2021 European Control Conference (ECC)
June 29 - July 2, 2021. Rotterdam, Netherlands

Incorporating Prediction in Control Barrier Function Based Distributive Multi-Robot Collision Avoidance

Pravin Mali[1], K. Harikumar [1], Arun Kumar Singh[2], K. Madhava Krishna[1] and P.B. Sujit[3]

Abstract— **Control barrier function (CBF) constraints provide a rigorous characterization of the space of control inputs that ensure the satisfaction of state constraints, such as collision avoidance, at all time instants. However, CBFs are highly non-linear and non-convex and thus, when incorporated within an optimization-based algorithm such as Model Predictive Control (MPC), leads to a computationally challenging problem. Existing works by-pass the computational intractability by collapsing the horizon of the MPC to a single step, although this comes at the cost of severe degradation of performance.**

In this paper, we present two contributions to ensure the real-time performance of CBFs based MPC over long horizons in the context of multi-robot collision avoidance. First, we propose a customized Project Gradient Descent Method that incurs minimal computational overhead over existing one-step approaches but leads to a substantial improvement in trajectory smoothness, time to reach the goal, etc. Our second contribution lies in applying the proposed MPC to both quadrotors and fixed-wing aerial aircrafts (FWA). In particular, we show that the formulation for the quadrotors can be readily extended to the latter by deriving additional CBFs for the curvature and forward velocity constraints. We validate our algorithm with an extensive simulation of up to 10 robots in challenging benchmark scenarios.

I. Introduction

Collision avoidance among robots is essential for the safe operation of the robots performing different collaborative missions. In a typical scenario, the execution of the mission would be the primary objective, and the collision avoidance strategy would be invoked when the robots are in the collision zone. As the robot density increases, the probability of collision will increase and hence the collision avoidance strategy will be invoked more frequently. The popular methods of multi-robot collision avoidance are the velocity obstacles (VO) [1] and the related algorithms like the reciprocal velocity obstacles (RVO) [2] and the optimal reciprocal collision avoidance (ORCA) [3]. A repulsive strategy based collision avoidance is employed for multi-robot consensus control in [4]. When the robots approach the limiting safe distance, the repulsive strategy dominates over the consensus control. MPC based iterative techniques are employed for multi-robot collision avoidance in [5]-[6]. MPC based algorithms are capable of handling bounded unmeasured disturbance inputs acting on the system dynamics [7]. Recently, reinforcement learning based methods are applied for multi-robot collision avoidance in complex scenarios containing stationary and moving obstacles [8]. The learning-based methods require a large amount of training data and are not easily made available for robots like unmanned aerial vehicles (UAVs) performing complex missions.

The safety barrier certificates (SBC) provides theoretical guarantees for collision avoidance in the absence of the external disturbances or uncertainties in the system dynamics [9]. The main idea behind the SBC is to find a suitable CBF which ensures that the robot starting from a collision-free zone remains inside it [10]. The use of CBFs to design safety-critical control systems is discussed in [11]. Lyapunov like inequality conditions can be imposed on the derivative of the CBF to ensure the forward invariance of the safety set. The CBFs are of two types viz. zeroing CBF and reciprocal CBF [12]. Zeroing CBF vanishes on the boundary of the safety set, whereas the reciprocal CBF tends to infinity on the boundary. A zeroing CBF based on the relative distance between the robots, their velocity and maximum breaking acceleration is defined in [9]. A nominal controller is designed for each robot that tracks the reference trajectory in the absence of other robots. Quadratic programming (QP) based optimization problem is solved for each robot at each time instant to generate control action close to the nominal controller satisfying the inequality constraints corresponding to the forward invariance of the safety set as modelled by CBF. Although CBF based approaches like [9] can robustly handle collision avoidance, the resulting trajectories are highly sub-optimal. As shown in Section IV, the robots often come to a near-halt to avoid imminent collisions. Not only this behaviour is undesirable, for certain systems like FWA, reducing the forward velocity beyond a certain threshold is infeasible. Furthermore, reaching near-zero velocity leads to a non-smooth control trajectory, and a considerable increase in the time taken to reach the goal when compared to the obstacle-free case. Our key conjecture is that the sub-optimal behaviour with current CBF based approaches such as [9] is that they do not reason about the over-time evolution of collision avoidance CBF constraints for the chosen control inputs. In fact, the current QP based approaches can be considered as MPC with a unit horizon. Other researchers have also made similar observations but in the context of stabilization problems of under-actuated systems [13].

In this paper, we improve the method presented in [9] and propose a MPC for multi-robot collision avoidance that hinges on formulating collision avoidance CBF constraints over longer horizons. However, the underlying optimization becomes highly non-linear and non-convex which is difficult to solve in real-time. Our main algorithmic contribution is

1. Robotics Research Center, IIIT-Hyderabad, India. 2. Institute of Technology, University of Tartu. 3. IISER Bhopal, Bhopal, India. Emails: pravin.mali@research.iiit.ac.in, harikumar.k@iiit.ac.in, arun.singh@ut.ee, mkrishna@iiit.ac.in, sujit@iiserb.ac.in. The research was partly supported by grant COVSG24 and PSG605 from Estonian Research Council.

