Chance-Constrained Collision Avoidance for MAVs in Dynamic Environments

Hai Zhu and Javier Alonso-Mora

***Abstract*—Safe autonomous navigation of micro air vehicles in cluttered dynamic environments is challenging due to the uncertainties arising from robot localization, sensing and motion disturbances. This paper presents a probabilistic collision avoidance method for navigation among other robots and moving obstacles, such as humans. The approach explicitly considers the collision probability between each robot and obstacle and formulates a chance constrained nonlinear model predictive control problem (CCNMPC). A tight bound for approximation of collision probability is developed which makes the CCNMPC formulation tractable and solvable in real time. For multi-robot coordination we describe three approaches, one distributed without communication (constant velocity assumption), one distributed with communication (of previous plans) and one centralized (sequential planning). We evaluate the proposed method in experiments with two quadrotors sharing the space with two humans and verify the multi-robot coordination strategy in simulation with up to sixteen quadrotors.**

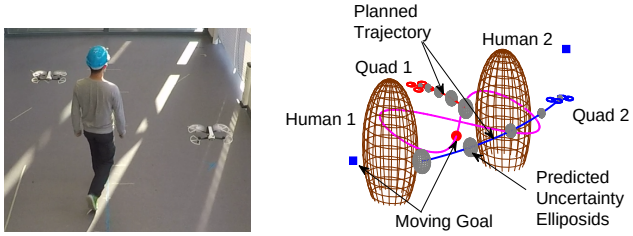
***Index Terms*—Path Planning for Multiple Mobile Robots or Agents, Collision Avoidance, Motion and Path Planning.**

I. Introduction

ON-LINE generation of collision free trajectories is of utmost importance for safe navigation among other robots and in human-populated environments. In these crowded and dynamic scenarios, reasoning about the uncertainties in self-localization, in estimation of the motion of other agents and in motion execution becomes increasingly relevant. Furthermore, tight coordination between the robots becomes essential.

In this paper, we present a probabilistic collision avoidance method for Micro Aerial Vehicles (MAVs) that accounts for robot localization and sensing uncertainties, as well as motion disturbances. The method leverages chance-constrained nonlinear model predictive control (CCNMPC) to plan a local trajectory, which ensures that the collision probability between each robot and obstacle is below a user specified threshold. By assuming that the uncertainties are Gaussian distributed, we transform the chance constraints into deterministic constraints on the robots' state mean and covariance. Thus, a tractable constrained optimization problem is obtained and solved in a receding horizon fashion and on-line.

Furthermore, we discuss and compare three methods for planning among other robots, a distributed approach where only the sensed velocity and position of neighboring robots are used, a distributed approach where previous plans of other robots are communicated, and a centralized approach for multi-robot coordination where a sequential planning scheme is employed.



(a) Snapshot from experiment

(b) Schematic of quadrotors, human, plan and uncertainties

Fig. 1: Probabilistic collision avoidance among obstacles.

The main contributions of this work are:

- An on-line collision avoidance method for navigation in three dimensional dynamic environments, which utilizes stochastic nonlinear model predictive control to plan safe trajectories with a specified probability of collision.
- A tighter bound for chance constraints over ellipsoidal obstacles, which accounts for robot localization, sensing uncertainties and disturbances.
- Incorporation of collision avoidance chance constraints into three frameworks for multi-robot motion planning (sequential, distributed with/without communication).

We evaluate our proposed method in experiments with a team of quadrotors, see Fig. 1 for an example with two quadrotors avoiding two walking humans.

II. Related Work

Several approaches exist for collision avoidance in dynamic environments among other MAVs, which include velocity obstacles [1], decentralized NMPC [2] and sequential NMPC [3]. However, these approaches were deterministic and did not account for uncertainties in perception and motion. The concept of velocity obstacles was extended to handle motion uncertainties by using conservative bounding volumes [4]. Yet, the robot dynamics were not fully modeled and the motion was limited by planning a constant velocity motion. These issues can be overcome by using NMPC for planning. [5] introduced a decentralized NMPC where robot motion uncertainties were taken into account by enlarging the robots with their 3-σ confidence ellipsoids. However, bounding volumes can be conservative and lead to infeasible solutions in cluttered

Manuscript received: September 10, 2018; Revised December 1, 2018; Accepted December 28, 2018.

