}) = \{\boldsymbol{x} | \boldsymbol{A}\boldsymbol{x} \preceq \boldsymbol{b}\},\ \mathbb{E}(\boldsymbol{q}, \boldsymbol{Q}) = \{\boldsymbol{x} | (\boldsymbol{x} - \boldsymbol{q})^T \boldsymbol{Q}^{-1} (\boldsymbol{x} - \boldsymbol{q}) \leq 1)\} \tag{6.5}$$

The continuous-time dynamics are converted to discrete-time using zero-order-hold assumption with the sampling time T_s,

$$\boldsymbol{x}[k+1] = \boldsymbol{A}_d \boldsymbol{x}[k] + \boldsymbol{B}_d \boldsymbol{u}[k] + \boldsymbol{D}_d \boldsymbol{w}[k],\ \boldsymbol{x}[0] = \boldsymbol{x}_0 \tag{6.6}$$

6.2 Collision Avoidance Assistance in UAV under Uncertainty

A schematic view of the collision avoidance assistance is presented in Fig. 6.1. In this algorithm, the operator navigates the UAV using translational acceleration commands while collision avoidance assistance modifies the operator commands to ensure collision-free operation in the presence of uncertainties. To address uncertainties in problem formulation, the worst-case RMPC approach is used. In this method, the collision-free operation is ensured for all possible uncertainties.

The nominal dynamic equation of the UAV is defined as

$$\bar{\boldsymbol{x}}[k+1] = \boldsymbol{A}_d\bar{\boldsymbol{x}}[k] + \boldsymbol{B}_d\boldsymbol{u}[k],\ \bar{\boldsymbol{x}}[0] = \bar{\boldsymbol{x}}_0 \tag{6.7}$$

where $\bar{\boldsymbol{x}}_0 \in \mathbb{R}^{n_x}$ is the initial states of nominal system. Defining state deviation due to uncertainties as $\boldsymbol{x}_e[k] = \boldsymbol{x}[k] - \bar{\boldsymbol{x}}[k]$, the state deviation dynamic is formulated by

$$\boldsymbol{x}_e[k+1] = \boldsymbol{A}_d\boldsymbol{x}_e[k] + \boldsymbol{D}_d\boldsymbol{w}[k],\ \boldsymbol{x}_e[0] = \boldsymbol{x}_{e_0} \tag{6.8}$$

where $\boldsymbol{x}_{e_0} \in \mathbb{X}_0 \ominus \bar{\boldsymbol{x}}_0$ and $\boldsymbol{x}_e[k] \in \mathbb{X}_e[k]$ with $\mathbb{X}_e[k] \subset \mathbb{R}^{n_x}$ is the set of all possible state deviation due to uncertainties. Here $\ominus$ denotes the Pontryagin difference operation.

To ensure collision-free operation, the UAV should be in safe obstacle-free region for all possible uncertainties, i.e., $\boldsymbol{x}[k] \in \mathbb{X}_s[k]$ where $\mathbb{X}_s[k]$ is the safe region. Given $\boldsymbol{x}[k] \in \mathbb{X}_e[k] \oplus \bar{\boldsymbol{x}}[k]$, the collision-free motion is ensured if nominal states are in the tightened safe region, i.e, $\bar{\boldsymbol{x}}[k] \in \mathbb{X}_s[k] \ominus \mathbb{X}_e[k]$.

To ensure the UAV follows the operator commands as close as possible while avoiding the collision for all possible uncertainties, discrete-time RMPC is formulated as

$$\min_{\boldsymbol{u}[0,N-1]} \sum_{k=0}^{N-1} (\boldsymbol{u}[k] - \boldsymbol{u}_h[k])^T \boldsymbol{R}_u (\boldsymbol{u}[k] - \boldsymbol{u}_h[k]) \tag{6.9a}$$

$$\text{s. t.} : \bar{\boldsymbol{x}}[k+1] = \boldsymbol{A}_d[k]\bar{\boldsymbol{x}}[k] + \boldsymbol{B}_d[k]\boldsymbol{u}[k] \tag{6.9b}$$

$$\bar{\boldsymbol{x}} \in \mathbb{X}_s[k] \ominus \mathbb{X}_e[k] \tag{6.9c}$$

$$\boldsymbol{u}[k] \in \mathbb{U} \tag{6.9d}$$

where $\boldsymbol{u}[k], \boldsymbol{u}_h[k] \in \mathbb{R}^{n_u}$ are the UAV inputs and predicted operator's commands, respectively. $\boldsymbol{x}[k] \in \mathbb{R}^{n_x}$ is UAV states, $\boldsymbol{R}_u = \boldsymbol{I}_{n_u}$, and N is the number of samples within the MPC time horizon. Constraints related to system dynamics are represented in (6.9b). Tightened state constraints ensuring collision-free motion for all possible uncertainties are presented in (6.9c). Input constrains are formulated in (6.9d).

To solve the optimization-based collision avoidance assistance and find the global optimum solution in real-time, state constraints should be formulated by linear inequality constraints. First, reachable states of the system (6.6) and (6.8) at each sample are approximated by minimum volume enclosing ellipsoid. Next, each obstacle is inflated due to uncertainties, and obstacle-free region is approximated by polyhedral using the safe convex approximation region introduced in Chapter 7. Finally, the inner polytopic approximations of the tightened state constraints are calculated.

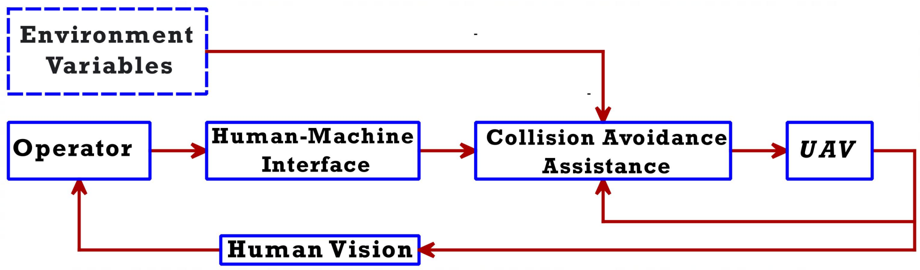

Figure 6.1: UAV teleoperation with automated collision avoidance assistance

6.3 Reach Set Approximation

Consider the continuous-time linear dynamic system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}(t), \ \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{6.10}$$

where $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_x}$ and $\boldsymbol{B} \in \mathbb{R}^{n_x \times n_u}$. $\boldsymbol{u} \in \mathbb{E}(\boldsymbol{q}_u, \boldsymbol{Q}_u)$ with $\mathbb{E}(\boldsymbol{q}_u, \boldsymbol{Q}_u)$ is the ellipsoidal input set with the center $\boldsymbol{q}_u \in \mathbb{R}^{n_u}$ and the shape matrix $\boldsymbol{Q}_u \in \mathbb{R}^{n_u \times n_u}$. $\boldsymbol{x}_0 \in \mathbb{E}(\boldsymbol{q}_{x_0}, \boldsymbol{Q}_{x_0})$ with $\mathbb{E}(\boldsymbol{q}_{x_0}, \boldsymbol{Q}_{x_0})$ being the ellipsoidal initial state set with the center $\boldsymbol{q}_{x_0} \in \mathbb{R}^{n_x}$ and the shape matrix $\boldsymbol{Q}_{x_0} \in \mathbb{R}^{n_x \times n_x}$. The parameters of minimum volume ellipsoid $(\mathbb{E}(\boldsymbol{q}_x(t), \boldsymbol{Q}_{\boldsymbol{x}}\boldsymbol{(t)}))$ enclosing the reach set for this system are given in (B.18) and (B.19).

The continuous-time dynamics are converted to discrete-time using zero-order-hold assumption with the sampling time T_s,

$$\boldsymbol{x}[k+1] = \boldsymbol{A}_d\boldsymbol{x}[k] + \boldsymbol{B}_d\boldsymbol{u}[k], \ \boldsymbol{x}[0] = \boldsymbol{x}_0 \tag{6.11}$$

The parameters of the minimum volume ellipsoid $(\mathbb{E}(\boldsymbol{q}_x(t), \boldsymbol{Q}_{\boldsymbol{x}}\boldsymbol{(t)}))$ enclosing the reach set $\mathbb{X}(t)$ for continuous-time linear system (6.10) are expressed in (B.18) and

(B.19). Using zero-order hold assumption in (B.18) and solving (B.19) at each sample time, the parameters of the minimum volume ellipsoid ($\mathbb{E}(\boldsymbol{q}_x[k], \boldsymbol{Q})_{\boldsymbol{x}}[\boldsymbol{k}]$) enclosing the reach set for discrete-time linear system (6.11) are derived as

$$\boldsymbol{q}_x[k+1] = \boldsymbol{A}_d\boldsymbol{q}_x[k] + \boldsymbol{B}_d\boldsymbol{q}_u[k], \;\; \boldsymbol{q}[0] = \boldsymbol{0}_{n_x} \tag{6.12}$$

$$\boldsymbol{Q}_x[k+1] = \boldsymbol{P}^{-1}[k+1] \tag{6.13}$$

where $\boldsymbol{P}[k+1] \in \mathbb{R}^{n_x \times n_x}$ is the only solution of recursive equation $\boldsymbol{P}_{n+1} = \Lambda(\boldsymbol{P}_n)$

$$\Lambda(\boldsymbol{P}_n) = \left(\sqrt{Tr\left(\boldsymbol{P}_n\bar{\boldsymbol{Q}}_0^T\right)} + \int_{t_0}^{t_{k+1}} \sqrt{Tr\left(\boldsymbol{P}_n\bar{\boldsymbol{Q}}_u^T(\tau)\right)}d\tau \right. \\ \left. \frac{\bar{\boldsymbol{Q}}_0}{\sqrt{Tr\left(\boldsymbol{P}_n\bar{\boldsymbol{Q}}_0^T\right)}} + \int_{t_0}^{t_{k+1}} \frac{\bar{\boldsymbol{Q}}_u(\tau)}{\sqrt{Tr\left(\boldsymbol{P}_n\bar{\boldsymbol{Q}}_u^T(\tau)\right)}}d\tau \right)^{-1} \tag{6.14}$$

where t_{k+1} is the time instance corresponding to $(k+1)$-th sample. $\bar{\boldsymbol{Q}}_0 \in \mathbb{R}^{n_x \times n_x}$ and $\bar{\boldsymbol{Q}}_u(\tau) \in \mathbb{R}^{n_x \times n_x}$ are defined as

$$\bar{\boldsymbol{Q}}_0 = \boldsymbol{\Phi}(t_{k+1}, t_0)\boldsymbol{Q}_{x_0}\boldsymbol{\Phi}^T(t_{k+1}, t_0) \tag{6.15}$$

$$\bar{\boldsymbol{Q}}_u(\tau) = \boldsymbol{\Phi}(t_{k+1}, \tau)\boldsymbol{B}(\tau)\boldsymbol{Q}_u(\tau)\left(\boldsymbol{\Phi}(t_{k+1}, \tau)\boldsymbol{B}(\tau)\right)^T \tag{6.16}$$

where $\boldsymbol{\Phi}(t_{k+1}, \tau) \in \mathbb{R}^{n_x \times n_x}$ is the state-transition matrix of system (6.10).

Starting from an initial positive-definite matrix $\boldsymbol{P}_0$, the recursive equation (6.14) converges to a unique solution [72]. Here it is assumed $\boldsymbol{P}_0 = \boldsymbol{I}_{n_x}$ and the termination condition for (6.14) is defined as

$$\mu(\boldsymbol{P}_{n+1}, \boldsymbol{P}_n) = \frac{|det\boldsymbol{P}_{n+1})^{-1/2} - (det\boldsymbol{P}_n)^{-1/2}|}{(det\boldsymbol{P}_n)^{-1/2}} \tag{6.17}$$

Here *det* and $|.|$ denote the determinant of matrix and the absolute value of scalar, respectively. The termination condition is satisfied when the normalized rate of change in the volume of ellipsoid is less than the desired tolerance.

Remark. *[73] Considering the initial state set approaches to a point ($\boldsymbol{Q}_{x_0} = \epsilon \boldsymbol{I}_{n_s}$ with $\epsilon \to 0$), (6.14) converges to*

$$\Lambda(\boldsymbol{P}_n) = \left(\int_{t_0}^{t_{k+1}} \sqrt{Tr\left(\boldsymbol{P}_n \bar{\boldsymbol{Q}}_u^T(\tau)\right)} d\tau \int_{t_0}^{t_{k+1}} \frac{\bar{\boldsymbol{Q}}_u(\tau)}{\sqrt{Tr\left(\boldsymbol{P}_n \bar{\boldsymbol{Q}}_u^T(\tau)\right)}} d\tau \right)^{-1} \tag{6.18}$$

To find minimum volume ellipsoid enclosing the reach set of the UAV, the linear dynamic equations of the UAV in (6.6) are formulated as

$$\boldsymbol{x}[k+1] = \boldsymbol{A}_d \boldsymbol{x}[k] + \boldsymbol{u}_t[k] \tag{6.19}$$

where $\boldsymbol{u}_t[k] \in \mathbb{U}_t[k]$ is the total input with $\boldsymbol{u}_t[k] = \boldsymbol{B}_d \boldsymbol{u}[k] + \boldsymbol{D}_d \boldsymbol{w}[k]$ and $\mathbb{U}_t[k] = \boldsymbol{B}_d \mathbb{U}[k] + \boldsymbol{D}_d \mathbb{W}[k]$.

In this thesis, the polytopic input set $\mathbb{U} = \mathbb{P}(\boldsymbol{A}_u, \boldsymbol{b}_u)$ in (6.6) is approximated by minimum volume enclosing ellipsoid. This ellipsoid is the so-called Lowner-Jhon ellipsoid [17] and is obtained as

$$\boldsymbol{q}_u = -\boldsymbol{L}_u^{-1} \boldsymbol{c}_u \tag{6.20}$$

$$\boldsymbol{Q}_u = (\boldsymbol{L}_u^T \boldsymbol{L}_u)^{-1} \tag{6.21}$$

where $\boldsymbol{L}_u \in \mathbb{R}^{n_x \times n_x}$ and $\boldsymbol{c}_u \in \mathbb{R}_x^n$ are calculated by solving the convex optimization problem,

$$\begin{aligned} \min_{\boldsymbol{L}_u, \boldsymbol{c}_u} \quad & \log\det \boldsymbol{L}_{\boldsymbol{u}}^{-1} \\ \text{s.t.} \quad & ||\boldsymbol{L}_u \boldsymbol{v}_{u_i} + \boldsymbol{c}_u||_2 \leq 1, \qquad i = 1, ..., n_u \end{aligned} \tag{6.22}$$

where v_{u_i} and n_u are the i-th vertex of polytope and the number of polytope vertices, respectively.

Given the ellipsoidal input set $\mathbb{E}(\boldsymbol{q}_u, \boldsymbol{Q}_u) \supset \mathbb{U}$ and ellipsoidal disturbance set ($\mathbb{W} = \mathbb{E}(\boldsymbol{q}_w, \boldsymbol{Q}_w)$), the total input set $\mathbb{U}_t$ is approximated by minimum volume ellipsoid $\mathbb{E}(\boldsymbol{q}_t, \boldsymbol{Q}_t)$ enclosing the Minkowski sum of two ellipsoids. The parameters of this ellipsoid are calculated as [43]

$$\boldsymbol{q}_{u_t} = \boldsymbol{B}_d \boldsymbol{q}_u + \boldsymbol{D}_d \boldsymbol{q}_w \tag{6.23}$$

$$\boldsymbol{Q}_{u_t} = (1 + \frac{1}{\beta})\boldsymbol{Q}_{u_b} + (1 + \beta)\boldsymbol{Q}_{u_d} \tag{6.24}$$

where $\boldsymbol{Q}_{u_b}, \boldsymbol{Q}_{u_d} \in \mathbb{R}^{n_x \times n_x}$

$$\boldsymbol{Q}_{u_b} = \boldsymbol{B}_d \boldsymbol{Q}_u \boldsymbol{B}_d^T \tag{6.25}$$

$$\boldsymbol{Q}_{u_d} = \boldsymbol{D}_d \boldsymbol{Q}_w \boldsymbol{D}_d^T \tag{6.26}$$

β is derived based on recursive equation

$$\beta_{n+1} = \sqrt{\frac{\Sigma_{i=1}^{n_s}(1 + \beta_n \lambda_i)^{-1}}{\Sigma_{i=1}^{n_s}\lambda_i(1 + \beta_n \lambda_i)^{-1}}} \tag{6.27}$$

where λ_i is the i-th eigenvalue of $\boldsymbol{R} = \boldsymbol{Q}_{u_b}^{-1}\boldsymbol{Q}_{v_b}$.

If $n_u < n_x$ or $n_w < n_x$, $\boldsymbol{Q}_{u_b}$ or $\boldsymbol{Q}_{u_w}$ becomes degenerate and instead of using (6.25) and (6.26), they are calculated by

$$\boldsymbol{Q}_{u_b} = \boldsymbol{B}_d \boldsymbol{Q}_u \boldsymbol{B}_d^T + \epsilon^2 I_{n_x} \tag{6.28}$$

$$\boldsymbol{Q}_{u_d} = \boldsymbol{D}_d \boldsymbol{Q}_w \boldsymbol{D}_{\boldsymbol{d}}^T + \epsilon^2 \boldsymbol{I}_{n_x} \tag{6.29}$$

where ϵ is a small value making matrices non-degenerate

Given the ellipsoidal total input set $\mathbb{E}(\boldsymbol{q}_{u_t}, \boldsymbol{Q}_{u_t})$ and ellipsoidal initial state set $\mathbb{E}(\boldsymbol{q}_{x_0}, \boldsymbol{Q}_{x_0})$, and using (6.12) and (6.13), the UAV reach set at each sample is approximated by minimum volume enclosing ellipsoid $E(\boldsymbol{q}_x[k], \boldsymbol{Q}_x[k]) \supset \mathbb{X}[k]$. Here $\boldsymbol{q}_x[k] \in \mathbb{R}^{n_x}$ and $\boldsymbol{Q}_x[k] \in \mathbb{R}^{n_x \times n_x}$ are the center and the shape matrix of the ellipsoidal reach set, respectively.

Given the ellipsoidal disturbance set $\mathbb{E}(\boldsymbol{q}_w, \boldsymbol{Q}_w)$ and ellipsoidal initial state set $\mathbb{E}(\boldsymbol{q}_{e_0}, \boldsymbol{Q}_{e_0}) = \mathbb{X}_0 - \bar{\boldsymbol{x}}_0$, and using using (6.12) and (6.13), the state deviation reach set of the UAV at each sample is approximated by minimum volume enclosing ellipsoid $\mathbb{E}(\boldsymbol{q}_{x_e}[k], \boldsymbol{Q}_{x_e}[k]) \supset \mathbb{X}_e[k]$. Here $\boldsymbol{q}_{x_e}[k] \in \mathbb{R}^{n_x}$ and $\boldsymbol{Q}_{x_e}[k] \in \mathbb{R}^{n_x \times n_x}$ are the center and the shape matrix of the ellipsoidal state deviation set, respectively.

6.4 Safe Convex Region Approximation

Obstacle-free space is usually non-convex in its original form and creating non-convex constraints in the optimization problem. To overcome this problem, the obstacle-free space can be approximated by a convex region. In this study, each obstacle is modelled by a polytopic set with uncertainties in its motion

$$\mathbb{X}_u^i[k] = \{\boldsymbol{x} | \boldsymbol{A}_{x_u}^i (\boldsymbol{x} - \boldsymbol{p}_c^i[k]) \preceq \boldsymbol{d}_{x_u}^i\} \tag{6.30}$$

where $\boldsymbol{A}^i_{x_u} \in \mathbb{R}^{m_i\times 3}$ and $\boldsymbol{b}^i_{x_u} \in \mathbb{R}^{m_i}$ are parameters of i-th polytopic obstacle (out of n_o) and m_i is the number of faces of i-th polytope. $\boldsymbol{p}^i_c[k] \in \boldsymbol{x}^i_c[k]$ is the position subvector of the center of i-th obstacle and $\boldsymbol{x}^i_c[k] \in \mathbb{R}^{n_x}$ represents states of i-th polytopic obstacle derived by

$$\boldsymbol{x}^i_c[k+1] = \boldsymbol{A}^i_c\boldsymbol{x}^i_c[k] + \boldsymbol{B}^i_c\boldsymbol{u}^i_c[k],\ \boldsymbol{x}^i_c[0] = \boldsymbol{x}^i_{c_0} \tag{6.31}$$

where $\boldsymbol{A}^i_c \in \mathbb{R}^{n_x\times n_x}$ and $\boldsymbol{B}^i_c \in \mathbb{R}^{n_x\times n_u}$. $\boldsymbol{x}^i_{c_0} \in \mathbb{X}^i_{c_0}$ is the initial state of the i-th obstacle where $\mathbb{X}^i_{c_0} \subset \mathbb{R}^{n_x}$ is the initial state set. $\boldsymbol{u}^i_c[k] \in \mathbb{U}^i_c$ is the input to the system where $\mathbb{U}^i_c \in \mathbb{R}^{n_u}$ is the set of possible inputs acting on the i-th obstacle.

To approximate the reach set corresponding to each obstacle ($\mathbb{X}^i_c[k]$) by minimum volume enclosing ellipsoid, ellipsoidal input set $\mathbb{U}_c = \mathbb{E}_{u_c}(\boldsymbol{q}^i_{u_c}, \boldsymbol{Q}^i_{u_c})$ and ellipsoidal initial state set $\mathbb{X}_{c_0} = \mathbb{E}(\boldsymbol{q}^i_{c_0}, \boldsymbol{Q}^i_{c_0})$ are considered. Using (6.12) and(6.13), the ellipsoidal approximation of reach set of each obstacle at each sample ($\mathbb{E}(\boldsymbol{q}^i_{x_c}[k], \boldsymbol{Q}^i_{x_c}[k])$) is obtained. Here $\boldsymbol{q}^i_{x_c}[k] \in \mathbb{R}^{n_x}$ and $\boldsymbol{Q}^i_{x_c}[k] \in \mathbb{R}^{n_x\times n_x}$ are the center and shape matrix of ellipsoidal reach set, respectively. To find the reachable position of each obstacle, the projection of ellipsoidal reach set in positional coordinates is calculated

$$\mathbb{E}(\boldsymbol{q}^i_{p_c}[k], \boldsymbol{Q}^i_{p_c}[k]) = prj_p(\mathbb{E}(\boldsymbol{q}^i_{x_c}[k], \boldsymbol{Q}^i_{x_c}[k]) \tag{6.32}$$

here $\boldsymbol{q}^i_{p_c}[k] \in \mathbb{R}^3$ is the position subvector of $\boldsymbol{q}^i_{x_c}[k]$ and $\boldsymbol{Q}^i_{p_c}[k] \in \mathbb{R}^{3\times 3}$ is shape matrix of projected ellipsoid in positional coordinates calculated as (proof in appendix C.3).

$$\boldsymbol{Q}^i_{p_c}[k] = (\boldsymbol{U}_p - \boldsymbol{U}^T_{pv}\boldsymbol{U}^{-1}_v\boldsymbol{U}_{pv})^{-1} \tag{6.33}$$

$\boldsymbol{U}_p, \boldsymbol{U}_{pv}, \boldsymbol{U}_v \in \mathbb{R}^{3\times3}$ are submatrices of $(\boldsymbol{Q}^i_{x_c}[k])^{-1}$

$$\begin{bmatrix} \boldsymbol{U}_p & \boldsymbol{U}_{pv}^T \\ \boldsymbol{U}_{pv} & \boldsymbol{U}_v \end{bmatrix} = \boldsymbol{Q}^i_{x_c}[k] \tag{6.34}$$

To find the outer polytopic approximation of each obstacle under uncertainties defined in (6.30), the following proposition is presented.