978-94-6384-236-5 ©2021 EUCA

the development and validation of a customized projected gradient descent approach to solve the optimization problem. We use the RMSProp variant of gradient descent [14] over a non-smooth cost augmented with violations of the collision avoidance CBF constraints. We develop a novel projection step that not only considers proximity to the control input computed from gradient descent but also ensures smoothness in the projected control input profile. We also show the extension of the CBF based approach in [9] for the FWA by deriving additional barrier constraints for curvature bounds and forward velocities. We present numerical simulations that highlights the advantages of the proposed approach in terms of time-saving, reduced control effort and better control smoothness when compared to the conventional single-step collision avoidance method using the CBF.

The rest of the paper is organized as follows. In section II, we provide the second-order dynamics of a quadrotor and FWA followed by the CBF based multi-robot collision avoidance method. The details of the proposed MPC based multi-robot collision avoidance using CBF is given in section III. The numerical simulation results are provided in section IV, followed by conclusions in section V.

II. Preliminaries

A. Dynamics of a quadrotor and fixed-wing aircraft

We model holonomic robots such as quadrotors with homogeneous double-integrator dynamics. Denoting the position and velocity coordinate of the i^{th} robot as $\mathbf{p}_i(t) = [x_i(t),\, y_i(t)]^T$ and $\mathbf{v}_i(t) = [v_{xi}(t),\, v_{yi}(t)]^T$ respectively, the motion model can be represented in the following form.

$$\dot{\mathbf{p}}_i(t) = \mathbf{v}_i(t) \tag{1}$$

$$\dot{\mathbf{v}}_i(t) = \mathbf{u}_i(t) \tag{2}$$

where $\mathbf{u}_i = [u_{xi}(t),\, u_{yi}(t)]^T$ is the bounded acceleration control input. The dynamics in (1) and (2) is written in the following state space representation.

$$\dot{\mathbf{x}}_i(t) = \mathbf{A}\mathbf{x}_i(t) + \mathbf{B}\mathbf{u}_i(t) \tag{3}$$

where $\mathbf{x}_i(t) = [x_i(t),\, v_{xi}(t),\, y_i(t),\, v_{yi}(t)]^T$ and

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix},\, \mathbf{B} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{4}$$

For FWA, the dynamics given in (3) and (4) is applicable with additional constraints on the curvature of the flight trajectory and the lower bound on the flight velocity. The constraint on the velocity is given below.

$$v_{min} \leq \parallel \mathbf{v}_i(t) \parallel_2 \leq v_{max} \tag{5}$$

where the upper limit v_{max} is applicable to both quadrotor and FWA. The lower limit $v_{min} = 0$ for quadrotor, whereas for FWA $v_{min} > 0$ determined by the stall speed of the FWA. The curvature constraint is given by the below equation.

$$\frac{|\dot{\psi}(t)|}{\parallel \mathbf{v}_i(t) \parallel_2} \leq \kappa_{max} \tag{6}$$

where $\psi(t) = tan^{-1}(\frac{v_{yi}(t)}{v_{xi}(t)})$ and κ_{max} is the maximum curvature of the FWA's flight trajectory, often restricted by the wing stall conditions.

B. Barrier Certificates and Control Barrier Function

Definition I: For the dynamical system given by (3), let $\mathbf{x}_i(t) \in \mathbb{R}^4$ denote the state at time t and $\mathcal{S}_\mathbf{x} \in \mathbb{R}^4$ denote the region in the state space for which the system is in a safe operating condition. A function $\mathcal{B}(\mathbf{x}_i(t)) \in C^1(\mathbf{x}_i(t))$ is called a barrier certificate or barrier function (BF) for the system given by (3) if the following condition holds [15].

$$\textit{Condition I}: \begin{cases} \inf\limits_{\mathbf{x}_i(t) \in Int(\mathcal{S}_\mathbf{x})} \mathcal{B}(\mathbf{x}_i(t)) \geq 0 \\ \lim\limits_{\mathbf{x}_i(t) \to \partial\mathcal{S}_\mathbf{x}} \mathcal{B}(\mathbf{x}_i(t)) = \infty \\ \dot{\mathcal{B}}(\mathbf{x}_i(t)) \leq 0,\, \forall t > 0, \end{cases} \tag{7}$$

where $Int(\mathcal{S}_\mathbf{x})$ and $\partial\mathcal{S}_\mathbf{x}$ denotes the interior and boundary of $\mathcal{S}_\mathbf{x}$ respectively. The inequality $\dot{\mathcal{B}}(\mathbf{x}_i) \leq 0$ in *Condition I* ensures that a trajectory starting within a safe set remains in the safe set $\forall t > 0$. In order to incorporate a more flexible control, the inequality $\dot{\mathcal{B}}(\mathbf{x}_i(t)) \leq 0$ can be relaxed to

$$\dot{\mathcal{B}}(\mathbf{x}_i(t)) \leq \frac{\alpha}{\mathcal{B}(\mathbf{x}_i(t))}, \tag{8}$$

where $\alpha > 0$. Let $h(\mathbf{x}_i) \in C^1(\mathbf{x}_i(t))$ be a function defined on $\mathbb{R}^4 \to \mathbb{R}$ that satisfies the following condition.