This paper was recommended for publication by Editor Nak Young Chong upon evaluation of the Associate Editor and Reviewers' comments. This work has been supported in part by the Netherlands Organisation for Scientific Research, Applied Sciences domain (NWO-TTW Veni).

The authors are with the Department of Cognitive Robotics, Delft University of Technology, Mekelweg 2, 2628 CD Delft, Netherlands {h.zhu, j.alonsomora}@tudelft.nl.

Digital Object Identifier (DOI): see top of this page.

environments. In this work we explicitly consider the collision probability and formulate a chance constrained NMPC problem. A chance constrained MPC problem was formulated by [6] for systems with linear dynamics and planar motion, where rectangular regions were computed and overlaps avoided in a centralized mixed integer program formulation. Our proposed approach is not centralized and can be applied to robots with nonlinear dynamics navigating in three-dimensional spaces.

If one assumes set-bounded motion uncertainty models, then robust MPC can be employed to plan safe trajectories [7], or a guaranteed trajectory tracking error bound [8] can be used. However, uncertainties described by Gaussian probability distributions, such as those resulting from Kalman filters, are unbounded. If we consider Gaussian distributions, then objects can be approximated by larger bounding volumes that correspond to sigma hulls [9] which are based on confidence levels. With this method collision checking can be performed very fast. However, the enlarged bounding volumes generally overestimate the collision probability [10]. Hence, when navigating in cluttered environments, the approach tends to lead to sub-optimal or infeasible solutions [11].

By assuming a constant probability density of the robot position within the obstacle, the collision probability can be approximated by the density multiplied by the volume occupied by the obstacle. [12] uses the probability density of the center of the obstacle, while [10] uses the maximum density on the surface of the obstacle to provide an upper bound of collision probability. Both methods are fast, but they only work well when the sizes of objects are relatively very small compared with their position uncertainties. The collision probability can be computed directly via sampling [13]. However, this is computationally intensive and thus not eligible for real time collision avoidance.

Another alternative is to consider convex polygonal obstacles [14], [15]. Under the assumption that object positions follow Gaussian distributions, the resulting linear chance constraints can be transformed directly into deterministic constraints of the mean and covariance of the positions. However polygonal obstacles are ill-posed for online constrained optimization, where smooth shapes are preferred to avoid local minima. In this paper, we consider spherical robots and ellipsoidal dynamic obstacles. We locally linearize the nonlinear collision avoidance constraints and the corresponding chance constraints are reformulated into deterministic constraints on the robot's state mean and covariance. Such a linearization technique was used for deterministic multi-agent collision avoidance [16]. We mathematically formalize its use in the context of probability-based stochastic collision avoidance.

III. Preliminaries

Throughout this paper vectors are denoted in bold letters, $\mathbf{x}$, matrices in capital, M, and sets in mathcal, $\mathcal{A}$. $\|\mathbf{x}\|$ denotes the Euclidean norm of $\mathbf{x}$ and $\|\mathbf{x}\|_Q = \mathbf{x}^T Q \mathbf{x}$ denotes the weighted squared norm. A hat $\hat{\mathbf{x}}$ denotes the mean of a random variable $\mathbf{x}$. $\Pr[\cdot]$ indicates the probability of an event and $p[\cdot]$ indicates the probability density function. A superscript $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$. The super index $\cdot^k$ indicates the value at time k. The sub index $\cdot_i$ indicates robot or obstacle i.