Proposition 1. *Consider the set* $\mathbb{X}$ *is defined as*

$$\mathbb{X} = \{\boldsymbol{x} : \boldsymbol{A}_a(\boldsymbol{x}+\boldsymbol{y}) \leq \boldsymbol{b}_a,\ y \in \mathbb{E}(\boldsymbol{q}_a, \boldsymbol{Q}_a), \boldsymbol{b}_a \in \mathbb{R}^n_{\geq 0}\} \tag{6.35}$$

The inner and outer polytopic approximation of set $\mathbb{X}$ *is derived as* $\mathbb{P}(\boldsymbol{A}_{in}, \boldsymbol{b}_{in}) \subseteq \mathbb{X} \subseteq \mathbb{P}(\boldsymbol{A}_{out}, \boldsymbol{b}_{out})$

$$\boldsymbol{A}_{in} = \boldsymbol{A}_{out} = \boldsymbol{A}_a \tag{6.36}$$

$$\boldsymbol{b}_{in} = \boldsymbol{b}_a - \boldsymbol{A}_a\boldsymbol{q}_a - ||\boldsymbol{A}_a\boldsymbol{L}_a||_\bullet \tag{6.37}$$

$$\boldsymbol{b}_{out} = \boldsymbol{b}_a - \boldsymbol{A}_a\boldsymbol{q}_a + ||\boldsymbol{A}_a\boldsymbol{L}_a||_\bullet \tag{6.38}$$

where $\boldsymbol{L}_a$ *is the Cholesky factorization of* $\boldsymbol{Q}_a$. $||\boldsymbol{A}_a\boldsymbol{L}_a||_\bullet$ *is defined as*

$$||\boldsymbol{A}_a\boldsymbol{L}_a||_\bullet = [||\boldsymbol{A}_a^{1,*}\boldsymbol{L}_a||_2,\ ||\boldsymbol{A}_a^{2,*}\boldsymbol{L}_a||_2,\ \ldots,\ \boldsymbol{A}_a^{m,*}\boldsymbol{L}_a||_2]^T \tag{6.39}$$

where $\boldsymbol{A}_a^{1,*}$ *is the first row (out of* m*) of* $\boldsymbol{A}_a$.

Proof. Inserting the ellipsoidal equation $\boldsymbol{y} = \boldsymbol{q}_a + \boldsymbol{L}_a\boldsymbol{w},\ ||\boldsymbol{w}||_2 \leq 1$ in (6.35), the set

X is expressed as

$$\mathbb{X} = \{\boldsymbol{x} : \boldsymbol{A}_a\boldsymbol{x} \preceq \boldsymbol{b}_a - \boldsymbol{A}_a\boldsymbol{q}_a - \boldsymbol{A}_a\boldsymbol{L}_a\boldsymbol{w}, ||w|| \leq 1, \boldsymbol{b}_a \in \mathbb{R}^n_{\geq 0}\} \tag{6.40}$$

Considering $\boldsymbol{A}_a^{i,*}\boldsymbol{L}_a\boldsymbol{w} \leq ||\boldsymbol{A}_a^{i,*}\boldsymbol{L}_a||$ for all $||w|| \leq 1$, it is concluded if

$$\boldsymbol{A}_a\boldsymbol{x} \preceq \boldsymbol{b}_a - \boldsymbol{A}_a\boldsymbol{q}_a - ||\boldsymbol{A}_a\boldsymbol{L}_{\boldsymbol{a}}||_{\bullet}, \text{ then } \boldsymbol{A}_a\boldsymbol{x} \preceq \boldsymbol{b}_a - \boldsymbol{A}_a\boldsymbol{y} \tag{6.41}$$

Moreover, it is concluded if

$$\boldsymbol{A}_a\boldsymbol{x} \preceq \boldsymbol{b}_a - \boldsymbol{A}_a\boldsymbol{y}, \text{ then } \boldsymbol{A}_a\boldsymbol{x} \preceq \boldsymbol{b}_a - \boldsymbol{A}_a\boldsymbol{q}_a + ||\boldsymbol{A}_a\boldsymbol{L}_{\boldsymbol{a}}||_{\bullet} \tag{6.42}$$

Using Proposition 1, the outer polytopic approximation for each obstacle is derived as $\mathbb{P}(\boldsymbol{A}_o, \boldsymbol{b}_o)$ where $\boldsymbol{b}_o^i = \boldsymbol{d}_{x_c}^i + \boldsymbol{A}_o^j\boldsymbol{q}_{p_c}^j[k] + ||\boldsymbol{A}_o\boldsymbol{L}_c^i[k]||_2$ and $\boldsymbol{L}_c^i[k] \in \mathbb{R}^{3\times3}$ is the Cholesky factorization of $\boldsymbol{Q}_{p_c}^i[k]$.

In this thesis, the approximated convex safe region is represented by the polyhedron $\mathbb{P}(\boldsymbol{A}_p[k], \boldsymbol{b}_p[k])$ where $\boldsymbol{A}_p[k] \in \mathbb{R}^{m_p\times3}$ and $\boldsymbol{b}_p[k] \in \mathbb{R}^{m_p}$ are the polyhedron parameters and m_p is the number of its faces. This polyhedron is obtained as the intersection of half-spaces created by the convex region generation algorithm introduced in Chapter 5. In this algorithm, a safe half-space with maximum reachable region is determined by finding the separating hyperplane between the polyhedral obstacle and target point of the UAV at each sample time. Polytopic velocity constraints in the form of $\mathbb{P}(\boldsymbol{A}_v, \boldsymbol{b}_v)$ where $\boldsymbol{A}_v \in \mathbb{R}^{m_v\times n_v}$ and $\boldsymbol{b}_v \in \mathbb{R}^{m_v}$ are the parameters of polytopic velocity set and m_v is the number of the faces of the polytope, are imposed to ensure safe movement of the UAV.

Given the polyhedral safe region and polytopic velocity set, the safe region can be modelled by polyhedron $X_s[k] = \mathbb{P}(\boldsymbol{A}_x[k], \boldsymbol{b}_x[k])$ with

$$\boldsymbol{A}_x[k] = \begin{bmatrix} \boldsymbol{A}_p[k] & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_v \end{bmatrix}, \ \boldsymbol{b}_x[k] = \begin{bmatrix} \boldsymbol{b}_p[k] \\ \boldsymbol{b}_v \end{bmatrix} \tag{6.43}$$

6.5 Ellipsoidal-based RMPC

To formulate the ellipsoidal-based RMPC (6.9a-6.9d) as a convex optimization problem with linear constraints, $\mathbb{X}_s[k] \ominus \mathbb{X}_e[k]$ should be approximated by a polyhedral set. In this thesis, the safe region is modelled by polyhedron and the state deviation reach set is approximated by the ellipsoidal set. Therefore, the constraint in (6.9c) can be formulated as Pontryagin difference of polyhedron and ellipsoid. To approximate this constraint by inner polytopic set, the following proposition is presented.

Proposition 2. *Consider the polytopic set $\mathbb{P}(\boldsymbol{A}_x, \boldsymbol{b}_x)$ and ellipsoidal set $\mathbb{E}(\boldsymbol{q}_y, \boldsymbol{Q}_y)$, the set $\mathbb{Z} = \mathbb{P}(\boldsymbol{A}_x, \boldsymbol{b}_x) \ominus \mathbb{E}(\boldsymbol{q}_y, \boldsymbol{Q}_y)$ can be approximated by inner polytopic set $\mathbb{P}_z(\boldsymbol{A}_z, \boldsymbol{b}_z) \subseteq \mathbb{Z}$ where*

$$\boldsymbol{A}_z = \boldsymbol{A}_x, \ \boldsymbol{b}_z = \boldsymbol{b}_x - \boldsymbol{A}_x \boldsymbol{q}_y - ||\boldsymbol{A}_x \boldsymbol{L}_y||_{\bullet} \tag{6.44}$$

where $\boldsymbol{L}_y$ is the Cholesky factorization of $\boldsymbol{Q}_y$. $||\boldsymbol{A}_x \boldsymbol{L}_y||_{\bullet}$ is defined as

$$||\boldsymbol{A}_x \boldsymbol{L}_y||_{\bullet} = [||\boldsymbol{A}_x^{1,*} \boldsymbol{L}_y||_2, \ ||\boldsymbol{A}_x^{2,*} \boldsymbol{L}_y||_2, \ \ldots, \ \boldsymbol{A}_x^{m,*} \boldsymbol{L}_y||_2]^T \tag{6.45}$$

where $\boldsymbol{A}_x^{1,*}$ is the first row (out of m) of $\boldsymbol{A}_a$.

Proof. The set $\mathbb{Z}$ is obtained as Minkowski difference of polytopic and ellipsoidal sets $\mathbb{Z} = \{\boldsymbol{z} : |\boldsymbol{z} \oplus \mathbb{E}(\boldsymbol{q_y}, \boldsymbol{Q_y}) \subseteq \mathbb{P}(\boldsymbol{A_x}, \boldsymbol{b_x})\}$. Using polytopic equation, the set Z is formulated as

$$\mathbb{Z} = \{\boldsymbol{z} : |\boldsymbol{A}_x(\boldsymbol{z} + \boldsymbol{y}) \preceq \boldsymbol{b}_x,\ \boldsymbol{y} \in \mathbb{E}(\boldsymbol{q}_y, \boldsymbol{Q}_y)\} \tag{6.46}$$

Using Proposition 1, the set $\mathbb{Z}$ is approximated by inner polytope $\mathbb{P}_{in}$.

Using Proposition 2, the state constraint (6.9c) is approximated by inner polytopic set $\mathbb{P}(\boldsymbol{A}_{\bar{x}}, \boldsymbol{b}_{\bar{x}})$. Therefore, the ellipsoidal based RMPC in (6.9a-6.9d) is formulated as quadratic program with linear constraints

$$\min_{\boldsymbol{u}[0,N-1]} \sum_{k=0}^{N-1} (\boldsymbol{u}[k] - \boldsymbol{u}_h[k])^T \boldsymbol{R} (\boldsymbol{u}[k] - \boldsymbol{u}_h[k]) \tag{6.47a}$$

$$\text{s. t.} : \bar{\boldsymbol{x}}[k+1] = \boldsymbol{A}_d \bar{\boldsymbol{x}}[k] + \boldsymbol{B}_d \boldsymbol{u}[k] \tag{6.47b}$$

$$\boldsymbol{A}_{\bar{x}}[k]\boldsymbol{x}[k] \preceq \boldsymbol{b}_{\bar{x}}[k] \tag{6.47c}$$

$$A_u[k]\boldsymbol{u} \preceq \boldsymbol{b}_u \tag{6.47d}$$

where $\boldsymbol{A}_{\bar{x}}[k] \in \mathbb{R}^{m_x \times n_x}$ and $\boldsymbol{b}_{\bar{x}}[k] \in \mathbb{R}^{m_x}$ are the parameters of polyhedral safe region.

6.6 Experiment

A schematic view of the experimental setup is demonstrated in Fig.4.3. The update rate and time horizon of the MPC-based collision avoidance assistance are $T_s = 30\ ms$ and $t_f = 1.5\ s$, respectively. The operator's acceleration commands are assumed constant over the MPC time horizon. The parameters of polytopic input and velocity constraints in (6.47c) and (6.43) are selected as

$$\boldsymbol{A}_u = \boldsymbol{A}_v = [-\boldsymbol{I}_3,\ \boldsymbol{I}_3]^T \tag{6.48}$$

$$\boldsymbol{b}_u = [2,\ 0.5,\ 2,\ 2,\ 0.5,\ 2]^T,\ \boldsymbol{b}_v = [1.5,\ 0.5,\ 1.5,\ 1.5,\ 0.5,\ 1.5]^T \tag{6.49}$$

Uncertainty in the form of disturbance is considered for the UAV. This is to model errors in the inner-loop acceleration controller. The disturbance set is modelled by the ellipsoid $\boldsymbol{w} \in \mathbb{E}(\boldsymbol{q}_w, \boldsymbol{Q}_w)$, where the parameters of the ellipsoid are determined based on several experiments

$$\boldsymbol{q}_w = \boldsymbol{0}_3,\ \boldsymbol{Q}_w = diag(0.75, 0.19, 1.47) \tag{6.50}$$

Two-dimensional representation of the obstacles are shown in Fig. 6.2 and Fig. 6.2, where the height of each obstacle is one meter. Uncertainties in the form ellipsoidal initial state set $\mathbb{E}(\boldsymbol{q}^i_{c_0}, \boldsymbol{Q}^i_{c_0})$ and ellipsoidal input set $\mathbb{E}(\boldsymbol{q}_{u_c}, \boldsymbol{Q}_{u_c})$ are considered for each obstacle. The following parameters are considered for ellipsoidal sets.

$$\boldsymbol{q}^i_{c_0} = [(\boldsymbol{p}^i_c)^T, \boldsymbol{0}^T_3]^T,\ \boldsymbol{Q}^i_{c_0} = diag(0.05^2, \epsilon^2, \epsilon^2, 0.1^2, \epsilon^2, \epsilon^2) \tag{6.51}$$

$$\boldsymbol{q}_{u_c} = \boldsymbol{0}_3,\ \boldsymbol{Q}_{u_c} = diag(0.1^2, \epsilon^2, \epsilon^2) \tag{6.52}$$

where $\boldsymbol{p}^i_c \in \mathbb{R}^3$ is the center of i-th obstacle and ϵ is the small value making the ellipsoids non-degenerate.

Several optimization problem must be solved in the proposed collision avoidance algorithm. In (6.22), the minimum volume ellipsoids covering the polytopes are formulated as semidefinite program [13], which is solved off-line using Mosek solver [5]. The convex region generation algorithm presented in Chapter 5 and proposed collision avoidance algorithm in (6.47a-6.47d) are formulated as quadratic program and are solved in real-time using QSQP solver with its code generation [86],[10].

6.6.1 Flight Test

Two experiments are carried out to demonstrate the performance of the proposed collision avoidance assistance in the presence of uncertainties. The first experiment considers a case where the UAV is subject to disturbance. The second experiment investigates a scenario where disturbance on the UAV and uncertainty in the obstacle positions are considered.

In the first experiment, two pairs of obstacles are placed in the environment such that the path in front of the UAV is blocked by the second pair of obstacles, as demonstrated in Fig. 6.2. The operator teleoperates the UAV by providing the acceleration commands as demonstrated in Fig. 6.3(b). The collision avoidance generates the corrective commands as shown in Fig. 6.3(b) to modify the UAV trajectory and assist the operator in guiding it through the opening between the obstacles. To find obstacles with the chance of collision, the intersection of UAV reach set with each obstacle at each sample is examined (Fig. 6.2(a,c,e)). If the obstacle is identified as an unsafe obstacle with the chance of collision, the separating plane is created using the convex region generation presented in Chapter 7. Collision-free motion for all possible disturbances is guaranteed by ensuring the state deviation reach sets do not intersect with the separating planes as shown in Fig. 6.2(b,d,f).

In the second experiment, uncertainties are considered in both the UAV and obstacles. The trajectory of the UAV is presented in Fig. 6.4. In this experiment, inflated obstacles due to uncertainties at each sample time are obtained and their intersection with the UAV reach set is examined, e.g., see Fig. 6.4(a,c,e). As shown in Fig. 6.4(b,d,f)), obstacles with the chance of collision at each sample time are identified, and the separating planes are created between the UAV and the inflated obstacles. Fig. 6.5(a) shows corrective commands that modify the operator's commands in Fig. 6.5(b). The corrective commands assist the operator in guiding the UAV to pass through the opening between the obstacles for all possible uncertainties, i.e, the state deviation sets do not intersect with inflated obstacles as demonstrated in Fig. 6.4(b,d,f).

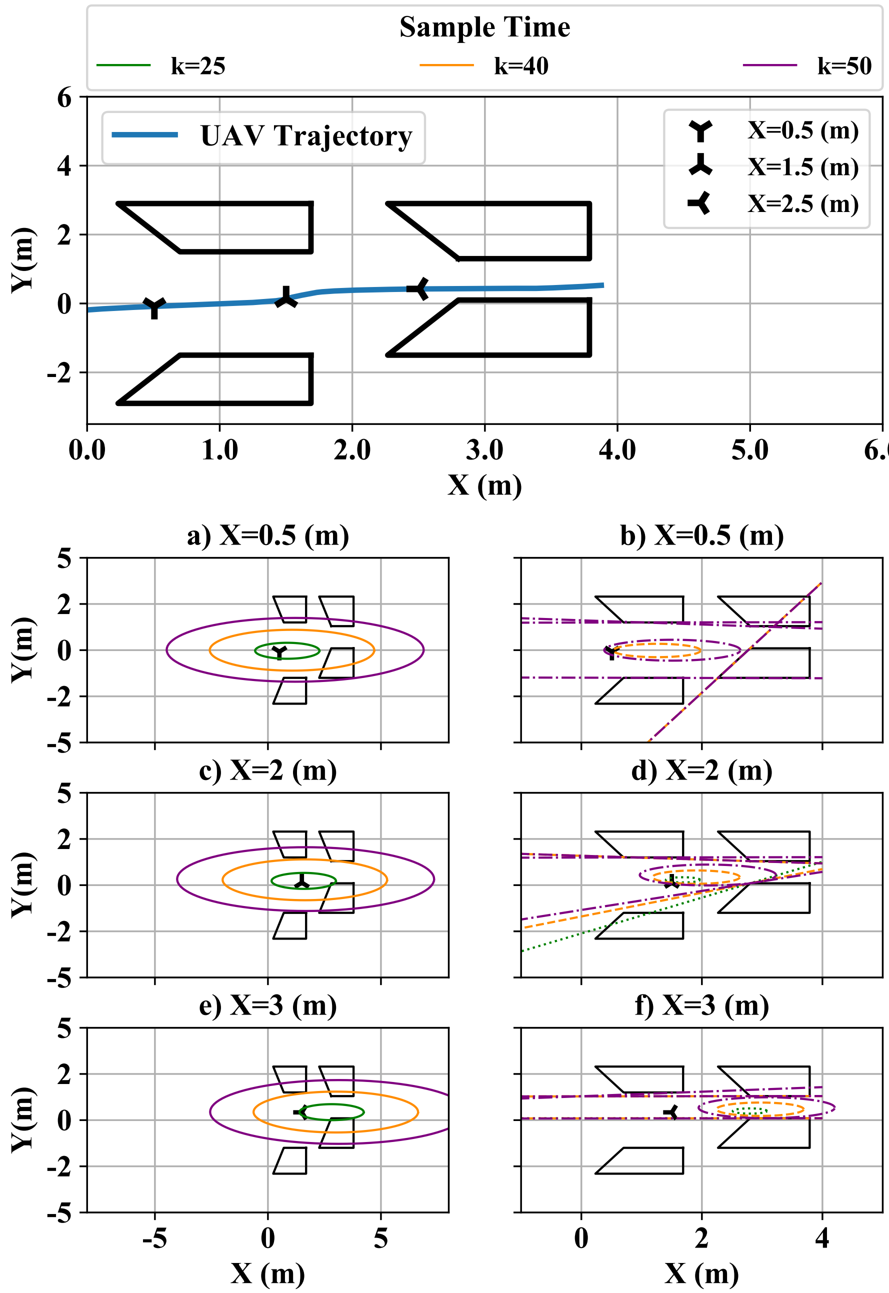

Figure 6.2: Two-dimensional representation of the UAV trajectory. Two dimensional representation of the UAV reach set at three different samples for a) X=0.5 (m), c)X=2 (m), e) X=3 (m). Two-dimensional representation of the state deviation sets and separating planes at three different samples for b) X=0.5 (m), d)X=2 (m), f) X=3 (m).

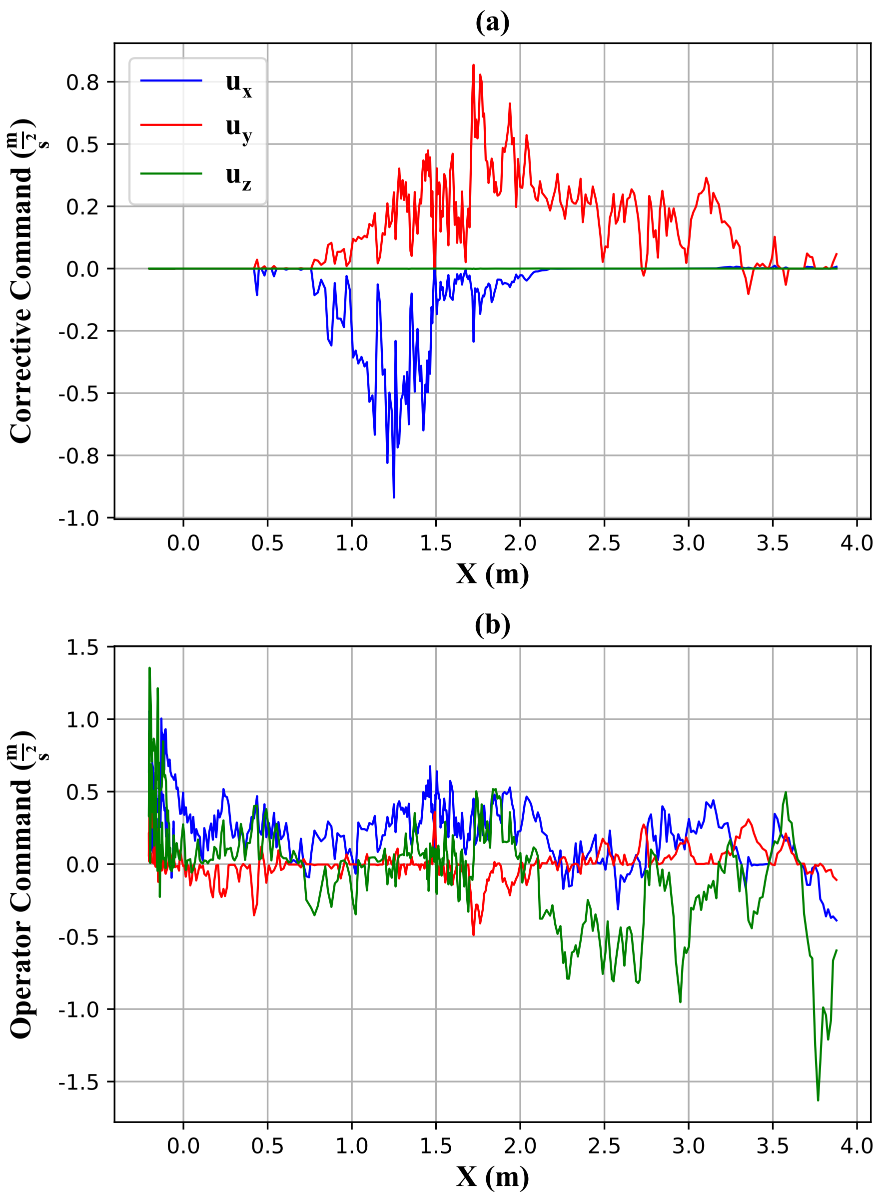

Figure 6.3: a) Corrective commands created by the proposed collision avoidance assistance. b) Operator commands.