$$\textit{Condition II}: \begin{cases} h(\mathbf{x}_i(t)) > 0,\, \forall \mathbf{x}_i(t) \in Int(\mathcal{S}_\mathbf{x}) \\ h(\mathbf{x}_i(t)) = 0,\, \forall \mathbf{x}_i(t) \in \partial\mathcal{S}_\mathbf{x} \end{cases} \tag{9}$$

Definition II: A function $\mathcal{B}(\mathbf{x}_i(t)) \in C^1(\mathbf{x}_i(t))$ is called a CBF for the system given by (3) if the following condition holds [10].

$$\textit{Condition III}: \begin{cases} \frac{1}{\kappa_1(h(\mathbf{x}_i(t)))} \leq \mathcal{B}(\mathbf{x}_i(t)) \leq \frac{1}{\kappa_2(h(\mathbf{x}_i(t)))} \\ \inf\limits_{\mathbf{u}_i(t) \in \mathcal{U}_i} [\frac{\partial\mathcal{B}(\mathbf{x}_i)}{\partial\mathbf{x}_i}(\mathbf{A}\mathbf{x}_i(t) + \mathbf{B}\mathbf{u}_i(t)) - \frac{\alpha}{\mathcal{B}(\mathbf{x}_i(t))}] \leq 0, \end{cases} \tag{10}$$

where $\kappa_1(.)$ and $\kappa_2(.)$ are locally Lipschitz, class $\mathcal{K}$ functions and $\mathcal{U}_i$ is the set of feasible control inputs. By choosing

$$\mathcal{B}(\mathbf{x}_i(t)) = h^{-z}(\mathbf{x}_i(t)) \tag{11}$$

where $\{z \in \mathbb{N}\}$. The second inequality of *Condition III* can be written as given below.

$$\underset{\mathbf{u}_i \in \mathcal{U}_i}{\text{Sup}} [\frac{\partial h(\mathbf{x}_i(t))}{\partial \mathbf{x}_i(t)}(\mathbf{A}\mathbf{x}_i(t) + \mathbf{B}\mathbf{u}_i(t)) + \frac{\alpha}{z} h(\mathbf{x}_i(t))^{(2z+1)}] \geq 0 \tag{12}$$

The next section obtains a suitable $h(\mathbf{x}_i(t))$ satisfying the inequality given in (12) for the multi-robot collision avoidance problem as described in [9].

C. SBC based Reactive collision Avoidance

Lets consider the relative dynamics of i^{th} and j^{th} robots given by

$$\Delta\dot{\mathbf{x}}_{ij}(t) = \mathbf{A}\Delta\mathbf{x}_{ij}(t) + \mathbf{B}\Delta\mathbf{u}_{ij}(t) \tag{13}$$

where $\Delta\mathbf{x}_{ij}(t) = \mathbf{x}_i(t) - \mathbf{x}_j(t)$ and $\Delta\mathbf{u}_{ij}(t) = \mathbf{u}_i(t) - \mathbf{u}_j(t)$. Let $\Delta\mathbf{p}_{ij}(t)$, $\Delta\mathbf{v}_{ij}(t)$ be the relative position and velocity for i^{th} and j^{th} robot respectively. Let the minimum safe distance to be maintained between the robots be d_s. Then it can be verified that the following expression satisfies the condition for $h(\mathbf{x}_i(t))$ given in *Condition II*, where $|\mathbf{u}_i| \leq a_i$ represents the maximum acceleration of the i^{th} robot.

$$h(\Delta\mathbf{x}_{ij}(t)) = \sqrt{2(a_i + a_j)(\| \Delta\mathbf{p}_{ij}(t) \|_2 - d_s)} + \Delta\hat{\mathbf{p}}_{ij}(t)^T\Delta\mathbf{v}_{ij}(t) \tag{14}$$

where $\Delta\hat{\mathbf{p}}_{ij} = \frac{\Delta\mathbf{p}_{ij}}{\|\Delta\mathbf{p}_{ij}\|_2}$. The boundary region of $\mathbf{S_x}$ stated in *Condition II* is given by $\partial\mathbf{S_x} = (\| \Delta\mathbf{p}_{ij} \|_2 = d_s,\ \Delta\hat{\mathbf{p}}_{ij}^T\Delta\mathbf{v}_{ij} = 0)$ for $h(\Delta\mathbf{x}_{ij}(t))$ given in (14). The term $\Delta\hat{\mathbf{p}}_{ij}^T\Delta\mathbf{v}_{ij} = 0$ ensure that the component of relative velocity along the relative position vector is zero when the magnitude of the relative position vector is approaching the safe distance d_s. By substituting the expression for $h(\Delta\mathbf{x}_{ij}(t))$ given by (14) in (12), the following inequality is obtained.