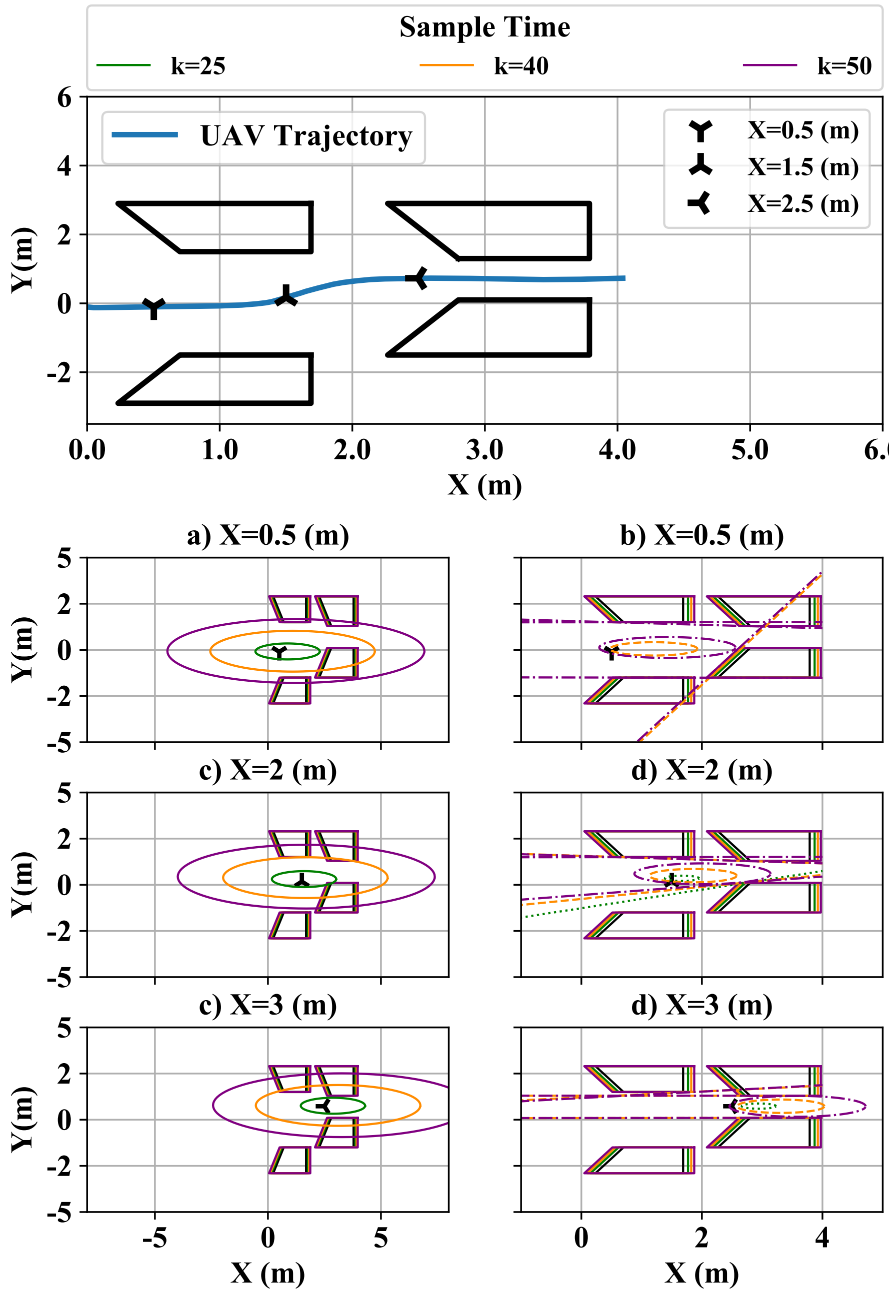

Figure 6.4: Two-dimensional representation of the UAV trajectory. Two-dimensional representation of the UAV reach set and inflated obstacles at three different samples for a) X=0.5 (m), c)X=2 (m), e) X=3 (m). Two-dimensional representation of the state deviation sets, inflated obstacles, and separating planes at three different samples for b) X=0.5 (m), d)X=2 (m), f) X=3 (m).

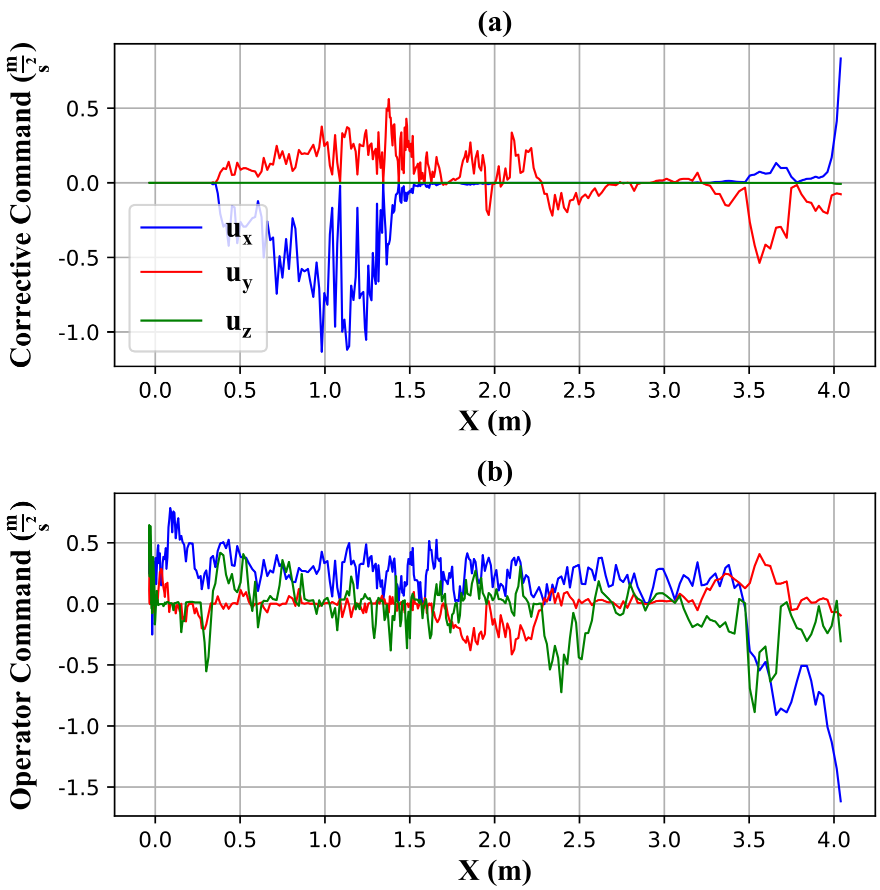

Figure 6.5: a) Corrective commands created by the proposed collision avoidance assistance. b) Operator commands.

Chapter 7

Recursive Feasibility in Collision Avoidance

MPC-based collision avoidance is a powerful tool to assist the operator safely navigate the UAV in an obstacle-rich environment. The MPC-based collision avoidance algorithm repeatedly solves an optimization problem to modify the operator's command in order to avoid collision. A fundamental problem in finite-time MPC is the lack of guaranteed recursive feasibility. In particular, the feasibility of MPC at the initial time does not guarantee its feasibility at all iterations. This could potentially place the UAV into a state where collision with obstacles would become unavoidable.

In this chapter, new MPC-based collision avoidance assistance algorithms with guaranteed recursive feasibility for UAV teleoperation with/without uncertainties are introduced. Recursive feasibility is guaranteed by adding proper terminal constraints to the MPC-based collision avoidance introduced in Chapter 5. To extend the collision avoidance assistance with recursive feasibility for the UAV under uncertainties, a novel ellipsoidal tube-based MPC is introduced. In this algorithm, the operator teleoperates the UAV by providing linear acceleration commands while the ellipsoidal tube-based MPC modifies these commands to ensure collision-free operation in the presence of disturbances. Collision avoidance is formulated as a convex optimization problem in three steps. First, an ellipsoidal approximation of the robust positively invariant (RPI) set is derived using a new RPI set approximation based on ellipsoidal techniques. Next, a polytopic approximation of the obstacle-free space is derived using the SVM algorithm. Finally, an inner polytopic approximation of the tightened constraints is obtained using geometrical relation between the ellipsoidal RPI set and polytopic safe region.

7.1 Reachability analysis and Set Invariance theory

Consider the discrete-time linear dynamic system

$$\boldsymbol{x}[k+1] = \boldsymbol{A}_d \boldsymbol{x}[k] + \boldsymbol{D}_d \boldsymbol{w}[k] \tag{7.1}$$

where $\boldsymbol{w}[k] \in \mathbb{W}$ is disturbance input and $\boldsymbol{x}_0 \in \mathbb{X}_0$ is the initial states of the system. In this thesis, it is assumed $\boldsymbol{A}_d \in \mathbb{R}^{n_x \times n_x}$ is strictly stable and $\boldsymbol{w}_d[k] \in \mathbb{W}_d$ with $\mathbb{W}_d \in ComC(\mathbb{R}^{n_x})$. Here $\boldsymbol{w}_d[k] = \boldsymbol{D}_d \boldsymbol{w}[k]$ and $ComC(\mathbb{R}^{n_x})$ denotes the collection of convex and compact sets containing the origin in $\mathbb{R}^{n_x}$.

Definition 1. *The reach set* $\mathbb{X}[k]$ *for the system* (7.1) *is defined as a set of all states* *at k-sample which can be reached from initial states* $\boldsymbol{x}_0 \in \mathbb{X}_0$ *under all possible disturbance inputs* $\boldsymbol{w}_d[.] \in \mathbb{W}_d$.

The reach set at k-th sample is derived as $\mathbb{X}[k] = F_k(X_0)$ where $F_k(X_0)$ is a mapping function defined as

$$F_k(X_0) = \boldsymbol{A}_d^k \mathbb{X}_0 \oplus \bigoplus_{i=0}^{k-1} \boldsymbol{A}_d^i \mathbb{W}_d \tag{7.2}$$

and $\oplus$ denotes the Minkowski sum, and $\bigoplus_{i=0}^{k-1} \boldsymbol{A}_d^i \mathbb{W}_d = \mathbb{W}_d \oplus \boldsymbol{A}_d \mathbb{W}_d \oplus ... \oplus \boldsymbol{A}_d^{k-1} \mathbb{W}_d$.

Definition 2. *The set* $\mathbb{S}$ *is called robust positively invariant (RPI) set for the system (7.1) if for all* $\boldsymbol{x} \in \mathbb{S}$ *and* $\boldsymbol{w}_d \in \mathbb{W}_d$, $\boldsymbol{A}_d \boldsymbol{x} + \boldsymbol{w}_d \in \mathbb{S}$, *i.e.,* $\boldsymbol{F}_1(\mathbb{S}) \subseteq \mathbb{S}$.

Definition 3. *The set* $\bar{\mathbb{S}}$ *is called minimum RPI set for system (7.1) if it is RPI set and is contained in every RPI set, i.e.,* $\boldsymbol{F}_1(\bar{\mathbb{S}}) \subseteq \bar{\mathbb{S}}$ *and* $\bar{\mathbb{S}} \subseteq \mathbb{S}$ *for all* $\mathbb{S}$ *satisfying* $\boldsymbol{F}_1(\mathbb{S}) \subseteq \mathbb{S}$.

7.2 Collision Avoidance in the UAV

A schematic view of the proposed share control strategy is presented in Fig. 7.1. The operator tele-operates the UAV by providing the acceleration commands while a model predictive collision avoidance assistance modifies the operator commands to ensure collision-free operation. The discrete-time MPC is formulated as

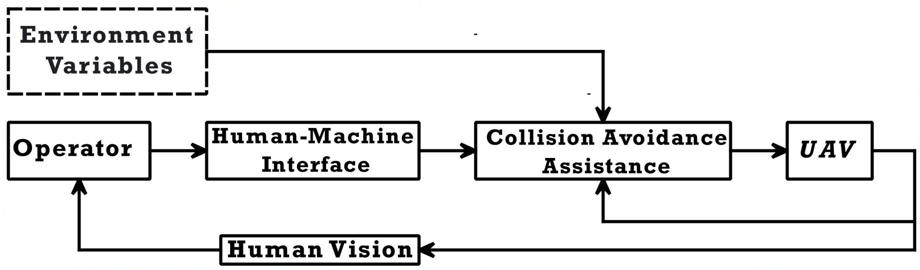

Figure 7.1: UAV teleoperation with automated collision avoidance assistance

$$\min_{\boldsymbol{u}[0,N-1]|k} \sum_{i=0}^{N-1} (\boldsymbol{u}[i|k] - \boldsymbol{u}_h[i|k])^T \boldsymbol{R}_u[i|k](\boldsymbol{u}[i|k] - \boldsymbol{u}_h[i|k]) \tag{7.3a}$$

$$\text{s.t.} : \boldsymbol{x}[i+1|k] = \boldsymbol{A}_d\boldsymbol{x}[i|k] + \boldsymbol{B}_d\boldsymbol{u}[i|k] \tag{7.3b}$$

$$\boldsymbol{v}[N|k] = \boldsymbol{0}_3 \tag{7.3c}$$

$$\boldsymbol{u}[i|k] \in \mathbb{U} \tag{7.3d}$$

$$\boldsymbol{x}[i|k] \in \mathbb{X}_s[k] \tag{7.3e}$$

where $\boldsymbol{u}[i|k], \boldsymbol{u}_h[i|k] \in \mathbb{R}^{n_u}$ are the UAV nominal inputs and predicted operator commands at i-th sample with respect to k-th sample (current inputs), respectively. $\boldsymbol{x}[i|k] \in \mathbb{R}^{n_x}$ is the UAV states, $\boldsymbol{R}_u[i|k] \in \mathbb{R}^{n_u \times n_u}$, and N is the number of samples within the MPC time horizon. Constraints related to system dynamics are represented in (7.3b). Terminal velocity constraints are presented in (7.3c) where $\boldsymbol{v}[N|k] \in \mathbb{R}^3$ is the velocity of the UAV at the terminal sample time. Input constrains are formulated in (7.3d) where $\mathbb{U}$ is the set of admissible inputs. State constraints including obstacle avoidance constraints and velocity constrains are presented in (7.3e), where $\mathbb{X}_s[i|k]$ represents state constraints.

7.2.1 Safe Convex Region Generation

Obstacle-free space is usually non-convex in its original form, creating non-convex constraints in the optimization problem. To overcome this problem, the obstacle-free space can be approximated by a convex region. In this chapter, each obstacle is modelled by polytopic set $\mathbb{P}^j_{x_u} = \mathbb{P}(\boldsymbol{A}^j_{x_u}, \boldsymbol{b}^j_{x_u})$ where

$$\mathbb{P}(\boldsymbol{A}^j_{x_u}, \boldsymbol{b}^j_{x_u}) = \{\boldsymbol{x} : \boldsymbol{A}^j_{x_u}\boldsymbol{x} \preceq \boldsymbol{b}^j_{x_u}\} \tag{7.4}$$

where $\boldsymbol{A}^j_{x_u} \in \mathbb{R}^{m_j \times 3}$ and $\boldsymbol{b}^j_{x_u} \in \mathbb{R}^{m_j}$ are the parameters of j-th (out of n_o) polytopic obstacle and m_j is the number of faces of j-th polytopic obstacle.

The obstacle-free region is approximated by the intersection of half-spaces generated by separating planes between the UAV and obstacles. Each separating plane is obtained using the hard-margin SVM algorithm [41]. This algorithm finds a separating plane between two disjoint sets σ and φ with maximum distance to the two sets, i.e.,

$$\max_{\boldsymbol{\alpha},\eta}(\min_{\boldsymbol{x}_i \in \sigma}\frac{|\boldsymbol{\alpha}^T\boldsymbol{x}_i + \eta|}{||\alpha||_2} + \min_{\boldsymbol{y}_j \in \varphi?}\frac{|\boldsymbol{\alpha}^T\boldsymbol{y}_j + \eta|}{||\alpha||_2}) \tag{7.5}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^3$ and $\eta \in \mathbb{R}$ are the parameters of separating plane and $\boldsymbol{x}_i,\ \boldsymbol{y}_j \in \mathbb{R}^3$ are the i-th and j-th members of sets σ and φ, respectively.

To utilize the SVM algorithm in finding the separating plane between the UAV and each obstacle, σ is modelled as singleton set representing the UAV position ($\sigma = \{\boldsymbol{P}_{cg}\}$) and φ_j denotes the vertices of j polytopic obstacle ($\varphi_j = \{\boldsymbol{v}^j_{o_1},\ \boldsymbol{v}^j_{o_2},\ ...,\ \boldsymbol{v}^j_{o_{n_j}}\}$). To maximize the safe region, it is assumed the separating plane corresponding to each obstacle intersects with the closest vertex, i.e, $\min_{\boldsymbol{v}^j_{o_i} \in \varphi_j} |\alpha_j^T \boldsymbol{v}^j_{o_i} + \eta_j| = 0$. Without loss of generality, it is assumed $|\alpha_j^T \boldsymbol{p}_{cg} + \eta_j| = 1$ and the SVM algorithm to find the j-th separating plane is formulated as

$$\min_{\alpha_j, \eta_j} \quad \boldsymbol{\alpha_j^T \alpha_j} \tag{7.6a}$$

$$\text{s.t.} : \boldsymbol{\alpha_j^T} \boldsymbol{p}_{cg} + \eta_j \geq 1 \tag{7.6b}$$

$$\boldsymbol{\alpha_j^T} \boldsymbol{v}^j_{o_i} + \eta_j \leq 0 \qquad i = 1, ..., n_j \tag{7.6c}$$

The polytopic approximation of obstacle-free region is determined by the intersection of half-spaces created by separating planes, i.e., $\mathbb{P}_p = \mathbb{P}(\boldsymbol{A}_p, \boldsymbol{b}_p)$ where $\boldsymbol{A}_p[k] = [\alpha_1,\ \alpha_2,\ ...,\ \alpha_{n_o}]^T$ and $\boldsymbol{b}_p[k] = [\eta_1,\ \eta_2,\ ...,\ \eta_{n_o}]^T$. Furthermore, polytopic velocity constraints are imposed by the user for safety, i.e., $\mathbb{P}_v = \mathbb{P}(\boldsymbol{A}_v, \boldsymbol{b}_v)$ where $\boldsymbol{A}_v \in \mathbb{R}^{m_v \times n_v}$ and $\boldsymbol{b}_v \in \mathbb{R}^{m_v}$ are the parameters of polytopic velocity set and m_v is the number of the faces of the polytope. Given the polytopic obstacle-free region and velocity set, the safe region can be modelled by polytope $\mathbb{X}_s[k] = \mathbb{P}_{x_s}[k]$ with $\mathbb{P}_{x_s}[k] = \mathbb{P}(\boldsymbol{A}_x[k], \boldsymbol{b}_x[k])$

$$\boldsymbol{A}_x[k] = \begin{bmatrix} \boldsymbol{A}_p[k] & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_v \end{bmatrix}, \ \boldsymbol{b}_x[k] = \begin{bmatrix} \boldsymbol{b}_p[k] \\ \boldsymbol{b}_v \end{bmatrix} \tag{7.7}$$

7.2.2 MPC-based Collision Avoidance

The actuator limits are modelled by a polytopic feasible input set $\mathbb{U} = \mathbb{P}(\boldsymbol{A}_u, \boldsymbol{b}_u)$ where $\boldsymbol{A}_u \in \mathbb{R}^{m_u \times n_u}$, $\boldsymbol{b}_u \in \mathbb{R}^{m_u}$ and m_u is the number of faces of polytope. Therefore, the MPC in (7.3a-7.3e) is formulated as quadratic program with linear constraints

$$\min_{\boldsymbol{u}[0,N-1]|k} \sum_{i=0}^{N-1} (\boldsymbol{u}[i|k] - \boldsymbol{u}_h[i|k])^T \boldsymbol{R}_u[i|k](\boldsymbol{u}[i|k] - \boldsymbol{u}_h[i|k]) \tag{7.8a}$$

$$\text{s. t.} : \boldsymbol{x}[i+1|k] = \boldsymbol{A}_d \boldsymbol{x}[i|k] + \boldsymbol{B}_d \boldsymbol{u}[i|k] \tag{7.8b}$$

$$\boldsymbol{v}[N|k] = \boldsymbol{0}_3 \tag{7.8c}$$

$$\boldsymbol{A}_u \boldsymbol{u}[i|k] \preceq \boldsymbol{b}_u \tag{7.8d}$$

$$\boldsymbol{A}_x[k] \boldsymbol{x}[i|k] \preceq \boldsymbol{b}_x[k] \tag{7.8e}$$

where $\boldsymbol{A}_u \in \mathbb{R}^{m_u \times n_u}$ and $\boldsymbol{b}_u \in \mathbb{R}^{m_u}$ are the parameters of polytopic nominal input set. $\boldsymbol{A}_x[k] \in \mathbb{R}^{m_x \times n_x}$ and $\boldsymbol{b}_x \in \mathbb{R}^{m_x}$ are the parameters of polytopic safe region for nominal system.

7.2.3 Recursive feasibility and stability

Recursive feasibility and stability are highly desirable characteristics that should be constructed into the MPC-based collision avoidance. Stability of the system is largely dependent on operator manual behaviour. Despite extensive research in this area, a full understanding of the human control behaviour remains illusive [70]. However, simple dynamic systems such as single or double integrators with/without disturbances are stabilizable by a human operator. They are often used as standard plant model to investigate the human cognitive and control behaviour in many studies [33, 65, 66, 91]. In this thesis, combination of a low-level inner-loop controller and mapping function render the UAV dynamics to that of a single integrator dynamic system. The collision avoidance assistance algorithms acts as a bounded disturbance added to the operator's commanded acceleration, i.e., see Fig. 3.5. The problem of UAV teleoperation with automated collision avoidance can be viewed as that of a manual control of single integrator dynamic system with disturbances by the human operator. The MPC-based collision avoidance is formulated in such a way that it would minimize interference with operator's commands.

In the MPC-based collision avoidance, the optimization problem is repeatedly solved to find the optimal corrective inputs that would help the operator avoid collisions with obstacles. A fundamental problem in finite-time MPC is the lack of guaranteed recursive feasibility, i.e., the feasibility of the MPC at the initial time would not guarantee its feasibility at all future iterations. This could place the UAV into a state where collision with obstacles would become unavoidable. Recursive feasibility can be guaranteed by adding terminal equality constraints to the MPC optimization problem formulation.

Proposition 1. *Adding zero velocity terminal constraints (7.31c) would guarantee recursive feasibility of the MPC-based collision avoidance assistance presented in (7.31a-7.31e) for time-invariant state constraints ($\mathbb{P}_{x_s}[k] = \mathbb{P}_{x_s}$).*

Proof. Assume the feasibility of $\boldsymbol{x}[0|k]$ and let $\boldsymbol{u}^*[0, N-1]|k = [\boldsymbol{u}^*[0|k], \boldsymbol{u}^*[1|k], ..., \boldsymbol{u}^*[N-1|k]]$ be the optimal control sequence at k-th sample. Considering zero velocity terminal constraints, there exists a feasible control sequence $\boldsymbol{u}[0, N-1]|k+1 = [\boldsymbol{u}^*[1|k], \boldsymbol{u}^*[2|k], ..., \boldsymbol{u}^*[N-1|k],\ \mathbf{0}]$ that keeps the system at previous states ($\boldsymbol{x}[N|k+1] = \boldsymbol{x}[N|k] \in \mathbb{P}_{x_s}$). Therefore, existence of solution and recursive feasibility is guaranteed.

Using the above proposition, recursive feasibility is guaranteed in the MPC-based collision avoidance with time-invariant state constraints. However, the safe convex region generated by the SVM algorithm is updated at each iteration making the state constraints time-varying. If the MPC becomes infeasible due to time-varying state constraint at (k+1)-th iteration, state constraints at (k)-iteration can be used instead.

7.3 Collision Avoidance in UAV under Uncertainty

In real-world scenarios, uncertainties are unavoidable and can significantly degrade the performance of collision avoidance algorithm. In the presence of uncertainties, the optimization problem may become infeasible resulting in collisions. Robust model predictive controllers (RMPC) have been developed to deal with uncertainties in the predictive control of dynamical systems. Worst-case MPC is a popular approach in this category (Chapter 6). It finds a conservative solution by considering the worst-case scenario for the uncertainties. The method is ill-suited for use with unstable dynamic systems [80] due to fast-growing uncertainty in the system states, which can lead to the infeasibility of optimization-based collision avoidance. Furthermore, guaranteeing recursive feasibility is challenging using this method.

An alternative method to address uncertainty is tube-based MPC introduced in [54] for a linear dynamic system under bounded disturbances. By generating control laws instead of control actions, this method separates the robustness problem from the MPC problem. A shared control strategy presented in Fig. 7.1 is introduced to incorporate this method in collision avoidance assistance. Tube-based MPC is used to account for disturbances in the formulation of collision avoidance assistance control algorithm. By introducing a feedback controller, the robustness problem is separated from the MPC problem, i.e, the MPC is solved for nominal system and the resulting input is modified by the feedback control law.

The feedback control law is defined as $\boldsymbol{u}[k] = \bar{\boldsymbol{u}}[k] + \boldsymbol{K}(\boldsymbol{x}[k] - \bar{\boldsymbol{x}}[k])$ where $\boldsymbol{K} \in \mathbb{R}^{n_u \times n_x}$ is the control feedback gain. $\bar{\boldsymbol{x}}[k] \in \mathbb{R}^{n_x}$ $\bar{\boldsymbol{u}}[k] \in \mathbb{R}^{n_u}$ are the nominal states and inputs of the UAV.

$$\bar{\boldsymbol{x}}[k+1] = \boldsymbol{A}_d \bar{\boldsymbol{x}}[k] + \boldsymbol{B}_d \bar{\boldsymbol{u}}[k] \tag{7.9}$$

Defining the controller error as $\boldsymbol{e}[k] = \boldsymbol{x}[k] - \bar{\boldsymbol{x}}[k]$, the controller error dynamics can be written as

$$\boldsymbol{e}[k+1] = \boldsymbol{A}_d \boldsymbol{e}[k] + \boldsymbol{w}_d[k] \tag{7.10}$$

where the feedback control gain matrix $\boldsymbol{K}$ is selected such that the eigenvalues of $\boldsymbol{A}_k = \boldsymbol{A}_d + \boldsymbol{B}_d \boldsymbol{K}$ are in unit ball. $\boldsymbol{w}_d[k] \in \mathbb{W}_d[k]$ with $\boldsymbol{w}_d[k] = \boldsymbol{D}\boldsymbol{w}[k]$ and $\mathbb{W}_d[k] = \boldsymbol{D}\mathbb{W}$. Given strictly stable $\boldsymbol{A}_d$ and $\mathbb{W}_d \in ComC(\mathbb{R}^{n_x})$ existence of minimum RPI set $\bar{\mathbb{S}} \in ComC(\mathbb{R}^{n_x})$ for system (7.10) is guaranteed [76], i.e, if $e[0] \in \bar{\mathbb{S}}$, then $e[k] \in \bar{\mathbb{S}}$ for all $k \in \mathbb{N}$.

To ensure the UAV follows the operator's commands as close as possible while avoiding the collision, a tube-based MPC for the nominal system is formulated as

$$\min_{\bar{\boldsymbol{u}}[0,N-1]|k} \sum_{i=0}^{N-1} (\bar{\boldsymbol{u}}[i|k] - \boldsymbol{u}_h[i|k])^T \boldsymbol{R}_u[i|k](\bar{\boldsymbol{u}}[i|k] - \boldsymbol{u}_h[i|k]) \tag{7.11a}$$

$$\text{s. t.} : \bar{\boldsymbol{x}}[i+1|k] = \boldsymbol{A}_d \bar{\boldsymbol{x}}[i|k] + \boldsymbol{B}_d \bar{\boldsymbol{u}}[i|k] \tag{7.11b}$$

$$\boldsymbol{v}[N|k] = \mathbf{0}_3 \tag{7.11c}$$

$$\bar{\boldsymbol{u}}[i|k] \in \mathbb{U} \ominus \boldsymbol{K}\bar{\mathbb{S}} \tag{7.11d}$$

$$\bar{\boldsymbol{x}}[i|k] \in \mathbb{X}_s[i|k] \ominus \bar{\mathbb{S}} \tag{7.11e}$$

where $\bar{\boldsymbol{u}}[i|k], \boldsymbol{u}_h[i|k] \in \mathbb{R}^{n_u}$ are the UAV nominal inputs and predicted operator commands at i-th sample with respect to k-th sample (current inputs), respectively. $\boldsymbol{x}[i|k] \in \mathbb{R}^{n_x}$ is UAV states, $\boldsymbol{R}_u[i|k] \in \mathbb{R}^{n_u \times n_u}$, and N is the number of samples within the MPC time horizon. Constraints related to system dynamics are represented in (7.11b). Terminal velocity constraints are presented in (7.11c) where $\boldsymbol{v}[N|k] \in \mathbb{R}^3$ is the velocity vector of the UAV at the terminal sample. Tightened input constrains are formulated in (7.11d) where $\mathbb{U}$ is the set of admissible inputs. Tightened state constraints including obstacle avoidance constraints and velocity constrains are presented in (7.11e) where $\mathbb{X}_s[i|k]$ represents state constraints. Here $\ominus$ denotes the Pontryagin difference operation.

To solve the optimization-based collision avoidance assistance and find the global optimum solution in real-time, input and state constraints should be formulated by linear inequality constraints. First, an ellipsoidal approximation of the minimum RPI set is obtained for the given set of all possible disturbances. The obstacle-free region is approximated by polytope $\mathbb{P}(\boldsymbol{A}_x, \boldsymbol{b}_x)$ using the SVM algorithm presented in Section 9.1.1. Finally, inner polytopic approximations of tightened input and state constraints are calculated.

7.3.1 Invariant Set Approximation

To utilize tube-based MPC, the minimum RPI set ($\bar{\mathbb{S}}$) should be determined. RPI set $\mathbb{S}[k]$ for discrete-time linear system (7.1) is given by [76].

$$\mathbb{S}[k] = F_k \oplus \lambda^k (1-\lambda)^{-1} \mu \mathbb{L}, \;\; F_0 = \{\mathbf{0}_{n_x}\} \tag{7.12}$$

where $\mathbb{L} \in ComCP(\mathbb{R}^{n_x})$ with $ComCP(\mathbb{R}^{n_x})$ denotes the collection of convex and compact set containing the origin in its interior, $\lambda \in [0, 1]$ is a scaler satisfying $\boldsymbol{A}_d^k \mathbb{L} \subseteq \lambda^k \mathbb{L}$ for all $k \in \mathbb{N}$. μ is defined as

$$\mu = \min_{\alpha} \{\alpha : \forall k \in \mathbb{N}, \;\; \alpha \geq 0, \;\; \mathbb{W}_d \subseteq \alpha \mathbb{L}\} \tag{7.13}$$

$\mathbb{S}[k]$ in (7.12) converges geometrically to minimum RPI set $\bar{\mathbb{S}} = \lim\limits_{k \to \infty} F_k$. Moreover, it is guaranteed $\bar{\mathbb{S}} \subseteq \mathbb{S}[k] \subseteq \bar{\mathbb{S}} + \epsilon$ where ϵ calculated as [76]

$$\epsilon = \lambda^k (1-\lambda)^{-1} \mu L \tag{7.14}$$

To find the RPI set at each sample, the corresponding exact reach set of the system should be calculated. Finding the exact reach set can be computationally prohibitive especially for a large value of k. To overcome this problem, the reach set can be approximated by geometrical shapes. In this work, the reach set at each sample time is approximated by minimum volume enclosing ellipsoid introduced in [72, 73]. In this method, the ellipsoidal input set $\mathbb{W} = \mathbb{E}_w$ with $\mathbb{E}_w = \mathbb{E}(\boldsymbol{q}_w, \boldsymbol{Q}_w)$ is considered for the system

$$\mathbb{E}_w = \{\boldsymbol{x} : (\boldsymbol{x} - \boldsymbol{q}_w)^T \boldsymbol{Q}_w^{-1} (\boldsymbol{x} - \boldsymbol{q}_w) \leq 1\} \tag{7.15}$$

where $\boldsymbol{q}_w \in \mathbb{R}^{n_w}$ and $\boldsymbol{Q}_w \in \mathbb{R}^{n_w \times n_w}$ are the center and shape matrix of the ellipsoidal input set, respectively.

The parameters of the ellipsoidal approximation of the reach set ($\mathbb{X}[k] \subseteq \mathbb{E}(\boldsymbol{q}_x[k], \boldsymbol{Q}_x[k])$ with zero initial condition are given by [73]

$$\boldsymbol{q}_x[k+1] = \boldsymbol{A}_d \boldsymbol{q}_x[k] + \boldsymbol{D}_d \boldsymbol{q}_{w_d}, \ \boldsymbol{q}[0] = \boldsymbol{0}_{n_x} \tag{7.16}$$

$$\boldsymbol{Q}_x[k+1] = \boldsymbol{P}^{-1}[k+1] \tag{7.17}$$

where $\boldsymbol{P}[k+1] \in \mathbb{R}^{n_x \times n_x}$ is the only solution of recursive equation $\boldsymbol{P}_{n+1} = \Lambda(\boldsymbol{P}_n)$

$$\Lambda(\boldsymbol{P}_n[k+1]) = \left(\int_{t_0}^{t_{k+1}} \sqrt{Tr\left(\boldsymbol{P}_n \bar{\boldsymbol{Q}}_{w_d}^T(\tau)\right)} d\tau \int_{t_0}^{t_{k+1}} \frac{\bar{\boldsymbol{Q}}_{w_d}(\tau)}{\sqrt{Tr\left(\boldsymbol{P}_n \bar{\boldsymbol{Q}}_{w_d}^T(\tau)\right)}} d\tau \right)^{-1} \tag{7.18}$$

where t_{k+1} is the time instance corresponding to $(k+1)$-th sample, and $\bar{\boldsymbol{Q}}_{w_d}(\tau) \in \mathbb{R}^{n_x \times n_x}$ is defined as

$$\bar{\boldsymbol{Q}}_{w_d}(\tau) = \Phi(t_{k+1}, \tau) \boldsymbol{D}_d \boldsymbol{Q}_w(\tau) (\Phi(t_{k+1}, \tau) \boldsymbol{D}_d)^T \tag{7.19}$$

Starting from an initial positive-definite matrix $\boldsymbol{P}_0$, the recursive equation (7.18) converges to a unique solution [72]. Here it is assumed $\boldsymbol{P}_0 = \boldsymbol{I}_{n_x}$ and the termination condition for (7.18) is defined as

$$\mu(\boldsymbol{P}_{n+1}, \boldsymbol{P}_n) = \frac{(|det\boldsymbol{P}_{n+1})^{-1/2} - (det\boldsymbol{P}_n)^{-1/2}|}{(det\boldsymbol{P}_n)^{-1/2}} \tag{7.20}$$

Here det and $|.|$ denote the determinant of matrix and the absolute value of scalar, respectively. The termination condition is satisfied when the normalized rate of change in the volume of ellipsoid is less than the desired tolerance.

An outer approximation of RPI set in (7.12) is derived by approximating the reach set with minimum volume ellipsoid enclosing it

$$\mathbb{S}'[k] = \mathbb{E}_x[k] \oplus \lambda^k(1-\lambda)^{-1}\mu\mathbb{L},\ \mathbb{E}_x[0] = \{\mathbf{0}_{n_x}\} \tag{7.21}$$

where μ is defined in (7.13).

The set $\mathbb{L}$ in (7.21) can be obtained by ellipsoidal set $\mathbb{E}_L = \mathbb{E}_L(\mathbf{0}_{n_x}, \boldsymbol{Q}_L)$ with center $\mathbf{0}_{n_x}$ and shape matrix $\boldsymbol{Q}_L \in \mathbb{R}^{n_x \times n_x}$ satisfying the discrete-time Lyapunov inequality [76]

$$\boldsymbol{A}_d^T\boldsymbol{Q}_L\boldsymbol{A}_d \leq \lambda^2\boldsymbol{Q}_L,\ \lambda \in [0,1) \tag{7.22}$$

Given $\mathbb{L} = \mathbb{E}_L$, then $\mathbb{S}'[k] = \mathbb{E}_x[k] \oplus \lambda^k(1-\lambda)^{-1}\mu\mathbb{E}_L$ which is the Minkowski sum of two ellipsoids. Calculating the exact value of $\mathbb{S}'[k]$ is computationally prohibitive. However, this set can be approximated by minimum volume enclosing ellipsoid. Consider two arbitrary ellipsoidal sets $\mathbb{E}_1 = \mathbb{E}(\boldsymbol{q}_1, \boldsymbol{Q}_1)$ and $\mathbb{E}_2 = \mathbb{E}(\boldsymbol{q}_2, \boldsymbol{Q}_2)$, the Minkowski sum of these ellipsoidal sets can be approximated by minimum volume ellipsoid enclosing it, i.e, $\mathbb{E}_1 \oplus \mathbb{E}_2 \subseteq \mathbb{E}$ with $\mathbb{E} = \mathbb{E}(\boldsymbol{q}, \boldsymbol{Q})$. The parameters of this ellipsoid are given by [43]

$$\boldsymbol{q} = \boldsymbol{q}_1 + \boldsymbol{q}_2 \tag{7.23}$$

$$\boldsymbol{Q} = (1+\beta^{-1})\boldsymbol{Q}_1 + (1+\beta)\boldsymbol{Q}_2 \tag{7.24}$$

where β is derived based on the recursive equation

$$\beta_{n+1} = \sqrt{\frac{\sum_{i=1}^{n_x}(1+\beta_n\lambda_i)^{-1}}{\sum_{i=1}^{n_x}\lambda_i(1+\beta_n\lambda_i)^{-1}}} \tag{7.25}$$

Here λ_i is the i-the eigenvalue of $(\boldsymbol{Q}_1)^{-1}\boldsymbol{Q}_2$.

Defining $\mathbb{E}_1 = \mathbb{E}[k]$ and $\mathbb{E}_2 = \lambda^k(1-\lambda)^{-1}\mu\mathbb{E}_L$ and using (7.23) and (7.24), $\mathbb{S}'[k]$ is approximated by minimum volume enclosing ellipsoid $\mathbb{E}_s[k]$

$$\mathbb{S}'[k] = \mathbb{E}_x[k] \oplus \lambda^k(1-\lambda)^{-1}\mu\mathbb{E}_L \subseteq \mathbb{E}_s[k] \tag{7.26}$$

Proposition 2. *The Ellipsoidal set $\mathbb{E}_s[k]$ is the outer approximation of RPI set $\mathbb{S}[k]$ for all $k \in \mathbb{N}$. Moreover, $\mathbb{E}_s[k]$ approaches to the minimum volume ellipsoid enclosing the minimum RPI set as $k \to \infty$*

Proof. Since $\mathbb{X}[k] = F_k \subseteq \mathbb{E}[k]$ and $\mathbb{S}[k] = \mathbb{F}_k \oplus \lambda^k(1-\lambda)^{-1}\mu\mathbb{L}$, it is concluded $\mathbb{S}[k] \subseteq \mathbb{S}'[k]$ for all $k \in \mathbb{N}$. Given $\mathbb{L} = \mathbb{E}_L$ and using (7.26), it is concluded $\mathbb{S}'[k] \subseteq \mathbb{E}_s[k]$ for all $k \in \mathbb{N}$. Therefore, $\mathbb{S}[k] \subseteq \mathbb{E}_s[k]$ for all $k \in \mathbb{N}$.

Since $\lim\limits_{k\to\infty}\mathbb{E}_s[k] = \lim\limits_{k\to\infty}\mathbb{E}[k]$ and $\lim\limits_{k\to\infty}\mathbb{E}[k]$ is the minimum volume ellipsoid enclosing the minimum RPI set $\bar{\mathbb{S}} = \lim\limits_{k\to\infty}F_k$, it is concluded $\lim\limits_{k\to\infty}\mathbb{E}_s[k]$ is the minimum volume ellipsoid enclosing the minimum RPI set.

Using (7.14) and (7.26), the outer ellipsoidal approximation of minimum RPI set

($\bar{\mathbb{E}}_s = \mathbb{E}_s[k_\epsilon]$) for the system (7.10) at arbitrary ϵ can be determined.

7.3.2 Ellipsoidal Tube-based MPC

To formulate the tube-based MPC (7.11a-7.11e) as convex optimization problem with linear constraints, $\mathbb{U} \ominus \boldsymbol{K}\bar{\mathbb{S}}$ and $\mathbb{X}_s[i|k] \ominus \bar{\mathbb{S}}$ should be approximated by polytopic sets. The feasible input set is modelled by polytopic set $\mathbb{U} = \mathbb{P}(\boldsymbol{A}_u, \boldsymbol{b}_u)$ where $\boldsymbol{A}_u \in \mathbb{R}^{m_u \times n_u}$, $\boldsymbol{b}_u \in \mathbb{R}^{m_u}$ and m_u is the number of faces of polytope. Using the SVM algorithm presented in Section 9.1.1, the safe region is modelled by the polytope $\mathbb{X}_s[k] = \mathbb{P}(\boldsymbol{A}_x, \boldsymbol{b}_x)$ where $\boldsymbol{A}_x[k]$ and $\boldsymbol{b}_x[k]$ are defined in (7.7).

Using Proposition 2, the RPI sets $\bar{\mathbb{S}}$ is approximated by outer ellipsoid $\bar{\mathbb{E}}_s$. The following proposition is introduced to approximate the constraints in (7.11d) and (7.11e) by a polytopic set.

Proposition 3. *Consider the Pontryagin difference $\mathbb{Z} = \mathbb{P}_x \ominus \mathbb{E}_y$ where $\mathbb{P}_x = \mathbb{P}(\boldsymbol{A}_x, \boldsymbol{b}_x)$ and set $\mathbb{E}_y = \mathbb{E}(\boldsymbol{q}_y, \boldsymbol{Q}_y)$ are polytopic and ellipsoidal sets, respectively. The inner polytopic approximation of set $\mathbb{Z} \supseteq \mathbb{P}z$ with $\mathbb{P}_z = \mathbb{P}(\boldsymbol{A}_z, \boldsymbol{b}_z)$ is derived*

$$\boldsymbol{A}_z = \boldsymbol{A}_x, \; \boldsymbol{b}_z = \boldsymbol{b}_x - \boldsymbol{A}_x \boldsymbol{q}_y - ||\boldsymbol{A}_x \boldsymbol{R}_y||_{\bullet} \tag{7.27}$$

where $\boldsymbol{R}_y$ is the Cholesky factorization of $\boldsymbol{Q}_y$ and $||\boldsymbol{A}_x \boldsymbol{R}_y||_{\bullet}$ is defined as

$$||\boldsymbol{A}_x \boldsymbol{R}_y||_{\bullet} = [||\boldsymbol{A}_x^{1,*} \boldsymbol{R}_y||_2, \; ||\boldsymbol{A}_x^{2,*} \boldsymbol{R}_y||_2, \; ..., \; \boldsymbol{A}_x^{n,*} \boldsymbol{R}_y||_2]^T \tag{7.28}$$

where $\boldsymbol{A}_x^{1,*}$ is the first row (out of n) of $\boldsymbol{A}_x$.