$$\begin{aligned}\Delta\mathbf{p}_{ij}(t)^T\Delta\mathbf{u}_{ij}(t) + \frac{\alpha}{z}h(\Delta\mathbf{x}_{ij}(t))^{(2z+1)} \| \Delta\mathbf{p}_{ij}(t) \|_2 \\ - (\Delta\hat{\mathbf{p}}_{ij}(t)^T\Delta\mathbf{v}_{ij}(t))^2 + \| \Delta\mathbf{v}_{ij}(t) \|_2^2 \\ + \frac{(a_i + a_j)\Delta\mathbf{p}_{ij}(t)^T\Delta\mathbf{v}_{ij}(t)}{\sqrt{2(a_i + a_j)(\| \Delta\mathbf{p}_{ij}(t) \|_2 - d_s)}} \geq 0\end{aligned} \tag{15}$$

The inequality given in (15) need to be satisfied to avoid collision between the i^{th} and j^{th} robot. The above inequality is rewritten as given below.

$$\mathbf{m}(\Delta\mathbf{x}_{ij}(t)) \begin{bmatrix}\mathbf{u}_i(t) & \mathbf{u}_j(t)\end{bmatrix}^T \leq \mathbf{d}_{ij}(\mathbf{x}_{ij}(t)) \tag{16}$$

where

$$\mathbf{m}(\Delta\mathbf{x}_{ij}(t)) = \begin{pmatrix}-\Delta\mathbf{p}_{ij}(t)^T & \Delta\mathbf{p}_{ij}(t)^T\end{pmatrix} \tag{17}$$

$$\begin{aligned}\mathbf{d}_{ij}(\mathbf{x}_{ij}(t))) = \frac{\alpha}{z}h(\Delta\mathbf{x}_{ij}(t))^{(2z+1)} \| \Delta\mathbf{p}_{ij}(t) \|_2 \\ - (\Delta\hat{\mathbf{p}}_{ij}(t)^T\Delta\mathbf{v}_{ij}(t))^2 + \| \Delta\mathbf{v}_{ij}(t) \|_2^2 \\ + \frac{(a_i + a_j)\Delta\mathbf{p}_{ij}(t)^T\Delta\mathbf{v}_{ij}(t)}{\sqrt{2(a_i + a_j)(\| \Delta\mathbf{p}_{ij}(t) \|_2 - d_s)}}\end{aligned} \tag{18}$$

Remark 1. *Let the planning horizon belong to interval $[t_0, t_n]$. Then at the current time instant t, the relative state $\Delta\boldsymbol{x}_{ij}(t_0)$ is known through available sensors. Thus, the barrier constraints given in (16) at time t_0 is affine with respect to control variables $\boldsymbol{u}_i, \boldsymbol{u}_j$.*

One-step Reactive MPC based on convex QP: Remark 1 forms the basis of an efficient reactive planner for collision avoidance between robots. Let $\boldsymbol{\gamma}(\mathbf{x}_i(t))$ be a nominal control law that generates the nominal control action for the i^{th} robot to follow a reference trajectory in the absence of obstacles. Then the solution of the following centralized optimization problem computes the minimum possible deviation from the nominal feedback that is required at the current time instant t_0 to avoid imminent collisions.

$$\begin{aligned}&\underset{\mathbf{u}_i(t_0),\mathbf{u}_j(t_0)}{\text{argmin}} \sum_{i=1}^{m} \| \mathbf{u}_i(t_0) - \boldsymbol{\gamma}(\mathbf{x}_i(t_0)) \|_2^2 \\ &\mathbf{m}(\Delta\mathbf{x}_{ij}(t_0)) \begin{bmatrix}\mathbf{u}_i(t_0) & \mathbf{u}_j(t_0)\end{bmatrix}^T \leq \mathbf{d}_{ij}(\mathbf{x}_{ij}(t_o)), (\forall\, i, j,\, i \neq j) \\ &\| \mathbf{u}_i(t_0) \|_\infty \leq a_i,\ \| \mathbf{u}_j(t_0) \|_\infty \leq a_j\end{aligned} \tag{19}$$

The QP (19) computes control inputs for all the robots simultaneously. To achieve better scalability in terms of the number of robots, the following decentralized version of (19) can be obtained.

$$\begin{aligned}&\underset{\mathbf{u}_i(t_0)}{\text{argmin}} \| \mathbf{u}_i(t_0) - \boldsymbol{\gamma}(\mathbf{x}_i(t_0)) \|_2^2 \\ &\bar{\mathbf{m}}_{ij}(\Delta\mathbf{x}_{ij}(t_0))\mathbf{u}_i(t_0) \leq \bar{\mathbf{d}}_{ij}(\mathbf{x}_{ij}(t_0))\, \forall j \\ &\| \mathbf{u}_i(t) \|_\infty \leq a_i\end{aligned} \tag{20}$$

where $\bar{\mathbf{m}}(\Delta\mathbf{x}_{ij}(t)) = -\Delta\mathbf{p}_{ij}(t)^T$ and $\bar{\mathbf{d}}_{ij}(t) = \frac{a_i}{a_i + a_j}\mathbf{d}_{ij}(t)$. In the QP given in (20), each robot infers the current relative position and velocities of other robots and acts independently.