Proof. The set $\mathbb{Z}$ is obtained as Pontryagin difference of polytopic and ellipsoidal sets $\mathbb{Z} = \{\boldsymbol{z} : |\boldsymbol{z} \oplus \mathbb{E}_{\boldsymbol{y}} \subseteq \mathbb{P}_{\boldsymbol{x}}\}$. Defining the parameter of polytope as $\mathbb{P}_x = \mathbb{P}(\boldsymbol{A}_x, \boldsymbol{b}_x)$, the set Z is obtained as

$$\mathbb{Z} = \{\boldsymbol{z} : |\boldsymbol{A}_x(\boldsymbol{z}+\boldsymbol{y}) \preceq \boldsymbol{b}_x,\ \boldsymbol{y} \in \mathbb{E}(\boldsymbol{q}_y, \boldsymbol{Q}_y)\} \tag{7.29}$$

Substituting ellipsoidal equation $\boldsymbol{y} = \boldsymbol{q}_y + \boldsymbol{R}_y\boldsymbol{w},\ ||\boldsymbol{w}||_2 \leq 1$ in (7.29) and considering $\boldsymbol{A}_x^{i,*}\boldsymbol{R}_y\boldsymbol{w} \leq ||\boldsymbol{A}_x^{i,*}\boldsymbol{R}_y||_2$ for all $||w|| \leq 1$, it is concluded that if

$$\boldsymbol{A}_x\boldsymbol{z} \preceq \boldsymbol{b}_x - \boldsymbol{A}_x\boldsymbol{q}_y - ||\boldsymbol{A}_x\boldsymbol{R}_y||_{\bullet} \text{ then } \boldsymbol{A}_x\boldsymbol{z} \preceq \boldsymbol{b}_x - \boldsymbol{A}_x\boldsymbol{y} \tag{7.30}$$

Using Proposition 3, the constraints in (7.11d) and (7.11e) are approximated by inner polytopic sets $\mathbb{P}_{\bar{u}}$ and $\mathbb{P}_{\bar{x}}$, respectively. Therefore, the tube-based MPC in (7.11a-7.11e) is formulated as a quadratic program with linear constraints

$$\min_{\bar{\boldsymbol{u}}[0,N-1]|k} \sum_{i=0}^{N-1} (\bar{\boldsymbol{u}}[i|k] - \boldsymbol{u}_h[i|k])^T \boldsymbol{R}_u[i|k](\bar{\boldsymbol{u}}[i|k] - \boldsymbol{u}_h[i|k]) \tag{7.31a}$$

$$\text{s. t.} : \bar{\boldsymbol{x}}[i+1|k] = \boldsymbol{A}_d\bar{\boldsymbol{x}}[i|k] + \boldsymbol{B}_d\bar{\boldsymbol{u}}[i|k] \tag{7.31b}$$

$$\boldsymbol{v}[N|k] = \boldsymbol{0}_3 \tag{7.31c}$$

$$\boldsymbol{A}_{\bar{u}}\bar{\boldsymbol{u}}[i|k] \preceq \boldsymbol{b}_{\bar{u}} \tag{7.31d}$$

$$\boldsymbol{A}_{\bar{x}}[k]\bar{\boldsymbol{x}}[i|k] \preceq \boldsymbol{b}_{\bar{x}}[k] \tag{7.31e}$$

where $\boldsymbol{A}_{\bar{u}} \in \mathbb{R}^{m_u \times n_u}$ and $\boldsymbol{b}_{\bar{u}} \in \mathbb{R}^{m_u}$ are the parameters of polytopic nominal input set. $\boldsymbol{A}_{\bar{x}}[k] \in \mathbb{R}^{m_x \times n_x}$ and $\boldsymbol{b}_{\bar{x}} \in \mathbb{R}^{m_x}$ are the parameters of polytopic safe region for nominal system.

7.4 Experiment

A schematic view of the experimental setup is presented in Fig. 4.3. The time horizon of the MPC-based collision avoidance assistance is $t_f = 1.5$ s, respectively. The operator acceleration commands are assumed constant over the MPC time horizon. $\boldsymbol{R}_u[i|k] = \boldsymbol{R}_u[i]$ is linearly divided between $1000\boldsymbol{I}_3$ to $\boldsymbol{I}_3$ in (7.31a).

The proposed collision avoidance assistance algorithms for UAV with/without uncertainties must solve several optimization problems. The separating plane between each obstacle and UAV is obtained using SVM derived in (7.6a-7.6c). This optimization is formulated as a quadratic program. Also, the proposed MPC-based collision avoidance assistance algorithm in (7.8a-7.8e) and (7.31a-7.31e) are formulated as quadratic program. All of these optimization problems are solved in real-time using the OSQP solver with its code generation [9, 87].

7.4.1 Collision Avoidance in UAV

Two experiments are designed to demonstrate the performance of the proposed collision avoidance algorithm with guaranteed recursive feasibility. In particular, the effects of terminal constraints in collision avoidance performance are investigated and compared with collision avoidance assistance presented in Chapter 4. The overall update rate of the MPC-based collision avoidance assistance is $T_s = 30\ ms$. the parameters of input and velocity constraints in (4.25c) and (4.25d) are selected as

$$\boldsymbol{A}_u = \boldsymbol{A}_v = [-\boldsymbol{I}_3,\ \boldsymbol{I}_3]^T \tag{7.32}$$

$$\boldsymbol{b}_u = [2,\ 0.3,\ 2,\ 2,\ 0.3,\ 2]^T \tag{7.33}$$

$$\boldsymbol{b}_v = [1.5,\ 0.3,\ 1.5,\ 1.5,\ 0.3,\ 1.5]^T \tag{7.34}$$

Two-dimensional representations of the obstacles are shown in Fig. 7.2 and Fig. 7.4 where the height of each obstacle is one meter.

In the first experiment, two obstacles are placed on two sides of the UAV as demonstrated in Fig. 7.2. This is similar to the first experiment in Chapter 4. The operator teleoperates the UAV to pass the opening between the obstacles. The corrective commands in Fig. 7.3(a) modify the operator's commands in Fig. 7.3(b) to ensure collision-free operation and guarantee recursive feasibility. A comparison of Fig. 7.3(a) with Fig.4.5(a) shows that the proposed collision avoidance acts conservatively to ensure terminal constraints satisfaction and to guarantee a feasible solution at each MPC iteration.

In the second experiment, an obstacle blocking the robot path as depicted in Fig. 7.4. This is similar to the second experiment in Chapter 4. The corrective commands are shown in Fig. 7.5(a). The modify the operator's commands to ensure the UAV stops before colliding with the obstacle. In collision avoidance assistance without guaranteed recursive feasibility, the corrective commands are only generated when the UAV is close to the obstacle as shown in Fig. 4.7(a). However, by adding the guaranteed recursive feasibility feature, the corrective commands are generated throughout the flight time to ensure the UAV satisfies the terminal constraints and

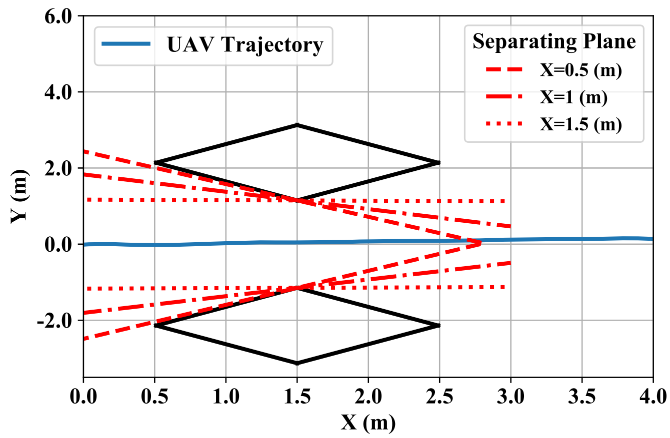

Figure 7.2: Two dimensional representation of the UAV trajectory using the proposed collision avoidance assistance.

MPC has a feasible solution at each iteration.

7.4.2 Collision Avoidance in UAV under Uncertainty

Two experiments are carried out to demonstrate the performance of the proposed collision avoidance algorithm. In particular, the utility of the new features of tube-based MPC and safe convex region generation are shown. The overall update rate of the MPC-based collision avoidance assistance is $T_s = 33\ ms$. The feedback control gain matrix is $\boldsymbol{K} = [2.02\boldsymbol{I}_3, 2.81\boldsymbol{I}_3]$ and $k_\epsilon = 3000$ to ensure $\epsilon \simeq 0$ in (7.14).

The feasible input and velocity sets are selected as cubes with origins at zero and with the edge length of $a_u = 6$ N and $a_v = 3$ m/s, respectively. Here errors in the inner loop acceleration controller are considered as the source of uncertainty in the UAV. This uncertainty is modelled by the ellipsoid $\boldsymbol{w} \in \mathbb{E}(\boldsymbol{q}_w, \boldsymbol{Q}_w)$ where the

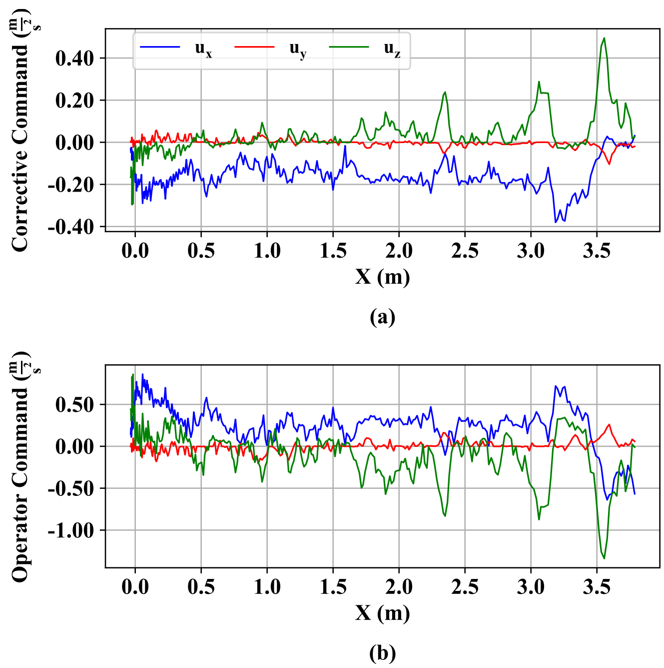

Figure 7.3: a) Corrective commands created by the proposed collision avoidance algorithm. b) Operator commands.

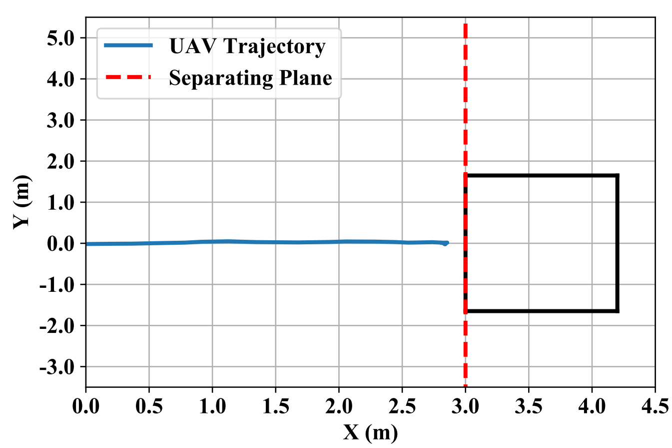

Figure 7.4: Two dimensional representation of the UAV trajectory using the proposed collision avoidance algorithm.

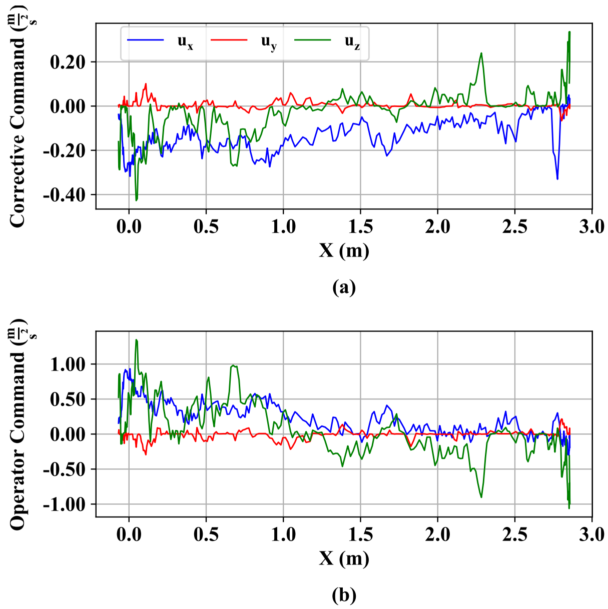

Figure 7.5: a) Corrective commands created by the proposed collision avoidance algorithm. b) Operator commands.

parameters of the ellipsoid are determined based on several experiments as

$$\boldsymbol{q}_w = \mathbf{0}_3,\ \boldsymbol{Q}_w = \mathrm{diag}(0.75, 0.75, 1.47) \tag{7.35}$$

where $diag(a_1, a_2, ..., a_n)$ is an $n \times n$ diagonal matrix. Two-dimensional representation of the obstacles are shown in Fig. 7.6 and Fig. 7.8 where the height of each obstacle is one meter.

In the first experiment, two pairs of obstacles are placed in the environment such that the path in front of the UAV is blocked by the second pair of obstacles, as demonstrated in Fig. 7.6. The operator's acceleration commands are shown in Fig. 7.7(b). The collision avoidance algorithm generates the corrective commands depicted in Fig. 7.7(a). These corrective commands help modify the UAV trajectory and assist

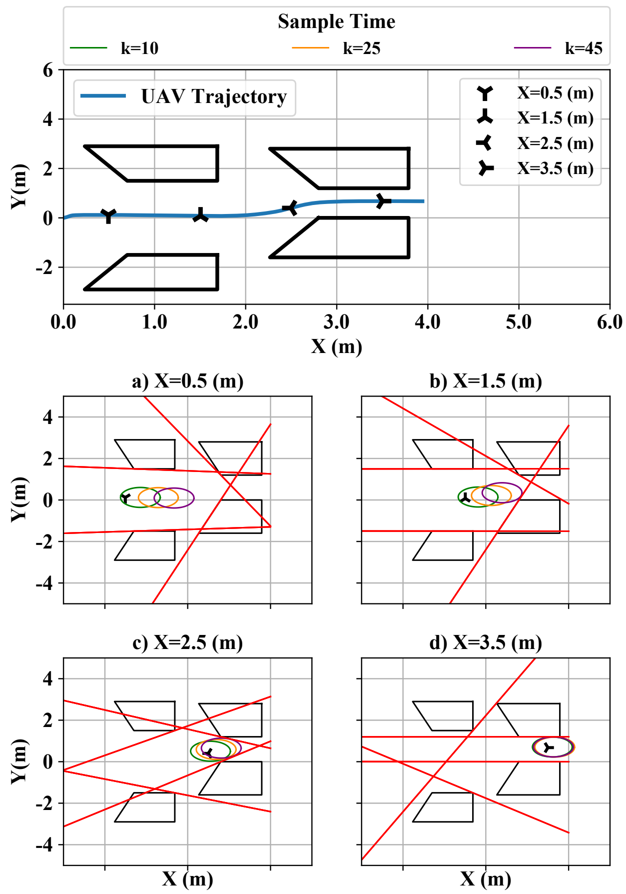

Figure 7.6: Two-dimensional representation of the UAV trajectory using the proposed collision avoidance algorithm. Two-dimensional representation of separating planes and RPI sets a) at X=0.5 (m), b) at X=1.5 (m), c) at X=2.5 (m), d) at X=3.5 (m).

the operator in guiding it through the opening between the obstacles. The RPI sets and separating planes at different sample time and UAV position are depicted in Fig. 7.6(a-d). Collision-free motion of the UAV for any disturbances is guaranteed since the ellipsoidal RPI sets are inside the polytopic safe region.

In the second experiment, three obstacles blocking the path in front of the UAV are placed in the environment. The trajectory of the UAV and the position of these obstacles are shown in Fig. 7.8. The corrective commands in Fig. 7.9(a) revise the

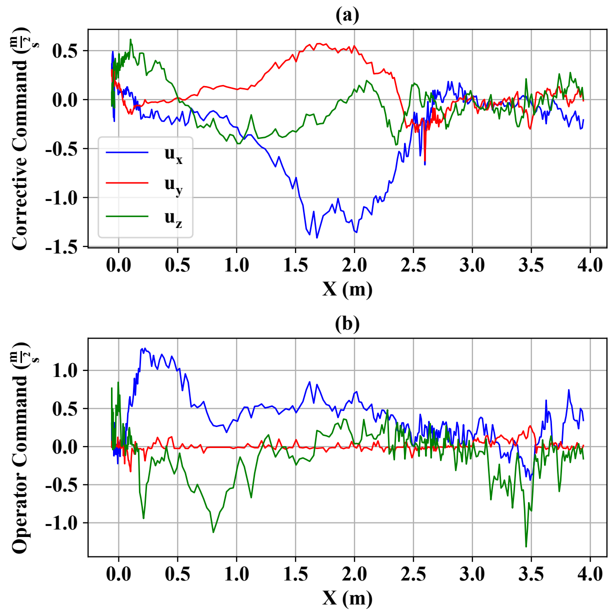

Figure 7.7: a) Corrective commands generated by the proposed collision avoidance assistance algorithm. b) Operator's commands.

operator's commands in 7.9(b) and ensure collision-free operation of the UAV for all possible disturbances. The RPI sets and separating planes at different UAV positions and sample time are depicted in 7.9(a-d). Updating the separating planes and tightened constraints in each MPC iteration yields the most suitable polytopic approximation of the obstacle-free region with respect to current UAV position. This leads to the opening the path in front of the UAV when it reaches close to the obstacles.

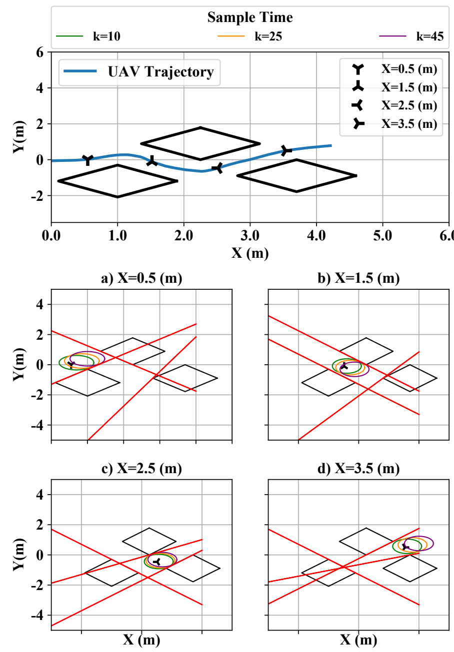

Figure 7.8: Two-dimensional representation of the UAV trajectory using the proposed collision avoidance algorithm. Two-dimensional representation of separating planes and RPI sets a) at X=0.5 (m), b) at X=1.5 (m), c) at X=2.5 (m), d) at X=3.5 (m).

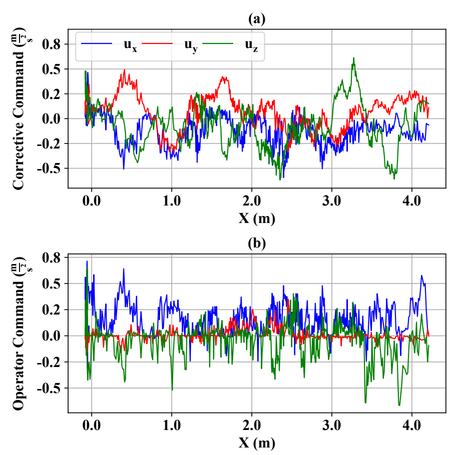

Figure 7.9: a) Corrective commands generated by the proposed collision avoidance assistance algorithm. b) Operators commands.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This thesis was concerned some key challenges of collision avoidance assistance in UAV teleoperation. A comprehensive treatment of this problem was presented in this work. This includes system model, shared control strategy, collision avoidance problem formulation, and system implementation and experimental evaluation. The shared control strategy design is intended to allow the operator focus on primary task objectives that benefit from human's cognitive and decision making capabilities. In particular, the secondary task of avoiding collisions with obstacles in the environment is delegated to an automated collision avoidance algorithm to ease operator's cognitive burden.

A new shared-control strategy for the human-in-the-loop operation of UAV was presented in Chapter 3. In this strategy, the operator teleoperates the UAV by providing acceleration commands while the collision avoidance algorithm modifies these commands to ensure collision-free operation. A mapping was introduced between the user inputs and the UAV acceleration command to facilitate manual control of the UAV. The mapping design was informed by the literature on human manual control, as well as workspace considerations for the HMI and UAV used in this research. A hierarchical control structure was introduced where the acceleration and angular velocities of the UAV are controlled by inner loop controllers, and its attitude angles are controlled by an attitude controller. This control structure simplifies the teleoperation task by rendering the closed-loop translational dynamics of the UAV into that of double-integrator system.

MPC-based collision avoidance algorithms were proposed that incorporated the UAV operational constraints and proactively prevented the operator from commanding it into a state where the collision with obstacles would become unavoidable. One of the key challenges in the formulation of the MPC optimization is the non-convexity of the constraints related to the obstacle-free space. A number of new MPC optimization formulations were proposed in this thesis that would approximate this space with a convex space. In Chapter 4, a collision avoidance algorithm based on the backward reachability analysis was presented. The analysis is used to identify obstacles with chance of collision with the UAV over the MPC time horizon. A polyhedral safe convex region was created using the SVM algorithm to separate the UAV from these obstacles. This approach led to a more realistic approximation of the obstacle-free space compared to a case where all obstacles would be considered.

Chapter 5 expanded on the work in Chapter 4 by utilizing the reachability analysis for linear dynamical systems to approximate the obstacle-free space with a safe convex region. To this end, safe half-spaces with maximum reachable region were generated for each unsafe obstacle. A polyhedral safe convex region was produced by the intersection of these half-spaces and was incorporated in the MPC-based collision avoidance optimization as linear inequality constraints. This approached yielded a comparatively less restrictive safe space for the UAV to operate, when compared to the one produced by the method in Chapter 4.

Chapter 6 presented a novel ellipsoidal RMPC to extend the collision avoidance assistance to UAV under uncertainties. The uncertainties considered were in the form of disturbances and measurement noise affecting the UAV and obstacles. This method calculates an ellipsoidal approximation of the reachable regions due to admissible inputs and uncertainty. It computes inflated polytopic obstacles considering all possible uncertainties and then generates a polyhedral approximation of the obstacle-free space using the safe convex region generation method presented in Chapter 5. It also uses geometrical relation between the ellipsoidal reach set due to uncertainties and polyhedral safe region to calculate an inner polyhedral approximation of the tightened constraints.

Chapter 7 investigated the problem of recursive feasibility in the MPC formulation of the collision avoidance algorithm. Normally, MPC formulations lack guaranteed recursive feasibility, i.e., the feasibility of the MPC at the initial time does not guarantee its feasibility at future iterations. This could potentially place the UAV into a state where collision with obstacles would become unavoidable. A novel MPC-based collision avoidance assistance algorithms with guaranteed recursive feasibility for UAV teleoperation with/without uncertainties was introduced. Recursive feasibility was guaranteed by adding proper terminal constraints to the MPC optimization formulation in Chapter 3. Furthermore, a novel ellipsoidal tube-based MPC was introduced to deal with uncertainties in the system. A tube-based MPC strategy was used to separate the robustness and MPC problems, and ensure collision-free operation for all possible disturbances on the UAV. To address the robustness problem, a feedback controller was designed to guarantee the existence of the RPI set; a reachability analysis was employed to calculate the outer ellipsoidal approximation of the RPI set. A polytopic safe convex region was created around the UAV using the SVM algorithm. The MPC optimization was formulated to keep the UAV in the safe region for all possible disturbances, while minimizing interference with the operator's commands.

All methods proposed in this thesis were implemented and their effectiveness were demonstrated in experiments under a number of different scenarios. These scenarios were designed to highlight various features of the proposed algorithms.

8.2 Future Work

There are a number of interesting avenues for further research based on the work presented in this thesis. These include but are not limited to the following problems:

- In this thesis, a uni-lateral teleoperation strategy was utilized in the shared control approach. In uni-lateral teleoperation, the operator only receives visual feedback from the task environment. Bi-lateral teleoperation brings additional feedback to the operator through force reflection via the HMI. This additional feedback channel can be used to more efficiently inform the operator of impending collisions so he/she can react and revise his/her commands accordingly. This is expected to further improve operator's situational awareness and enhance his/her overall ability in navigating the UAV in the task environment.
- This thesis focused primarily on the design of collision avoidance algorithms.

The shared control strategy may be improved by using further insight from the studies of the human manual control behaviour. This can potentially yield a more efficient shared control strategy that would decrease unnecessary workload on the operator. The reader is reffered to numbers of studies on the subject matter in [37, 90, 93, 95].

- The research in this thesis investigated a scenario where a single operator controls a single UAV in its task environment. The proposed algorithms can extend to multi-UAV operation in an obstacle-rich environment. The reachability-based collision avoidance algorithms presented in this study have a great potential to be generalized to multi-UAV systems, particularly the collision avoidance algorithm presented in Chapter 6. In the reachability-based collision avoidance algorithm for UAV with dynamic obstacles, the obstacles can easily be interpreted as other UAVs. This would allow for the formulation to be extended to multi-UAV scenarios.

- A novel ellipsoidal tube-based MPC for collision avoidance assistance in UAV teleoperation was presented in Chapter 7. In this work, the collision avoidance assistance in UAV under additive disturbances was formulated as a convex optimization problem with guaranteed recursive feasibility. This method can be extended to consider measurement noise simultaneously. The reader is refereed to a number of studies on the subject matter in [63, 64, 76]

- In this thesis, it was assumed that the configurations of the obstacles are known and the UAV position and velocity were measured by a motion tracking system in an indoor laboratory environment. However, the proposed shared control strategy only needs relative UAV positions and velocities with respect to each obstacle. On-board UAV sensors such as cameras and Lidars can be used to to obtain the environment and obstacle information in real-world scenarios.

- The operator's input commands were modelled as constant in the MPC formulations of this thesis. In reality, the operator is expected to react to obstacles in the environment. Therefore, a more realistic model of the operator behaviour can be developed and used in the MPC optimization formulation.
- In this thesis, a convex approximation of obstacle-free space is derived and the MPC-based collision avoidance is formulated as a convex optimization problem. In this method, the MPC solution is globally optimum in the approximated region. To find the optimal solution in obstacle-free space, the MPC-based collision avoidance can be formulated as a non-convex optimization problem. To reduce the computational time, the solution of the convex MPC introduced in this thesis can be used as an initial guess in the non-convex MPC.

Appendix A

System Dynamics

This chapter presents the general dynamic model of multi-rotor UAVs including under-actuated quadrotors and a fully actuated omnicopter. These include rigid body dynamics of the airframe and the dynamics of the propellers.

A.1 Rigid-Body Dynamics of the Airframe

The inertial frame (I) and the fixed-body frame mounted on the UAV (omnicopter) (B) are defined as depicted in Fig. A.1 for the purpose of developing the dynamics. In general, the airframe dynamics for under-actuated quadrotor [59] or fully actuated

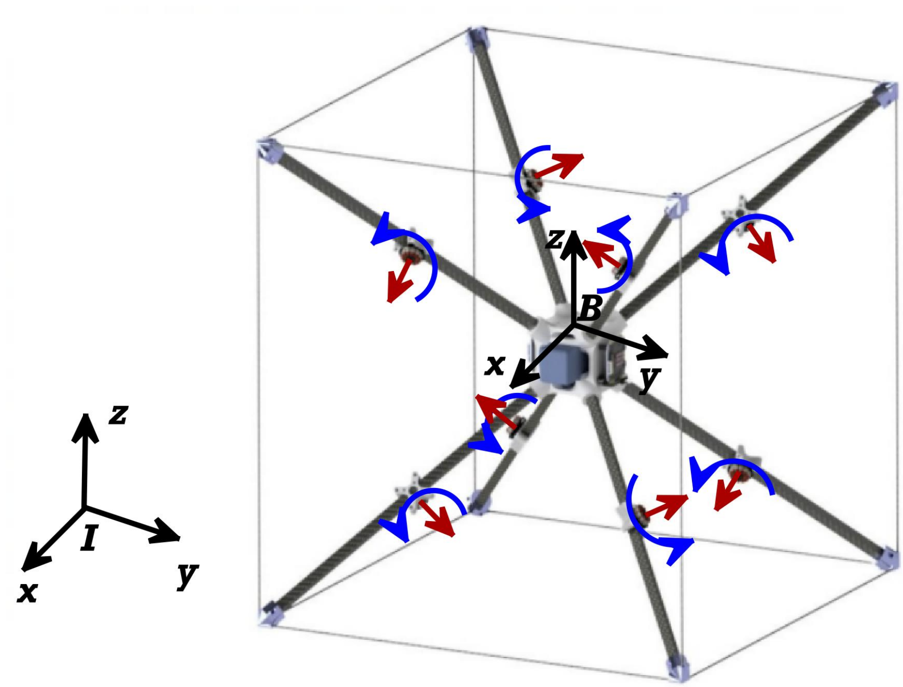

Figure A.1: The schematic view of the omnicopter with applied thrusts and torques

omnicopter [18]) can be expressed as

$$\dot{\boldsymbol{p}}_{cg} = \boldsymbol{v}_{cg} \tag{A.1a}$$

$$m\dot{\boldsymbol{v}}_{cg} = \boldsymbol{R}^T \boldsymbol{f}_b - m\boldsymbol{g} \tag{A.1b}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{W}^T(\boldsymbol{q})\boldsymbol{\omega} \tag{A.1c}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{\tau}_b \tag{A.1d}$$

where $\boldsymbol{p}_{cg} \in \mathbb{R}^3$ and $\boldsymbol{v}_{cg} \in \mathbb{R}^3$ denote the position and velocity of center of gravity of UAV, respectively. $\boldsymbol{\omega} \in \mathbb{R}^3$ is the vector of angular velocity and $\boldsymbol{q} = [q_1, q_2, q_3, q_4]^T$ is a unit quaternion representing the attitude of the body. $\boldsymbol{g} \in \mathbb{R}^3$ is the gravity vector, m is the mass of the system, $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is the moment of inertia, $\boldsymbol{f}_b \in \mathbb{R}^3$ and $\boldsymbol{\tau}_b \in \mathbb{R}^3$

are the force and torque in the body frame, respectively. $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3]$ is 3×3 rotation matrix defined by quaternions [30]

$$\boldsymbol{r}_1 = [q_0^2 + q_1^2 - q_2^2 - q_3^2,\ 2q_1q_2 - 2q_0q_3,\ 2q_1q_3 + 2q_0q_2]^T \tag{A.2}$$

$$\boldsymbol{r}_2 = [2q_1q_2 + 2q_0q_3,\ q_0^2 - q_1^2 + q_2^2 - q_3^2,\ 2q_2q_3 - 2q_0q_1]^T \tag{A.3}$$

$$\boldsymbol{r}_3 = [2q_1q_3 - 2q_0q_2,\ 2q_2q_3 + 2q_0q_1,\ q_0^2 - q_1^2 - q_2^2 + q_3^2]^T \tag{A.4}$$

W is the quaternion rate matrix defined by [30]

$$W = \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \tag{A.5}$$

The force and torque in body frame represent the net propellers force and torque applied to the UAV and can be generally linked to the propeller thrust forces using

$$\begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{\tau}_b \end{bmatrix} = \boldsymbol{M}_f \boldsymbol{f}_{prop} \tag{A.6}$$

For the omnicopter employed in this thesis, the mapping $\boldsymbol{M}_f \in \mathbb{R}^{6\times 8}$ is given by [35]

$$\boldsymbol{M}_f = \begin{bmatrix} \boldsymbol{D} \\ \boldsymbol{P} \times \boldsymbol{D} + k_r \boldsymbol{D} \end{bmatrix} \tag{A.7}$$

where $\times$ denotes the cross product, $k_r = 0.014014$ is the propeller thrust to drag ratio, $\boldsymbol{P} \in \mathbb{R}^{3\times 8}$ and $\boldsymbol{D} \in \mathbb{R}^{3\times 8}$ represent position and orientation of the rotors

$$\boldsymbol{P} = \frac{0.213}{\sqrt{3}} \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix} \tag{A.8}$$

$$\boldsymbol{D} = \begin{bmatrix} -a & b & a & -b & a & -b & -a & b \\ b & a & -b & -a & -b & -a & b & a \\ c & -c & c & -c & c & -c & c & -c \end{bmatrix} \tag{A.9}$$

where $a = 1/2 + 1/\sqrt{12}$, $b = 1/2 - 1/\sqrt{12}$ and $c = 1/\sqrt{3}$.

Remark. *To apply these dynamics to other multi-rotor UAVs, the mapping $\boldsymbol{M}_f$ must be accordingly modified.*

A.2 Propeller Dynamics

In multi-rotor UAVsd, propellers are driven by brushless DC motors, and thrust forces are created by the interaction of the propellers with airflow. The dynamics of a DC motor are given by [29, 96]

$$V_m = L_m \frac{dI_m}{dt} + R_m I_m + K_e \omega_m \tag{A.10}$$

$$\tau_m = \tau_l + \tau_d + J_r \dot{\omega_m} + b\omega_m \tag{A.11}$$

where V_m and I_m are motor input voltage and current, respectively. L_m is inductance, R_m is resistance, ω is motor angular velocity, J_r is rotor inertia, b is viscous damping coefficient of the motor and τ_d is disturbance on the system. τ_m and τ_l are motor and drag torques, respectively.

$$\tau_m = K_m I \tag{A.12}$$

$$|\tau_l| = K_t \omega_m^2 \tag{A.13}$$

where K_m and K_t are mechanical and drag torque constants, respectively.
Inductance term is usually small compared to other terms in DC motor and can be neglected [29]. Therefore, by inserting (A.10) to (A.12), the motor torque is derived

$$\tau_m = K_m \frac{V_m - K_e \omega_m}{R_m} \tag{A.14}$$

By substituting (A.13) and (A.14) in (A.11), the dynamic equation is obtained.

$$V_m = \frac{R_m J_r}{K_m}\dot{\omega} + \frac{K_t R_m}{K_m}\omega_m^2 sgn(\omega_m) + \left(\frac{bR_m}{K_m} + K_e\right)\omega_m + \frac{R_m}{K_m}\tau_d \tag{A.15}$$

Furthermore, thrust force in each propeller calculated by [59]

$$|f_{prop}| = K_f \omega_m^2 \tag{A.16}$$

where K_f is the thrust constant.
By substituting ω_m from (A.16) to (A.15), the dynamic equation of each propeller is derived

$$\alpha_1 \frac{\dot{f}_{prop}}{\sqrt{|f_{prop}|}} + \alpha_2 f_{prop} + \alpha_3 sgn(f_{prop})\sqrt{|f_{prop}|} + \alpha_4 sgn(f_{prop}) = V_m \tag{A.17}$$

where

$$\alpha_1 = \alpha_2 \frac{J_r \sqrt{k_f}}{2k_t},\ \alpha_2 = \frac{k_t R_m}{k_f k_m},\ \alpha_3 = \frac{R_m b + k_m k_e}{k_m \sqrt{k_f}},\ \alpha_4 = \frac{R_m}{k_m}\tau_d \tag{A.18}$$

For the system used in this thesis α_1-α_4 are identified in [35]

$$\alpha_1 = 1957.54, \alpha_2 = 417.57,\ \alpha_3 = 1281.51,\ \alpha_4 = -144.14 \tag{A.19}$$

Considering the symmetric shape of the omnicopter, the dynamic equation of the propellers can be linearized around the hovering mode.

$$\dot{\boldsymbol{f}}_{prop} = c_1 \boldsymbol{f}_{prop} + c_2 \boldsymbol{u}_{prop} \tag{A.20}$$

where $\boldsymbol{f}_{prop} \in \mathbb{R}^8$ and $\boldsymbol{u}_{prop} \in \mathbb{R}^8$ represent propeller forces and input voltages of the motors, respectively. c_1 and c_2 are defined as

$$c_1 = \frac{\alpha_2 sgn(f^h_{prop}) V^h_m}{\alpha_1 \left(-\alpha_3 + \sqrt{\alpha_3^2 + sgn(f^h_{prop}) 4\alpha_2 V^h_m}\right)} - \frac{3}{4\alpha_1}\left(-\alpha_3 + \sqrt{\alpha_3^2 + sgn(f^h_{prop}) 4\alpha_2 V^h_m}\right) - \frac{a_3}{a_1} \tag{A.21}$$

$$c_2 = \frac{-\alpha_3 + \sqrt{\alpha_3^2 + sgn(f^h_{prop}) 4\alpha_2 V^h_m}}{2\alpha_1\alpha_2} \tag{A.22}$$

where f^h_{prop} and V^h_m are the propeller force and the voltage of the motor in hovering mode.

$$V_m^h sgn(f_{prop}^h) = |V_{m_i}^h| = \alpha_2 |f_{prop_i}^h| + \alpha_3 \sqrt{|f_{prop_i}^h|} + \alpha_4 \tag{A.23}$$

where $\boldsymbol{f}_{prop}^h = [f_{prop_1}^h, f_{prop_2}^h, ..., f_{prop_8}^h]^T$ is calculated by using (A.6)

$$\boldsymbol{f}_{prop}^h = \boldsymbol{M}_f^\dagger \begin{bmatrix} \boldsymbol{f}_b^h \\ \boldsymbol{\tau}_b^h \end{bmatrix} \tag{A.24}$$

where $\boldsymbol{f}_b^h = [0, 0, g]^T$, $\boldsymbol{\tau}_b^h = \mathbf{0}_{3\times 1}$,and $\boldsymbol{M}_f^\dagger$ is the pseudoinverse of the mapping function.minimizing the norm of the propeller forces.

A.3 UAV Dynamics

In this section, the combined dynamics of the UAV and the propellers dynamics are determined. Taking the derivative of (A.6) and substituting it into (A.20) yield the dynamics of the net forces/torques acting on the rigid body as follows

$$\dot{\boldsymbol{f}}_b = c_1 \boldsymbol{f}_b + c_2 \boldsymbol{u}_a \tag{A.25}$$

$$\dot{\boldsymbol{\tau}}_b = c_1 \boldsymbol{\tau}_b + c_2 \boldsymbol{u}_\omega \tag{A.26}$$

where $\boldsymbol{u}_a \in \mathbb{R}^3$ and $\boldsymbol{u}_\alpha \in \mathbb{R}^3$ are mapped input voltages into the net forces/torques dynamics given by,

$$\begin{bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_\alpha \end{bmatrix} = \boldsymbol{M}_f \boldsymbol{v}_m \tag{A.27}$$

By inserting (A.25) and (A.26) in (A.1c) and (A.1d), the combined dynamics of the UAV are derived as

$$\dot{\boldsymbol{p}}_{cg} = \boldsymbol{v}_{cg} \tag{A.28}$$

$$m\dot{\boldsymbol{v}}_{cg} = \boldsymbol{R}^T \boldsymbol{f}_b - mg \tag{A.29}$$

$$\dot{\boldsymbol{f}}_b = c_1 \boldsymbol{f}_b + c_2 \boldsymbol{u}_a \tag{A.30}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{W}^T(\boldsymbol{q})\boldsymbol{\omega} \tag{A.31}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{\tau}_b \tag{A.32}$$

$$\dot{\boldsymbol{\tau}}_b = c_1 \boldsymbol{\tau}_b + c_2 \boldsymbol{u}_\omega \tag{A.33}$$

It should be noted that the above derivations assume that the propellers are identical. This model can be easily revised for the more general case in which this is not the case.

Appendix B

Reachability Analysis

Reachability analysis provides a powerful tool in identifying obstacles with a chance of collision, creating a safe convex region, and modelling uncertainties in the robot and obstacles. In a linear dynamic system, the reachable region can be approximated by predefined geometrical shapes. Due to the efficiency of ellipsoids in approximating the geometrical shapes and the simplicity of its equation, reachability analysis based on the ellipsoidal technique is one of the widely-used approaches in the linear dynamic system.

In this chapter some fundamental concepts in the reachability analysis of linear dynamic system are introduced. In particular, reachability analysis based on the intersection of external tight ellipsoids and minimum volume external ellipsoid are reviewed.

B.1 Reachability in Linear Dynamic Systems

This thesis utilizes the reachability analysis for linear dynamical systems with hard-bound on the control input. Here,

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{u} \tag{B.1}$$

where $\boldsymbol{A}(t) \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B}(t) \in \mathbb{R}^{n_x \times n_u}$ are continuous in t and $\boldsymbol{u}(t) \in \mathbb{U}(t)$ with $\mathbb{U}(t) : T_r \rightarrow \mathrm{conv}\mathbb{R}^{n_u}$ denotes a closed and convex set in $\mathbb{R}^{n_u}$ and T_r is the time interval of interest.

Reachability analysis can be categorized into forward and backward reachability. In forward reachability analysis, the reach set $\mathbb{X}(t)$ is defined as a set of all states at time t which can be reached from the initial states $\boldsymbol{x}_0 \in \mathbb{X}_0$ by admissible control inputs (see Fig. B.1) [51],

$$\mathbb{X}(t) = \{\boldsymbol{x} \in \mathbb{R}^{n_x} | \exists\ \boldsymbol{u}(.) \in \mathbb{U},\ \exists\ \boldsymbol{x}_0 \in \mathbb{X}_0,\ \boldsymbol{x} = \boldsymbol{x}(t, t_0, \boldsymbol{x}_0)\} \tag{B.2}$$

Here $\boldsymbol{x}(t, t_0, \boldsymbol{x}_0) \in \mathbb{R}^{n_x}$ is the solution of (B.1) at time t starting from initial time t_0,

$$\boldsymbol{x}(t, t_0, \boldsymbol{x}_0) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{\Phi}(t, \tau)\boldsymbol{B}(\tau)\boldsymbol{u}(\tau)d\tau \tag{B.3}$$

where $\boldsymbol{\Phi}(t, \tau) \in \mathbb{R}^{n_x \times n_x}$ is the state-transition matrix

$$\frac{\partial}{\partial t}\boldsymbol{\Phi}(t, \tau) = \boldsymbol{A}(t)\boldsymbol{\Phi}(t, \tau),\ \boldsymbol{\Phi}(\tau, \tau) = \boldsymbol{I}_n \tag{B.4}$$

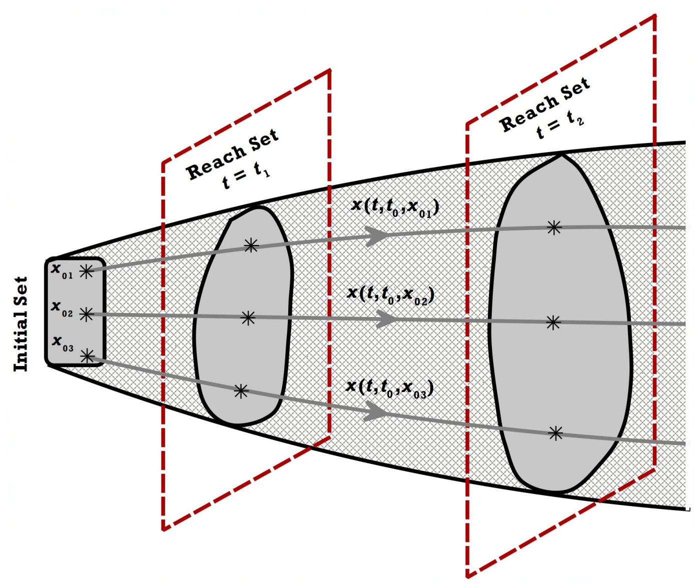

Figure B.1: Forward reach tube and reach set at different time instances with several sample trajectories

Moreover, forward reach tube $\overline{\mathbb{X}}(t)$ is defined as all the states which can be reached during the time interval of interest

$$\overline{\mathbb{X}}(t) = \cup\{\mathbb{X}(\tau) : \tau \in [t_0, t]\} \tag{B.5}$$

The reach tube can be approximated by the union of a finite number of reach sets, i.e.,

$$\overline{\mathbb{X}}(t) \approx \cup\{\mathbb{X}(\tau) : \tau \in [t_0, t_1, ..., t_k]\} \tag{B.6}$$

In backward reachability analysis (see Fig. B.2), the reach set $\mathbb{W}(t)$ is defined as the set of all states at time t that can be driven into target states $\boldsymbol{x}_f \in \mathbb{X}_f$ with

$\mathbb{X}_f \in \text{conv}\mathbb{R}^{n_x}$ at final time t_f by admissible control inputs.

$$\mathbb{W}(t) = \{\boldsymbol{w} : \exists\ \boldsymbol{u}(.) \in \mathbb{U},\ \exists\ \boldsymbol{x}_f \in \mathbb{X}_f,\ \boldsymbol{w} = \boldsymbol{x}(t, t_f, \boldsymbol{x}_f)\} \tag{B.7}$$

where $\boldsymbol{x}(t, t_f, \boldsymbol{x}_f) \in \mathbb{R}^{n_x}$ is the solution of (B.1) at time t knowing the final time t_f

$$\boldsymbol{x}(t, t_f, \boldsymbol{x}_f) = \boldsymbol{\Phi}(t, t_f)\boldsymbol{x}_f - \int_t^{t_f} \boldsymbol{\Phi}(t, \tau)\boldsymbol{B}(\tau)\boldsymbol{u}(\tau)d\tau \tag{B.8}$$

Backward reach tube $\overline{\boldsymbol{W}}(t)$ is defined as all states which can be lead to target set during the time interval of interest

$$\overline{\mathbb{W}}(t) = \cup\{\mathbb{W}(\tau) : \tau \in [t, t_f]\} \tag{B.9}$$

Backward reach tube can be approximated by the union of a finite number of reach sets, i.e,

$$\overline{\mathbb{W}}(t) = \cup\{\mathbb{W}(\tau) : \tau \in [t_k, t_{k+1}, ..., t_f]\} \tag{B.10}$$

In linear dynamic systems, the reachable region can be approximated by predefined geometrical shapes. Due to the efficiency and simplicity of ellipsoid in approximating the geometrical shapes, reachability analysis based on the ellipsoidal technique is used in this thesis. The reach set is approximated based on two methods of minimum volume external ellipsoid and intersection of external tight ellipsoids.