III. Proposed MPC

The QP based approaches presented in (19) and (20) do not take into account the long-term prediction of the robot's trajectories. Intuitively, this means that the robots initiate collision avoidance when they are close to each other, and the barrier constraint is approaching the boundary of the feasible boundary. This, in turn, forces robots to undertake sharp evasive manoeuvres. As we discuss in the next section, this effect gets accentuated for a FWA that do not have the ability to either stop at a place or generate motions in arbitrary directions at any given time instant. Thus, in this section, we present our main algorithmic contribution, which is a real-time MPC that incorporates long term prediction of robots' trajectories. We first present the derivations for holonomic robots such as quadrotor system and then subsequently extend to handle additional constraints posed by FWA. We begin by deriving the following discrete-time version of the dynamics given in (3) for a sampling time T .

$$\mathbf{x}_i(k+1) = \mathbf{A}_k\mathbf{x}_i(k) + \mathbf{B}_k\mathbf{u}_i(k) \tag{21}$$

where $\mathbf{x}_i(k) = [x_i(k),\, v_{xi}(k),\, y_i(k),\, v_{yi}(k)]^T$, $\mathbf{u}_i(k) = [u_{xi}(k),\, u_{yi}(k)]^T$ and

$$\mathbf{A}_k = \begin{pmatrix}1 & T & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & T\\ 0 & 0 & 0 & 1\end{pmatrix}, \mathbf{B}_k = \begin{pmatrix}0.5T^2 & 0\\ T & 0\\ 0 & 0.5T^2\\ 0 & T\end{pmatrix} \tag{22}$$

The latter developments in this section assume that $\mathbf{x}_i(k)$ is obtained purely as a function of $\mathbf{u}_i(k)$ at different instants k by a forward roll out of (22).

A. MPC for Quadrotors

The optimization underlying our MPC for the i^{th} robot can be written in the following manner

$$\begin{aligned}&\underset{\mathbf{u}_i(k)}{\text{argmin}} \sum_{k=1}^{n} \| \mathbf{u}_i(k) - \gamma(\mathbf{x}_i(k)) \|_2^2 \\ &\text{Subject to } \bar{\mathbf{m}}_{ij}(\Delta\mathbf{x}_{ij}(k))\mathbf{u}_i(k) \leq \bar{\mathbf{d}}_{ij}(\mathbf{x}_{ij}(k))\, \forall j, \forall k \\ &\| \mathbf{u}_i(k) \|_\infty \leq a_i, \forall k\end{aligned} \tag{23}$$

As evident, (23) is a multi-step extension of the reactive planner (20), wherein the cost and constraints are evaluated for all discrete instants k. An important thing to note that is that barring the first instant $k = 0$, i.e for $k \in [1, n]$, the $\bar{\mathbf{m}}_{ij}(\Delta\mathbf{x}_{ij}(k))$ will be highly non-linear and non-convex functions of $\mathbf{u}_i(k)$ for $k \in [1, n-1]$. Thus, (23) represents a difficult non-convex optimization problem. We propose a customized projected gradient descent optimizer for (23) that leverages the niche mathematical structures of the problem. To this end, we first construct an augmented cost function denoted as $\mathcal{L}$ by reformulating the barrier constraints as hinge loss $f(s) = \max(0, s)$ as:

$$\begin{aligned}\mathcal{L} = &\sum_{k=0}^{n} \parallel \mathbf{u}_i(k) - \boldsymbol{\gamma}(\mathbf{x}_i(k)) \parallel_2^2 \\ &+ \sum_{k=0}^{n}\sum_{j=1}^{m} \max(0, \bar{\mathbf{m}}_{ij}(\Delta\mathbf{x}_{ij}(k))\mathbf{u}_i(k) - \bar{\mathbf{d}}_{ij}(\mathbf{x}_{ij}(k)))\end{aligned} \tag{24}$$

The optimization problem given in (23) is reformulated using (24) as shown below.

$$\begin{aligned}&\underset{\mathbf{u}_i(k)}{\operatorname{argmin}} \mathcal{L} \\ &\parallel \mathbf{u}_i(k) \parallel_\infty \leq a_i, \forall k\end{aligned} \tag{25}$$

Gradient Descent Step: The cost function in (25) is non-smooth because of the presence of the hinge-loss function. However, cost functions with similar structure are frequently encountered during the training of deep neural networks and thus can be easily handled by the gradient descent variants like RMSProp [14]. Taking l gradient steps with RMSprop involves computing the following recursive updates.

$$E[\nabla\mathcal{L}^2]_l = 0.9E[\nabla\mathcal{L}^2]_{l-1} + 0.1\nabla\mathcal{L}_l^2, \tag{26}$$

$$\mathbf{u}_i(k) = \mathbf{u}_i(k) - \eta\frac{\nabla\mathcal{L}}{\sqrt{E[\nabla\mathcal{L}_l^2] + \epsilon}}, \tag{27}$$

where η is the so-called learning rate/step-size. We assume some initialization for $\mathbf{u}_i(k)$ to start the gradient descent. We use the warm-start initialization wherein solutions computed at previous MPC iterations are used to initialize the gradient descent at the current step.

Projection Step: The solution $\mathbf{u}_i(k)$ computed through (27) will invariably violate the acceleration bound limits. Furthermore, even though we have incorporated the barrier constraints in the augmented cost function $\mathcal{L}$ and each gradient step will reduce the constraint violation, there is no guarantee that the eventual solution will satisfy the barrier constraints. For example, due to the gradient descent getting stuck in the local minima or forced early stop of the gradient descent due to limited computation time. Thus, we now present our customized projection step that serves two purposes: (i) guaranteed satisfaction of acceleration bounds at all times and barrier constraints at the current time instant and (ii) ensuring that the projection retains smoothness in the control input trajectory. The projection can be written as the following QP problem:

$$\begin{aligned}\arg\min_{\widetilde{\mathbf{u}}_i(k)} \sum_{k=0}^{n} (\overbrace{\| \widetilde{\mathbf{u}}_i(k) - \mathbf{u}_i(k)\|_2^2}^{\textbf{regular projection}} + \|\widetilde{\mathbf{u}}_i(k) - \mathbf{u}_i(k)^{prior}\|_2^2) \\ + \sum_{k=0}^{n-1} (\|\widetilde{\mathbf{u}}_i(k+1) - \widetilde{\mathbf{u}}_i(k)\|_2^2), \\ \parallel \widetilde{\mathbf{u}}_i(k) \parallel_\infty \leq a_i, \forall k \\ \bar{\mathbf{m}}_{ij}(\Delta\mathbf{x}_{ij}(k))\mathbf{u}_i(k) \leq \bar{\mathbf{d}}_{ij}(\mathbf{x}_{ij}(k)),\end{aligned} \tag{28}$$

where $\widetilde{\mathbf{u}}_i(k)$ represents the projected control input. The first term in the cost function represents the conventional l_2 norm associated with projection cost. However, as shown we introduce two additional terms to induce smoothness in the control profile. To be more precise, the second term ensures that projected control input is close to some prior control inputs $\mathbf{u}_i(k)^{prior}$. In our implementation, we choose it to be the feasible control input obtained in the previous MPC iteration. The third term in the cost function minimizes the change in the projected control input over time.

The worst case will happen if the gradient steps (27) do not make any progress towards minimizing the augmented cost function. In this case, the projection step will essentially compute a feasible control input that satisfies the barrier constraints corresponding to the relative state at the current time instant. However, in practice, the gradient descent will invariably guide the solution towards the feasible region of barrier constraints over longer horizons.

B. Extension to Fixed-Wing Aircraft

This section adapts the MPC developed for quadrotors to FWA. We use the same double integrator system (21-22) but introduce additional constraints to model the conditions that a FWA cannot hover at-place and can only execute maneuvers with specified curvature upper bounds κ_{max}. The following expression gives the curvature constraint imposed on the flight trajectory.

$$\mathbf{f}_{curv} : -\kappa_{max} \leq \frac{\dot{x}_i(k)\ddot{y}_i(k) - \ddot{x}_i(k)\dot{y}_i(k)}{(\dot{x}_i(k)^2 + \dot{y}_i(k)^2)^{3/2}} \leq \kappa_{max}, \forall k \tag{29}$$

As can be seen, the curvature bounds are highly non-linear and non-convex. Thus, we adopt an approach similar to barrier constraints and incorporate the curvature bounds as hinge-loss penalties: $\max(0, \mathbf{f}_{curv})$ in the cost function. Furthermore, to ensure guaranteed satisfaction, we leverage the structure in (29). For known velocities at the current time instant, the curvature bounds are affine in the control variables. Thus, we incorporate the one-step curvature bounds in the projection step (28). This in turn, ensures that the immediate control actions that FWA chooses always satisfy the curvature bounds. The velocity bounds for a FWA can be written in the following form.

$$v_{min} \leq \parallel \mathbf{v}_i(k) \parallel_2 \leq v_{max}, \forall k \tag{30}$$

The forward velocity constraints are again non-linear. However, it is possible to derive a computationally simpler BF for them. To this end, we define two functions $h_1(\mathbf{x}_i(k)) = v_{max} - \parallel \mathbf{v}_i(k) \parallel_2$ and $h_2(\mathbf{x}_i(k)) = \parallel \mathbf{v}_i(k) \parallel_2 - v_{min}$ and use them to create the following barrier constraints for some constant $\beta > 0$, as given below.

$$\dot{h}_1(\mathbf{x}_i(k)) \geq -\beta h_1(\mathbf{x}_i(k)), \; \dot{h}_2(\mathbf{x}_i(k)) \geq -\beta h_2(\mathbf{x}_i(k)) \tag{31}$$

Now, $\dot{h}_1$ and $\dot{h}_2$ will be an affine function of acceleration for the current time step, but it will be non-linear and non-convex over the future time steps which is exactly the same structure we encountered for barrier constraints derived earlier for collision avoidance. Thus, we create a new hinge-loss penalty as $\sum_{k=1}^{n} \max(0, -\dot{h}_1 - \beta h_1) + \max(0, -\dot{h}_2 - \beta h_2)$ and add this to the RMSProp gradient descent. Finally, when doing the projection, we write (31) for the current time step as a linear function of $\mathbf{u}_i(k)$ and append them to the current set of inequalities.