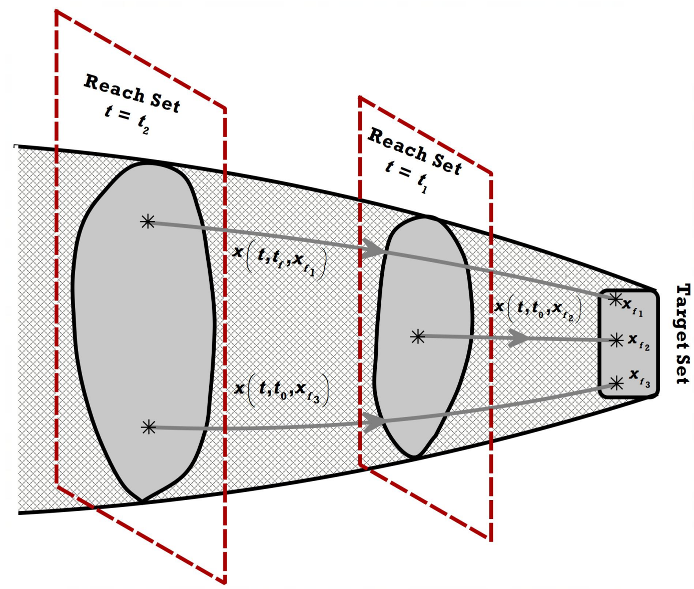

Figure B.2: Backward reach tube and reach set at different time instances with several sample trajectories

B.2 Minimum Volume External Ellipsoid

A number of criteria can be used to approximate the forward reach set by external ellipsoids. This section reviews a method that finds the minimum-volume external ellipsoid covering the reach set [23, 24, 71].

This method assumes ellipsoidal hard bounds for admissible input set and initial state set of the system in (B.1).

$$\mathbb{X}_0 = \mathbb{E}(\boldsymbol{q}_0, \boldsymbol{Q}_0), \ \mathbb{U}(t) = \mathbb{E}(\boldsymbol{q}_u(t), \boldsymbol{Q}_u(t)) \tag{B.11}$$

where $\boldsymbol{q}_0 \in \mathbb{R}^{n_x}$ and $\boldsymbol{Q}_0 \in \mathbb{R}^{n_x \times n_x}$ are the center and the shape matrix of ellipsoidal initial state set, respectively. $\boldsymbol{q}_u(t) \in \mathbb{R}^{n_u}$ and $\boldsymbol{Q}_u(t) \in \mathbb{R}^{n_u \times n_u}$ are the center and the shape matrix of ellipsoidal input set, correspondingly. Here $\mathbb{E}(\boldsymbol{q}, \boldsymbol{Q})$ represents an arbitrary ellipsoidal set.

$$\mathbb{E}(\boldsymbol{q}, \boldsymbol{Q}) = \left\{\boldsymbol{x} | (\boldsymbol{x} - \boldsymbol{q})^T \boldsymbol{Q}^{-1} (\boldsymbol{x} - \boldsymbol{q}) \leq 1\right\} \tag{B.12}$$

The forward reach set for linear dynamic system can be approximated by external ellipsoid ($\mathbb{X}(t) \subseteq E(\boldsymbol{q}_x(t), \boldsymbol{Q}_x(t))$ [23, 71]

$$\dot{\boldsymbol{q}}_x(\tau) = \boldsymbol{A}(\tau)\boldsymbol{q}_x(\tau) + \boldsymbol{B}(\tau)\boldsymbol{q}_u(\tau),\ \boldsymbol{q}_x(t_0) = \boldsymbol{q}_0 \tag{B.13}$$

$$\dot{\boldsymbol{Q}}_x(\tau) = \boldsymbol{A}(\tau)\boldsymbol{Q}_x(\tau) + \boldsymbol{Q}_x(\tau)\boldsymbol{A}^T(\tau) + \gamma(\tau)\boldsymbol{Q}_x(\tau) + \gamma^{-1}(\tau)\boldsymbol{Q}_{u_b}(\tau),\ \boldsymbol{Q}_x(t_0) = \boldsymbol{Q}_0 \tag{B.14}$$

where $\tau \in [t_0, t]$ and $\boldsymbol{Q}_{u_b}(\tau) \in \mathbb{R}^{n_x \times n_x}$

$$\boldsymbol{Q}_{u_b}(\tau) = \boldsymbol{B}(\tau)\boldsymbol{Q}_u(\tau)\boldsymbol{B}^T(\tau) \tag{B.15}$$

$\gamma(\tau)$ is the design variable obtained as

$$\gamma(\tau) = \sqrt{\frac{tr(\boldsymbol{P}(\tau)\boldsymbol{Q}_{u_b}(\tau))}{tr(\boldsymbol{P}(\tau)\boldsymbol{Q}_x(\tau))}} \tag{B.16}$$

and $\boldsymbol{P}(\tau) \in \mathbb{R}^{n_x \times n_x}$ is defined as

$$\dot{\boldsymbol{P}}(\tau) = \boldsymbol{P}(\tau)\boldsymbol{A}(\tau) - \boldsymbol{A}^T(\tau)\boldsymbol{P}(\tau),\ \boldsymbol{P}(t) = \frac{\partial L(\boldsymbol{Q}_x)}{\partial \boldsymbol{Q}_x}|_{\tau=t} \tag{B.17}$$

Here $L(\boldsymbol{Q}_x)$ is the objective function optimizing the external ellipsoid covering the reach set. By integrating (B.13) and(B.14), the parameters of external ellipsoid can be formulated as [73]

$$\boldsymbol{q}_x(t) = \boldsymbol{\Phi}(t,t_0)\boldsymbol{q}_0 + \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\boldsymbol{B}(\tau)\boldsymbol{q}_u(\tau)d\tau \tag{B.18}$$

$$\begin{aligned}\boldsymbol{Q}_x(t) = \Big(\sqrt{\langle \boldsymbol{P}_0, \boldsymbol{Q}_0\rangle} + \int_{t_0}^{t} \sqrt{\langle \boldsymbol{P}(\tau), \boldsymbol{Q}_{u_b}(\tau)\rangle} d\tau\Big) \\ \Big(\frac{\bar{\boldsymbol{Q}}_0}{\sqrt{\langle \boldsymbol{P}_0, \boldsymbol{Q}_0\rangle}} + \int_{t_0}^{t} \frac{\bar{\boldsymbol{Q}}_u(\tau)}{\sqrt{\langle \boldsymbol{P}(\tau), \boldsymbol{Q}_{u_b}(\tau)\rangle}} d\tau\Big)\end{aligned} \tag{B.19}$$

where $\boldsymbol{P}_0 = \boldsymbol{P}(t_0)$ and $\bar{\boldsymbol{Q}}_0, \bar{\boldsymbol{Q}}_u \in \mathbb{R}^{n_x \times n_x}$ are defined as

$$\bar{\boldsymbol{Q}}_u(\tau) = \boldsymbol{\Phi}(t,\tau)\boldsymbol{Q}_{u_b}\boldsymbol{\Phi}^T(t,\tau) \tag{B.20}$$

$$\bar{\boldsymbol{Q}}_0 = \boldsymbol{\Phi}(t,t_0)\boldsymbol{Q}_0\boldsymbol{\Phi}^T(t,t_0) \tag{B.21}$$

By defining $L(\boldsymbol{Q}_x) = c_n\sqrt{det\ \boldsymbol{Q}_x}$ where c_n is a constant value, the shape matrix of minimum volume external ellipsoid $\boldsymbol{Q}_x(t) = \boldsymbol{P}^{-1}(t)$ is calculated by recursive equation $\boldsymbol{P}_{n+1}(t) = \Lambda(\boldsymbol{P}_n(t))$ [72].

$$\begin{aligned}\Lambda(\boldsymbol{P}_n(t)) = \Bigg(\Big(\sqrt{\langle \boldsymbol{P}_n(t)\overline{\boldsymbol{Q}}_0\rangle} + \int_{t_0}^{t} \sqrt{\langle \boldsymbol{P}_n(t)\overline{\boldsymbol{Q}}_u(\tau)\rangle} d\tau\Big) \\ \Big(\frac{\overline{\boldsymbol{Q}}_0}{\sqrt{\langle \boldsymbol{P}_n(t)\overline{\boldsymbol{Q}}_0\rangle}} + \int_{t_0}^{t} \frac{\overline{\boldsymbol{Q}}_u(\tau)}{\sqrt{\langle \boldsymbol{P}_n(t)\overline{\boldsymbol{Q}}_u(\tau)\rangle}} d\tau\Big)\Bigg)^{-1}\end{aligned} \tag{B.22}$$

Starting from an arbitrary initial positive definite matrix $P_0(t)$, this iterative equation eventually coverages to a unique solution [72].

B.3 Intersection External Tight Ellipsoids

Reach set approximation based on tight ellipsoids [52, 53] is widely used in the reachability analysis of linear systems. Two variants of this approach approximate the reach set by the intersection of a finite number of tight internal or external ellipsoids. In this thesis, the external approximation of the reach set is of interest. Given that each tight external ellipsoid touches the reach set at an arbitrary direction, the backward reach set at each time instance of interest is approximated by the intersection of a finite number of tight ellipsoids [51, 52],

$$\mathbb{W}(t) = \cap\{\mathbb{E}(\boldsymbol{q}_w(t), \boldsymbol{Q}_w^l) \mid ||\boldsymbol{l}(t)||_2 = 1\} \tag{B.23}$$

where $\boldsymbol{q}_w(t) \in \mathbb{R}^{n_x}$ and $\boldsymbol{Q}_w^l(t) \in \mathbb{R}^{n_x \times n_x}$ are the parameters of ellipsoidal backward reach set at arbitrary direction $\boldsymbol{l}(t) \in \mathbb{R}^{n_x}$. Moreover,

$$\dot{\boldsymbol{q}}_w(t) = \boldsymbol{A}(t)\boldsymbol{q}_w(t) + \boldsymbol{B}(t)\boldsymbol{q}_u(t),\ \boldsymbol{q}_w(t_f) = \boldsymbol{q}_f \tag{B.24}$$

$$\dot{\boldsymbol{Q}}_w^l(t) = \boldsymbol{A}(t)\boldsymbol{Q}_w^l(t) + \boldsymbol{Q}_w^f(t)\boldsymbol{A}^T(t) - \gamma_w(t)\boldsymbol{Q}_w^l(t) - \gamma_w^{-1}(t)\boldsymbol{Q}_{u_b}(t),\ \boldsymbol{Q}_w(t_f) = \boldsymbol{Q}_f \tag{B.25}$$

where $\boldsymbol{q}_f \in \mathbb{R}^{n_x}$ and $\boldsymbol{Q}_f \in \mathbb{R}^{n_x \times n_x}$ are the center and shape matrix of ellipsoidal target set, and $\gamma_w(t)$ is a design variable given by

$$\gamma_w(t) = \sqrt{\frac{\langle \boldsymbol{l}(t), \boldsymbol{Q}_{u_b}(t)\boldsymbol{l}(t)\rangle}{\langle \boldsymbol{l}(t), \boldsymbol{Q}_w^l(t)\boldsymbol{l}(t)\rangle}} \tag{B.26}$$

with

$$\dot{\boldsymbol{l}}(t) = -\boldsymbol{A}^T(t)\boldsymbol{l}(t),\ \boldsymbol{l}(t_f) = \boldsymbol{l} \tag{B.27}$$

and $\boldsymbol{l} \in \mathbb{R}^{n_x}$ is arbitrary direction.

By integrating (B.24) and (B.25), the parameters of the external tight ellipsoid in each direction can be formulated as

$$\boldsymbol{q}_w(t) = \boldsymbol{\Phi}(t,t_f)\boldsymbol{q}_z(t) \tag{B.28}$$

$$\boldsymbol{Q}_w^l(t) = \boldsymbol{\Phi}(t,t_f)\boldsymbol{Q}_z^l(t)\boldsymbol{\Phi}^T(t,t_f) \tag{B.29}$$

Moreover, $\boldsymbol{q}_z(t) \in \mathbb{R}^{n_x}$ and $\boldsymbol{Q}_z^l \in \mathbb{R}^{n_x \times n_x}$, the parameters of the external tight ellipsoid in transformed space, are given by

$$\boldsymbol{q}_z(t) = \boldsymbol{q}_f - \int_t^{t_f} \overline{\boldsymbol{q}}_{u_f}(\tau)d\tau \tag{B.30}$$

$$\boldsymbol{Q}_z^l(t) = \Big(\sqrt{\langle \boldsymbol{l}, \boldsymbol{Q}_f\boldsymbol{l}\rangle} + \int_t^{t_f} \sqrt{\langle \boldsymbol{l}, \overline{\boldsymbol{Q}}_{u_f}\boldsymbol{l}\rangle}d\tau\Big)\Big(\frac{\boldsymbol{Q}_f}{\sqrt{\langle \boldsymbol{l}, \boldsymbol{Q}_f\boldsymbol{l}\rangle}} + \int_{t_0}^{t_f} \frac{\overline{\boldsymbol{Q}}_{u_f}(t)}{\sqrt{\langle \boldsymbol{l}, \overline{\boldsymbol{Q}}_{u_f}\boldsymbol{l}\rangle}}d\tau\Big) \tag{B.31}$$

Here $\overline{\boldsymbol{q}}_{u_f}(\tau) \in \mathbb{R}^{n_x}$ and $\overline{\boldsymbol{Q}}_{u_f}(\tau) \in \mathbb{R}^{n_x \times n_x}$ are computed as

$$\overline{\boldsymbol{q}}_{u_f}(\tau) = \boldsymbol{\Phi}(t_f,\tau)\boldsymbol{B}(\tau)\boldsymbol{q}_u(\tau) \tag{B.32}$$

$$\overline{\boldsymbol{Q}}_{u_f}(\tau) = \boldsymbol{\Phi}(t_f,\tau)\boldsymbol{Q}_{u_b}(\tau)\boldsymbol{\Phi}^T(t_f,\tau) \tag{B.33}$$

Remark. *By defining $L(\boldsymbol{Q}_w) = \langle \boldsymbol{l}, \boldsymbol{Q}_w\boldsymbol{l}\rangle$ in (B.17) where $\boldsymbol{l}$ is boundary condition in*

(B.27) and solving (B.19) for $\tau \in [t, t_f]$ *similar result for the shape matrix of external tight ellipsoid in (B.31) can be derived [23].*

Appendix C

Ellipsoidal Calculus

C.1 Derivation of Shape Matrix of Ellipsoidal Backward Reach Set

To find an analytical solution for the shape matrix of the ellipsoid ($\boldsymbol{Q}^i_{x_l}$) in (4.7), $h(\tau)$ and $\boldsymbol{H}(\tau) \in \mathbb{R}^{6\times 6}$ are derived by using (B.29)

$$h(\tau) = \int \sqrt{\boldsymbol{l}^T \overline{\boldsymbol{Q}}_{u_f}(\tau)\boldsymbol{l}}\, d\tau \tag{C.1}$$

$$\boldsymbol{H}(\tau) = \int \frac{\overline{\boldsymbol{Q}}_{u_f}(\tau)}{\sqrt{\boldsymbol{l}^T \overline{\boldsymbol{Q}}_{u_f}(\tau)\boldsymbol{l}}}\, d\tau \tag{C.2}$$

By substituting (3.8) and (3.9) in (B.33), $\overline{\boldsymbol{Q}}_{u_f}(\tau) \in \mathbb{R}^{6\times 6}$ is derived as

$$\overline{\boldsymbol{Q}}_{u_f}(\tau) = \boldsymbol{F}(\tau) \otimes \boldsymbol{Q}_u(\tau)$$

Here the symbol $\otimes$ represents Kronecker product and $F(\tau) \in \mathbb{R}^{2\times 2}$ is obtained as

$$\boldsymbol{F}(\tau) = \begin{bmatrix} (t_f-\tau)^2 & t_f-\tau \\ t_f-\tau & 1 \end{bmatrix}$$

By assuming that the shape matrix of the ellipsoidal input set is constant over the control time horizon ($\boldsymbol{Q}_u(\tau) = \boldsymbol{Q}_u$, $\boldsymbol{Q}_u \in \mathbb{R}^{3\times 3}$), (C.1) and (C.2) are reformulated as

$$\begin{aligned} h(\tau) &= \int f(\tau) d\tau \\ &= \int \sqrt{c_1(t_f-\tau)^2 + c_2(t_f-\tau) + c_3} d\tau \end{aligned} \tag{C.3}$$

$$\boldsymbol{H}(\tau) = \int_t^{t_f} \frac{\boldsymbol{F}(s)}{f(s)} ds \otimes \boldsymbol{Q}_u = [\boldsymbol{J}(\tau)]_t^{t_f} \otimes \boldsymbol{Q}_u \tag{C.4}$$

where

$$\begin{aligned} c_1 &= [l_1,\ l_3,\ l_5]\boldsymbol{Q}_u[l_1,\ l_3,\ l_5]^T \\ c_2 &= 2[l_1,\ l_3,\ l_5]\boldsymbol{Q}_u[l_2,\ l_4,\ l_6]^T \\ c_3 &= [l_2,\ l_4,\ l_6]\boldsymbol{Q}_u[l_2,\ l_4,\ l_6]^T \end{aligned}$$

and $[l_1,\ l_2,\ l_3,\ l_4,\ l_5,\ l_6]$ are the elements of the direction vector ($\boldsymbol{l} \in \mathbb{R}^6$) and $\boldsymbol{J}(\tau) \in \mathbb{R}^{2\times 2}$ is a symmetric time varying matrix

$$\boldsymbol{J}(\tau) = \begin{bmatrix} \boldsymbol{J}_{11}(\tau) & \boldsymbol{J}_{12}(\tau) \\ \boldsymbol{J}_{21}(\tau) & \boldsymbol{J}_{22}(\tau) \end{bmatrix}$$

Since $\boldsymbol{Q}_u$ is positive definite matrix, $c_1 \geq 0$ for all $[l_1,\ l_3,\ l_5] \in R^3$.

Given $c_1 > 0$, integral terms $h(\tau)$ and $\boldsymbol{J}(\tau)$ in (C.3)-(C.4) are calculated as

$$
\begin{aligned}
h(\tau) &= \int R_1 d\tau \\
&= \frac{2c_1(t_f-\tau)+c_2}{4c_1} R_1 + \frac{4c_1c_3 - c_2^2}{8c_1} \int \frac{d\tau}{R_1} \\
J_{11}(\tau) &= \int \frac{(t_f-\tau)^2}{R_1} d\tau \\
&= \frac{2c_1(t_f-\tau)-3c_2}{4c_1^2} R_1 + \frac{3c_2^2 - 4c_1c_3}{8c_1^2} \int \frac{d\tau}{R_1} \\
J_{12}(\tau) = J_{21}(\tau) &= \int \frac{t_f-\tau}{R_1} d\tau \\
&= (\frac{R_1}{c_1} - \frac{c_2}{2c_1} \int \frac{d\tau}{R_1}) \\
J_{22}(\tau) &= \int \frac{d\tau}{R_1} \\
&= \frac{1}{\sqrt{c_1}} \ln|2\sqrt{c_1}R_1 + 2c_1(t_f-\tau)+c_2|,\ 4c_1c_3 - c_2^2 \neq 0 \\
&= \frac{1}{\sqrt{c_1}} \ln|2c_1(t_f-\tau)+c_2|, \quad 4c_1c_3 - c_2^2 = 0
\end{aligned}
$$

where

$$
R_1 = \sqrt{c_1(t_f-\tau)^2 + c_2(t_f-\tau) + c_3}
$$

If $c_1 = 0$ and $c_2 \neq 0$

$$\begin{aligned}
h(\tau) &= \int R_2 d\tau = \frac{2R_2^3}{3c_2} \\
J_{11}(\tau) &= \int \frac{(t_f-\tau)^2}{R_2} d\tau \\
&= \frac{2}{5c_2}((t_f-\tau)^2 R_2 - 2c_3 \int \frac{t_f-\tau}{R_2} d\tau) \\
J_{12}(\tau) = J_{21}(\tau) &= \int \frac{t_f-\tau}{R_2} d\tau \\
&= \frac{2}{3c_2}((t_f-\tau)R_2 - c_3 \frac{2R_2}{c_2}) \\
J_{22}(\tau) &= \int \frac{d\tau}{R_2} = \frac{2R_2}{c_2}
\end{aligned}$$

where

$$R_2 = \sqrt{c_2(t_f-\tau)+c_3}$$

If $c_1 = 0$ and $c_2 = 0$

$$\begin{aligned}
h(\tau) &= \int \sqrt{c_3} d\tau = \sqrt{c_3}(t_f-\tau) \\
J_{11}(\tau) &= \int \frac{(t_f-\tau)^2}{\sqrt{c_3}} = \frac{(t_f-\tau)^3}{3\sqrt{c_3}} \\
J_{12}(\tau) = J_{21}(\tau) &= \int \frac{t_f-\tau}{\sqrt{c_3}} = \frac{(t_f-\tau)^2}{2\sqrt{c_3}} \\
J_{22}(\tau) &= \int \frac{1}{\sqrt{c_3}} d\tau = \frac{1}{\sqrt{c_3}}(t_f-\tau)
\end{aligned}$$

C.2 Derivation of Undesirable Ellipsoidal Set

Using the parameters of the i-th undesirable ellipsoidal set in (4.12) and (4.12), the equation of the ellipsoid can be written as

$$(\boldsymbol{p}-\boldsymbol{q}^i_{p_f})^T{\boldsymbol{Q}^i_{p_f}}^{-1}(\boldsymbol{p}-\boldsymbol{q}^i_{p_f}) = \frac{1-\epsilon(\boldsymbol{v}-\boldsymbol{q}_{v_f})^T{\boldsymbol{Q}^i_{v_f}}^{-1}(\boldsymbol{v}-\boldsymbol{q}_{v_f})}{1-\epsilon} \tag{C.5}$$

Moreover, $(\boldsymbol{v}-\boldsymbol{q}_{v_f})^T{\boldsymbol{Q}^i_{v_f}}^{-1}(\boldsymbol{v}-\boldsymbol{q}_{v_f}) \leq 1$ for all reachable velocities, where $\boldsymbol{q}_{v_f}$ and $\boldsymbol{Q}_{v_f}$ denote the parameters of minimum volume ellipsoid covering all feasible velocities in (5.16c). It can be conclude that

$$\alpha = \frac{1-\epsilon(\boldsymbol{v}-\boldsymbol{q}_{v_f})^T{\boldsymbol{Q}^i_{v_f}}^{-1}(\boldsymbol{v}-\boldsymbol{q}_{v_f})}{1-\epsilon} \geq 1, \quad 0<\epsilon<1$$

Dividing (C.5) by $\alpha \geq 1$ yields

$$\boldsymbol{p}-\boldsymbol{q}_{p_f})^T(\alpha\boldsymbol{Q}^i_{p_f})^{-1}(\boldsymbol{p}-\boldsymbol{q}_{p_f}) = 1 \tag{C.6}$$

Consequently, $\alpha\boldsymbol{Q}^i_{p_f} \geq \boldsymbol{Q}^i_{p_f}$. By choosing $\boldsymbol{q}^i_{p_f}$ and $\boldsymbol{Q}^i_{p_f}$ as the parameters of minimum volume ellipsoid covering the i-th enlarged obstacle and by considering $\alpha\boldsymbol{Q}^i_{p_f} \geq \boldsymbol{Q}^i_{p_f}$ it is concluded that $E(\boldsymbol{q}_{p_f}, \boldsymbol{Q}^i_{p_f}) \subseteq E(\boldsymbol{q}^i_{x_f}, \boldsymbol{Q}^i_{x_f})$ for any reachable velocity, i.e., undesirable set covers the enlarged obstacle for any reachable velocity.