IV. Numerical evaluations

This section provides the numerical evaluations showing the performance of the proposed collision avoidance algorithm for both quadrotors and FWA. The algorithm has been implemented using Python3 on a computer with Intel® Neon® CPU E5-2640 v4 running at 2.40 GHz. The gradient is computed using Autograd [16] library, which uses an automatic differentiation technique for the gradients, which empowers it to handle non-smooth functions as well. Control inputs have been predicted from the gradients using RMSprop. These control inputs were then later provided to the projection step, which generated the control commands that satisfy the BF using cvxpy [17].

A. Collision avoidance for quadrotors

This section aims to evaluate the collision avoidance algorithm for a benchmark condition in which eight quadrotors are positioned at a circle's circumference with radius 70m and destined to reach a diametrically opposite goal position. A PD controller is used to generate the nominal control action, $\boldsymbol{\gamma}(\mathbf{x}_i(k))$ for the quadrotors. The simulations are performed for the algorithm for three prediction horizon values: $n = 5$, $n = 10$ and $n = 15$ for a sampling time interval of $T = 0.1$s. RMSprop is executed for one iteration with $\eta = 0.1$ to get the control input. The acceleration and speed limit of a quadcopter was set to 2ms^{-2} and 10ms^{-1} respectively. The performance is compared with the conventional method denoted by $n = 1$. It can be inferred from Fig. 1 that the trajectories get smoother as the prediction horizon increases. Four parameters are considered here to quantify the performance: a) average of time taken to reach the destination, b) average of control effort, c) average of control smoothness (C_s) and d) average of total distance travelled (D_t) to reach the destination. A decrease in all the four-parameter values indicates an improvement in the performance as given in Table I. There is a considerable improvement in the performance for $n = 5$, $n = 10$ and $n = 15$ case when compared to the conventional $n = 1$ case. The improvement in performance obtained when the prediction horizon increased from $n = 5$ to $n = 10$ is higher than that when obtained by increasing n from 10 to 15. When compared to $n = 1$ case, $n = 15$ shows the best performance with more than 50% reduction in the average time taken to reach the destination, 33.9% reduction in the average control effort, 40.8% improvement in the smoothness of the control action, and 15.8% reduction in the D_t. The performance of the individual quadrotors is better for higher values of n when compared to $n = 1$ case.

TABLE I

% DECREASE IN PARAMETER VALUE W.R.T n=1 FOR QUADROTORS

Parameter	n=5	n=10	n=15
Total Time	31.6	50.5	54
Control effort	22.9	30.5	33.9
C_s	-14.4	-33.8	-40.8
D_t	10.8	15.3	15.8

TABLE II

% DECREASE IN PARAMETER VALUE W.R.T n=1 FOR FWA

Parameter	n=5	n=10
Total Time	64.7	74.7
Control effort	41.1	45
C_s	-53.4	-54.58
D_t	30.6	31.5

B. Collision avoidance for fixed-wing aircraft

The proposed algorithm is extended for the collision avoidance of non-holonomic robots like FWA. Pure proportional navigation (PPN) based algorithm is used to generate the nominal control action, $\boldsymbol{\gamma}(\mathbf{x}_i(k))$ for the FWA [18]. RMSprop is executed for only two iterations with $\eta = 0.02$ to get the desired control inputs. The maximum acceleration for an FWA is set to 5ms^{-2}, and the flight velocity is kept between 8ms^{-1} and 18ms^{-1}. The minimum turn radius is set to 30m for all the FWA. The algorithm is tested on a standard benchmark of ten FWA, having their initial position on the circumference of a circle of radius 500 m and destined to reach the diametrically opposite goal. The resulting trajectories of the ten FWA is shown in Fig. 2 for $n = 1$, $n = 5$ and $n = 10$. Similar to the case of quadrotors, the trajectory gets smoother for FWA as n increases from 1 to 10. The quantitative performance is given in Table II, for the four parameters. There is close to 75% reduction in the average time taken to reach the goal, 31.5 % reduction in the D_t by the FWA for $n = 10$ when compared to $n = 1$. Whereas for the case of average control effort and C_s, an improvement of 45% and close to 55% is obtained respectively. As with the increase in n beyond a value of 10, the computational time increases to a value that limits the operation of the system to be in real-time. So the implementation aspects of the algorithm for FWA, for the values of $n > 10$ needs to be further investigated.