C.3 Ellipsoid Projection

The boundary of an ellipsoid is defined by $E_e(\boldsymbol{q}, \boldsymbol{Q})$

$$E(\boldsymbol{q}, \boldsymbol{Q}) = \{\boldsymbol{x} | (\boldsymbol{x} - \boldsymbol{q})^T \boldsymbol{Q}^{-1} (\boldsymbol{x} - \boldsymbol{q}) = 1\} \tag{C.7}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ and $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ are parameters of the ellipsoid.

In order to calculate the projection of n dimensional ellipsoid into m dimensional space ($m < n$), $\boldsymbol{Q}^{-1}$ can be written as a block matrix

$$\boldsymbol{Q}^{-1} = \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2^T \\ \boldsymbol{U}_2 & \boldsymbol{U}_3 \end{bmatrix}$$

where $\boldsymbol{U}_1 \in \mathbb{R}^{m \times m}$, $\boldsymbol{U}_2 \in \mathbb{R}^{(n-m) \times m}$ and $\boldsymbol{U}_3 \in \mathbb{R}^{(n-m) \times (n-m)}$. Considering $\boldsymbol{Q}^{-1}$ is positive definite matrix, $\boldsymbol{U}_1$ and $\boldsymbol{U}_3$ are also positive definite and invertible matrices. Projection of n-dimensional ellipsoid into $\boldsymbol{y} = [x_1, x_2, ..., x_m]^T$ space is a set of points where $\nabla(F)$ has no component in $\boldsymbol{z} = [x_{m+1}, x_{m+2}, ..., x_n]^T$ direction.

$$\boldsymbol{U}_2(\boldsymbol{y} - \boldsymbol{q}_{y_c}) + \boldsymbol{U}_3(\boldsymbol{z} - \boldsymbol{q}_{z_c}) = 0$$

By substituting the above equation into ellipsoid equation (C.7), the following equation is obtained

$$(\boldsymbol{y} - \boldsymbol{q}_{y_c})^T (\boldsymbol{U}_1 - \boldsymbol{U}_2^T \boldsymbol{U}_3^{-1} \boldsymbol{U}_2)(\boldsymbol{y} - \boldsymbol{q}_{y_c}) = 1 \tag{C.8}$$

Because $\boldsymbol{Q}^{-1}$ is positive define matrix, the Shur complement $(\boldsymbol{U}_1 - \boldsymbol{U}_2^T\boldsymbol{U}_3^{-1}\boldsymbol{U}_2 = \boldsymbol{Q}^{-1}/\boldsymbol{U}_3)$ is positive definite matrix. Therefore, the projection is an ellipsoid calculated by C.8.

Bibliography

[1] Albaker, B. and Rahim, N. (2009). A survey of collision avoidance approaches for unmanned aerial vehicles. In *2009 international conference for technical postgraduates (TECHPOS)*, pages 1–7. IEEE.

[2] Althoff, M. and Krogh, B. H. (2013). Reachability analysis of nonlinear differential-algebraic systems. *IEEE Transactions on Automatic Control*, **59**(2), 371–383.

[3] Althoff, M., Stursberg, O., and Buss, M. (2010). Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear analysis: hybrid systems*, **4**(2), 233–249.

[4] Althoff, M., Frehse, G., and Girard, A. (2020). Set propagation techniques for reachability analysis. *Annual Review of Control, Robotics, and Autonomous Systems*, **4**.

[5] ApS, M. (2019). *MOSEK Optimizer API for Python 9.0.105.*

[6] Asarin, E., Bournez, O., Dang, T., and Maler, O. (2000). Approximate reachability analysis of piecewise-linear dynamical systems. In *International workshop on hybrid systems: Computation and control*, pages 20–31. Springer.

[7] Asarin, E., Dang, T., and Girard, A. (2007). Hybridization methods for the analysis of nonlinear systems. *Acta Informatica*, **43**(7), 451–476.

[8] Augugliaro, F., Schoellig, A. P., and D'Andrea, R. (2012). Generation of collision-free trajectories for a quadrocopter fleet: A sequential programming approach. In *2012 IEEE/RSJ international conference on Intelligent Robots and Systems*, pages 1917–1922. IEEE.

[9] Banjac, G., Stellato, B., Moehle, N., Goulart, P., Bemporad, A., and Boyd, S. (2017a). Embedded code generation using the OSQP solver. In *IEEE Conference on Decision and Control (CDC)*.

[10] Banjac, G., Stellato, B., Moehle, N., Goulart, P., Bemporad, A., and Boyd, S. (2017b). Embedded code generation using the osqp solver. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1906–1911. IEEE.

[11] Bareiss, D., van den Berg, J., and Leang, K. K. (2015). Stochastic automatic collision avoidance for tele-operated unmanned aerial vehicles. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4818–4825. IEEE.

[12] Bemporad, A., Borrelli, F., and Morari, M. (2003). Min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on automatic control*, **48**(9), 1600–1606.

[13] Ben-Tal, A. and Nemirovski, A. (2001). *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM.

[14] Biswas, S., Anavatti, S. G., and Garratt, M. A. (2019). A particle swarm optimization based path planning method for autonomous systems in unknown terrain. In *2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pages 57–63. IEEE.

[15] Blochliger, F., Fehr, M., Dymczyk, M., Schneider, T., and Siegwart, R. (2018). Topomap: Topological mapping and navigation based on visual slam maps. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3818–3825. IEEE.

[16] Boivin, E., Desbiens, A., and Gagnon, E. (2008). Uav collision avoidance using cooperative predictive control. In *2008 16th Mediterranean Conference on Control and Automation*, pages 682–688. IEEE.

[17] Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[18] Brescianini, D. and D'Andrea, R. (2016). Design, modeling and control of an omni-directional aerial vehicle. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3261–3266. IEEE.

[19] Chai, R., Tsourdos, A., Gao, H., Xia, Y., and Chai, S. (2021). Dual-loop tube-based robust model predictive attitude tracking control for spacecraft with system constraints and additive disturbances. *IEEE Transactions on Industrial Electronics*.

[20] Chen, J., Liu, T., and Shen, S. (2016). Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments. In *2016 IEEE*

International Conference on Robotics and Automation (ICRA), pages 1476–1483. IEEE.

[21] Chen, M. and Tomlin, C. J. (2018). Hamilton–jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management. *Annual Review of Control, Robotics, and Autonomous Systems*, **1**, 333–358.

[22] Chen, X., Abraham, E., and Sankaranarayanan, S. (2012). Taylor model flowpipe construction for non-linear hybrid systems. In *2012 IEEE 33rd Real-Time Systems Symposium*, pages 183–192. IEEE.

[23] Chernousko, F. (2005). Ellipsoidal state estimation for dynamical systems. *Nonlinear Analysis: Theory, Methods & Applications*, **63**(5-7), 872–879.

[24] Chernousko, F. and Ovseevich, A. (2004). Properties of the optimal ellipsoids approximating the reachable sets of uncertain systems. *Journal of optimization theory and applications*, **120**(2), 223–246.

[25] Choi, D., Lee, K., and Kim, D. (2020). Enhanced potential field-based collision avoidance for unmanned aerial vehicles in a dynamic environment. In *AIAA Scitech 2020 Forum*, page 0487.

[26] Chutinan, A. and Krogh, B. H. (2003). Computational techniques for hybrid system verification. *IEEE transactions on automatic control*, **48**(1), 64–75.

[27] Deits, R. and Tedrake, R. (2015a). Computing large convex regions of obstacle-free space through semidefinite programming. In *Algorithmic foundations of robotics XI*, pages 109–124. Springer.

[28] Deits, R. and Tedrake, R. (2015b). Efficient mixed-integer planning for uavs in cluttered environments. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 42–49. IEEE.

[29] Derafa, L., Madani, T., and Benallegue, A. (2006). Dynamic modelling and experimental identification of four rotors helicopter parameters. In *2006 IEEE international conference on industrial technology*, pages 1834–1839. IEEE.

[30] Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, **58**(15-16), 1–35.

[31] Diehl, M. and Bjornberg, J. (2004). Robust dynamic programming for min-max model predictive control of constrained uncertain systems. *IEEE Transactions on Automatic Control*, **49**(12), 2253–2257.

[32] Drop, F. M., Pool, D. M., Damveld, H. J., van Paassen, M. M., and Mulder, M. (2013). Identification of the feedforward component in manual control with predictable target signals. *IEEE transactions on cybernetics*, **43**(6), 1936–1949.

[33] Drop, F. M., Pool, D. M., van Paassen, M. R. M., Mulder, M., and Bülthoff, H. H. (2016). Objective model selection for identifying the human feedforward response in manual control. *IEEE transactions on cybernetics*, **48**(1), 2–15.

[34] Drop, F. M., Pool, D. M., Van Paassen, M. M., Mulder, M., and Bülthoff, H. H. (2018). Effects of target signal shape and system dynamics on feedforward in manual control. *IEEE transactions on cybernetics*, **49**(3), 768–780.

[35] Dyer, E. (2018). *Modelling and Control of an Omni-directional UAV*. Master's thesis, McMaster University.

[36] Dyer, E., Sirouspour, S., and Jafarinasab, M. (2019). Energy optimal control allocation in a redundantly actuated omnidirectional uav. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5316–5322. IEEE.

[37] Eraslan, E., Yildiz, Y., and Annaswamy, A. M. (2020). Shared control between pilots and autopilots: An illustration of a cyberphysical human system. *IEEE Control Systems Magazine*, **40**(6), 77–97.

[38] Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., and Maler, O. (2011). Spaceex: Scalable verification of hybrid systems. In *International Conference on Computer Aided Verification*, pages 379–395. Springer.

[39] Gao, F., Wang, L., Zhou, B., Zhou, X., Pan, J., and Shen, S. (2020). Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments. *IEEE Transactions on Robotics*, **36**(5), 1526–1545.

[40] Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. In *International Workshop on Hybrid Systems: Computation and Control*, pages 291–305. Springer.

[41] Gunn, S. R. *et al.* (1998). Support vector machines for classification and regression. *ISIS technical report*, **14**(1), 5–16.

[42] Ha, L. N. N. T., Bui, D. H. P., Hong, S. K., *et al.* (2019). Nonlinear control for autonomous trajectory tracking while considering collision avoidance of uavs based on geometric relations. *Energies*, **12**(8), 1551.

[43] Halder, A. (2018). On the parameterized computation of minimum volume outer ellipsoid of minkowski sum of ellipsoids. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4040–4045. IEEE.

[44] Hang, P., Xia, X., Chen, G., and Chen, X. (2021). Active safety control of automated electric vehicles at driving limits: A tube-based mpc approach. *IEEE Transactions on Transportation Electrification*.

[45] Hao, Y., Davari, A., and Manesh, A. (2005). Differential flatness-based trajectory planning for multiple unmanned aerial vehicles using mixed-integer linear programming. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 104–109. IEEE.

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

[47] Hou, X. and Mahony, R. (2015). Dynamic kinesthetic boundary for haptic teleoperation of vtol aerial robots in complex environments. *IEEE Transactions on systems, man, and cybernetics: Systems*, **46**(5), 694–705.

[48] Hoy, M., Matveev, A. S., and Savkin, A. V. (2015). Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica*, **33**(3), 463–497.

[49] Krendel, E. S. and McRuer, D. T. (1960). A servomechanisms approach to skill development. *Journal of the Franklin Institute*, **269**(1), 24–42.

[50] Kurdi, H., Al-Megren, S., Aloboud, E., Alnuaim, A. A., Alomair, H., Alothman, R., Muhayya, A. B., Alharbi, N., Alenzi, M., and Youcef-Toumi, K. (2020). Bee-inspired task allocation algorithm for multi-uav search and rescue missions. *International Journal of Bio-Inspired Computation*, **16**(4), 252–263.

[51] Kurzhanski, A. B. (2014). Dynamics and control of trajectory tubes. theory and computation. In *2014 20th International Workshop on Beam Dynamics and Optimization (BDO)*, pages 1–1. IEEE.

[52] Kurzhanski, A. B. and Varaiya, P. (2002a). On ellipsoidal techniques for reachability analysis. part i: external approximations. *Optimization methods and software*, **17**(2), 177–206.

[53] Kurzhanski, A. B. and Varaiya, P. (2002b). On ellipsoidal techniques for reachability analysis. part ii: Internal approximations box-valued constraints. *Optimization methods and software*, **17**(2), 207–237.

[54] Langson, W., Chryssochoos, I., Raković, S., and Mayne, D. Q. (2004). Robust model predictive control using tubes. *Automatica*, **40**(1), 125–133.

[55] Larsson, T. (2008). An efficient ellipsoid-obb intersection test. *journal of graphics tools*, **13**(1), 31–43.

[56] Le Guernic, C. and Girard, A. (2010). Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, **4**(2), 250–262.

[57] Lin, Z., Castano, L., Mortimer, E., and Xu, H. (2020). Fast 3d collision avoidance algorithm for fixed wing uas. *Journal of Intelligent & Robotic Systems*, **97**(3), 577–604.

[58] Liu, S., Watterson, M., Mohta, K., Sun, K., Bhattacharya, S., Taylor, C. J., and Kumar, V. (2017). Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments. *IEEE Robotics and Automation Letters*, **2**(3), 1688–1695.

[59] Mahony, R., Kumar, V., and Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics and Automation magazine*, **19**(3), 20–32.

[60] Mammarella, M. and Capello, E. (2020). Tube-based robust mpc processor-in-the-loop validation for fixed-wing uavs. *Journal of Intelligent & Robotic Systems*, **100**(1), 239–258.

[61] Margalit, D. and Rabinoff, J. (2018). Interactive linear algebra. *Georgia Institute of Technology*.

[62] Margellos, K. and Lygeros, J. (2011). Hamilton–jacobi formulation for reach–avoid differential games. *IEEE Transactions on Automatic Control*, **56**(8), 1849–1861.

[63] Mayne, D. Q., Raković, S. V., Findeisen, R., and Allgöwer, F. (2006). Robust output feedback model predictive control of constrained linear systems. *Automatica*, **42**(7), 1217–1222.

[64] Mayne, D. Q., Raković, S., Findeisen, R., and Allgöwer, F. (2009). Robust output feedback model predictive control of constrained linear systems: Time varying case. *Automatica*, **45**(9), 2082–2087.

[65] McRuer, D. T. and Krendel, E. S. (1959). The human operator as a servo system element. *Journal of the Franklin Institute*, **267**(5), 381–403.

[66] McRuer, D. T. and Krendel, E. S. (1974). Mathematical models of human pilot behavior. Technical report, ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT NEUILLY-SUR-SEINE (FRANCE).

[67] Mellinger, D., Kushleyev, A., and Kumar, V. (2012). Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In *2012 IEEE international conference on robotics and automation*, pages 477–483. IEEE.

[68] Menouar, H., Guvenc, I., Akkaya, K., Uluagac, A. S., Kadri, A., and Tuncer, A. (2017). Uav-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, **55**(3), 22–28.

[69] Mitchell, I. M., Bayen, A. M., and Tomlin, C. J. (2005). A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, **50**(7), 947–957.

[70] Mulder, M., Pool, D. M., Abbink, D. A., Boer, E. R., Zaal, P. M., Drop, F. M., van der El, K., and van Paassen, M. M. (2017). Manual control cybernetics: State-of-the-art and current trends. *IEEE Transactions on Human-Machine Systems*, **48**(5), 468–485.

[71] Ovseevich, A. (1997). On equations of ellipsoids approximating attainable sets. *Journal of optimization theory and applications*, **95**(3), 659–676.

[72] Ovseevich, A. (2010). Explicit formulas for ellipsoids approximating attainable sets: Ii. *Journal of Computer and Systems Sciences International*, **49**(3), 353–359.

[73] Ovseevich, A. and Taraban'ko, Y. V. (2007). Explicit formulas for the ellipsoids approximating attainability domains. *Journal of Computer and Systems Sciences International*, **46**(2), 194–205.

[74] Pham, H. X., La, H. M., Feil-Seifer, D., and Deans, M. C. (2018). A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **50**(4), 1537–1548.

[75] Radmanesh, M., Kumar, M., Guentert, P. H., and Sarim, M. (2018). Overview of path-planning and obstacle avoidance algorithms for uavs: a comparative study. *Unmanned systems*, **6**(02), 95–118.

[76] Rakovic, S. V. (2007). Minkowski algebra and banach contraction principle in set invariance for linear discrete time systems. In *2007 46th IEEE Conference on Decision and Control*, pages 2169–2174. IEEE.

[77] Ramalho, G. M., Carvalho, S. R., Finardi, E. C., and Moreno, U. F. (2018). Trajectory optimization using sequential convex programming with collision avoidance. *Journal of Control, Automation and Electrical Systems*, **29**(3), 318–327.

[78] Ramdani, N., Meslem, N., and Candau, Y. (2008). Reachability of uncertain nonlinear systems using a nonlinear hybridization. In *International Workshop on Hybrid Systems: Computation and Control*, pages 415–428. Springer.

[79] Recchiuto, C. T., Sgorbissa, A., and Zaccaria, R. (2016). Visual feedback with multiple cameras in a uavs human–swarm interface. *Robotics and Autonomous Systems*, **80**, 43–54.

[80] Saltık, M. B., Özkan, L., Ludlage, J. H., Weiland, S., and Van den Hof, P. M. (2018). An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects. *Journal of Process Control*, **61**, 77–102.

[81] Schouwenaars, T., De Moor, B., Feron, E., and How, J. (2001). Mixed integer programming for multi-vehicle path planning. In *2001 European control conference (ECC)*, pages 2603–2608. IEEE.

[82] Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., and Abbeel, P. (2014). Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, **33**(9), 1251–1270.

[83] Scokaert, P. O. and Mayne, D. Q. (1998). Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic control*, **43**(8), 1136–1142.

[84] Scott, J. K. and Barton, P. I. (2013). Bounds on the reachable sets of nonlinear control systems. *Automatica*, **49**(1), 93–100.

[85] Seo, J., Kim, Y., Kim, S., and Tsourdos, A. (2017). Collision avoidance strategies for unmanned aerial vehicles in formation flight. *IEEE Transactions on aerospace and electronic systems*, **53**(6), 2718–2734.

[86] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2018). Osqp: An operator splitting solver for quadratic programs. In *2018 UKACC 12th international conference on control (CONTROL)*, pages 339–339. IEEE.

[87] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2020). OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, **12**(4), 637–672.

[88] Sun, J., Tang, J., and Lao, S. (2017). Collision avoidance for cooperative uavs with optimized artificial potential field algorithm. *IEEE Access*, **5**, 18382–18390.

[89] Tang, L., Dian, S., Gu, G., Zhou, K., Wang, S., and Feng, X. (2010). A novel potential field method for obstacle avoidance and path planning of mobile robot. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 9, pages 633–637. IEEE.

[90] Tohidi, S. S. and Yildiz, Y. (2019). Adaptive human pilot model for uncertain systems. In *2019 18th European Control Conference (ECC)*, pages 2938–2943. IEEE.

[91] van der El, K., Pool, D. M., Damveld, H. J., van Paassen, M. R. M., and Mulder, M. (2015). An empirical human controller model for preview tracking tasks. *IEEE transactions on cybernetics*, **46**(11), 2609–2621.

[92] van der El, K., Pool, D. M., van Paassen, M. R. M., and Mulder, M. (2017). Effects of preview on human control behavior in tracking tasks with various controlled elements. *IEEE transactions on cybernetics*, **48**(4), 1242–1252.

[93] van der El, K., Padmos, S., Pool, D. M., van Paassen, M. M., and Mulder, M. (2018). Effects of preview time in manual tracking tasks. *IEEE Transactions on Human-Machine Systems*, **48**(5), 486–495.

[94] Xu, B. and Sreenath, K. (2018). Safe teleoperation of dynamic uavs through control barrier functions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7848–7855. IEEE.

[95] Xu, S., Tan, W., Efremov, A. V., Sun, L., and Qu, X. (2017). Review of control models for human pilot behavior. *Annual Reviews in Control*, **44**, 274–291.

[96] Yacef, F., Rizoug, N., Bouhali, O., and Hamerlain, M. (2017). Optimization of energy consumption for quadrotor uav. In *Proceedings of the International Micro Air Vehicle Conference and Flight Competition (IMAV), Toulouse, France*, pages 18–21.

[97] Yasin, J. N., Mohamed, S. A., Haghbayan, M.-H., Heikkonen, J., Tenhunen, H., and Plosila, J. (2020). Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches. *IEEE Access*, **8**, 105139–105155.

[98] Zheng, Y.-J., Du, Y.-C., Ling, H.-F., Sheng, W.-G., and Chen, S.-Y. (2019). Evolutionary collaborative human-uav search for escaped criminals. *IEEE Transactions on Evolutionary Computation*, **24**(2), 217–231.

[99] Zhong, X., Wu, Y., Wang, D., Wang, Q., Xu, C., and Gao, F. (2020). Generating large convex polytopes directly on point clouds. *arXiv preprint arXiv:2010.08744*.

[100] Zhou, Y., Hu, H., Liu, Y., Lin, S.-W., and Ding, Z. (2017). A real-time and fully distributed approach to motion planning for multirobot systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **49**(12), 2636–2650.