V. Conclusion

The paper presented a MPC based extension of the single-step collision avoidance method using the CBF. Considering the second-order translational dynamics of both quadrotor and FWA, an optimization problem is formulated that accounts for the collision avoidance and the robot constraints for the entire prediction horizon. A customized projected gradient descent method is used to solve the resulting nonlinear and non-convex optimization problem. Extensive numerical simulation results are presented for eight quadrotors and ten FWA showing the advantages of the proposed method when compared to the conventional single-step CBF based collision avoidance method. There are significant savings in

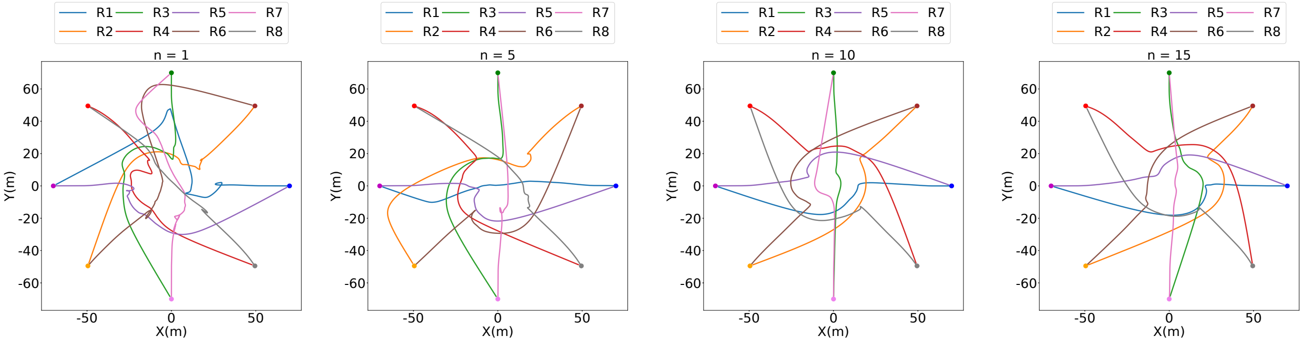

Fig. 1. Trajectories of eight quadrotors for different values of n.

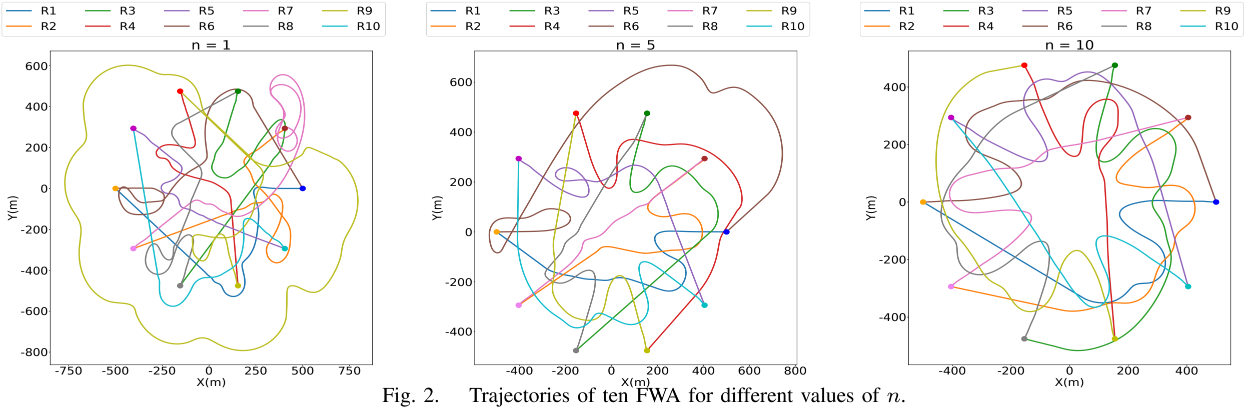

Fig. 2. Trajectories of ten FWA for different values of n.

the total mission time, reduction in path length and control effort for the proposed method when compared to the single-step method. Moreover, the trajectories get smoother with an increase in the value of the prediction horizon. Real-time implementation of the proposed algorithm to the robotic hardware would be an extension of the current work.

References

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

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

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

[4] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 539–557, 2004.

[5] D. H. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 4, 2003, pp. 3621–3626 vol.4.

[6] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 604–611, 2020.

[7] B.-O. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke, "The branching-course model predictive control algorithm for maritime collision avoidance," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1222–1249, 2019.

[8] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *The International Journal of Robotics Research*, p. 0278364920916531, 2020.

[9] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[10] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[11] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[12] F. Ferraguti, M. Bertuletti, C. T. Landi, M. Bonfè, C. Fantuzzi, and C. Secchi, "A control barrier function approach for maximizing performance while fulfilling to iso/ts 15066 regulations," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5921–5928, 2020.

[13] R. Grandia, A. J. Taylor, A. Singletary, M. Hutter, and A. D. Ames, "Nonlinear model predictive control of robotic systems with control lyapunov functions," *arXiv preprint arXiv:2006.01229*, 2020.

[14] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[15] S. Prajna, A. Jadbabaie, and G. J. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1415–1428, 2007.

[16] D. Maclaurin, D. Duvenaud, and R. P. Adams, "Autograd: Effortless gradients in numpy," in *ICML 2015 AutoML Workshop*, vol. 238, 2015.

[17] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[18] K. Harikumar, J. V. Pushpangathan, S. Dhall, and M. S. Bhat, "Integrated guidance and control framework for the waypoint navigation of a miniature aircraft with highly coupled longitudinal and lateral dynamics," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Oct-2020.