A. Robot Model

Consider a multi-robot system with n robots moving in a shared workspace $\mathcal{W} \subseteq \mathbb{R}^3$. We model each robot $i \in \mathcal{I} = \{1, 2, \dots, n\} \subset \mathbb{N}$ as an enclosing rigid *sphere* $\mathcal{S}_i$ with radius r_i. The dynamics of each robot $i \in \mathcal{I}$ are described by a stochastic nonlinear discrete-time model,

$$\mathbf{x}_i^{k+1} = \mathbf{f}_i(\mathbf{x}_i^k, \mathbf{u}_i^k) + \omega_i^k, \quad \mathbf{x}_i^0 \sim \mathcal{N}(\hat{\mathbf{x}}_i^0, \Gamma_i^0), \tag{1}$$

where $\mathbf{x}_i^k = [\mathbf{p}_i^k, \mathbf{v}_i^k, \phi_i^k, \theta_i^k, \psi_i^k]^T \in \mathcal{X}_i \subset \mathbb{R}^{n_x}$ denotes the state of the robot (position, velocity and orienting) and $\mathbf{u}_i^k \in \mathcal{U}_i \subset \mathbb{R}^{n_u}$ the control inputs at time k. $\mathcal{X}_i$ and $\mathcal{U}_i$ are the state space and control space respectively. The initial state $\mathbf{x}_i^0$ is considered as a Gaussian random variable with mean $\hat{\mathbf{x}}_i^0$ and covariance Γ_i^0, which are typically given by a state estimator (we employ an Unscented Kalman Filter (UKF)). $\mathbf{f}_i$ denotes the nonlinear dynamics. We consider uncorrelated process noise $\omega_i^k \sim \mathcal{N}(0, Q_i^k)$ with diagonal covariance matrix Q_i^k. In this paper, we employ the Parrot Bebop2 quadrotor to evaluate our method. See Appendix for the dynamics model.

B. Obstacle Model

For each obstacle $o \in \mathcal{I}_o = \{1, 2, \dots, n_o\} \subset \mathbb{N}$ at position $\mathbf{p}_o \in \mathbb{R}^3$, we model it as a non-rotating enclosing *ellipsoid* $\mathcal{S}_o$ with semi-principal axes (a_o, b_o, c_o) and rotation matrix R_o. Static obstacle positions are assumed to be available for planning. For dynamic obstacles, as in [17], we assume a constant velocity model with Gaussian noise $\omega_o(t) \sim \mathcal{N}(0, Q_o(t))$ in acceleration, i.e. $\ddot{\mathbf{p}}_o(t) = \omega_o(t)$. Given measured obstacle's position data, we estimate and predict their future positions and uncertainties with a linear Kalman Filter.

C. Collision Chance Constraints

1) Collision Condition: The collision condition of robot i with respect to robot j at time k is defined as

$$C_{ij}^k \coloneqq \{\mathbf{x}_i^k \,|\, \|\mathbf{p}_i^k - \mathbf{p}_j^k\| \leq r_i + r_j\} \,. \tag{2}$$

Collision checking between a robot i and an obstacle o requires calculating the minimum distance between a sphere and an ellipsoid, which can not be performed in closed form [18]. To this end, we approximate the obstacle with an enlarged ellipsoid and check if the robot's position is inside it. The collision condition is

$$C_{io}^k \coloneqq \{\mathbf{x}_i^k \,\Big|\, \|\mathbf{p}_i^k - \mathbf{p}_o^k\|_{\Omega_{io}} \leq 1\} \,, \tag{3}$$

where $\Omega_{io} = R_o^T \mathrm{diag}(1/(a_o + r_i)^2, 1/(b_o + r_i)^2, 1/(c_o + r_i)^2) R_o$.

2) Chance Constraints: The positions of the robots and obstacles are random variables described by unbounded probability distributions. Hence, the collision avoidance constraints can only be satisfied in a probabilistic manner, which are formulated as chance constraints for robot i:

$$\Pr(\mathbf{x}_i^k \notin C_{ij}^k) \geq 1 - \delta_r, \quad \forall j \in \mathcal{I}, j \neq i \tag{4}$$

$$\Pr(\mathbf{x}_i^k \notin C_{io}^k) \geq 1 - \delta_o, \quad \forall o \in \mathcal{I}_o \tag{5}$$

where δ_r, δ_o are the probability thresholds for inter-robot and robot-obstacle collision respectively.

D. Problem Formulation

We formulate a distributed collision avoidance problem. For each robot $i \in \mathcal{I}$, we formulate a discrete time chance constrained optimization problem with N time steps and planning horizon $\tau = N\Delta t$, where Δt is the time step.

Problem 1. *(Probabilistic Collision Avoidance with Chance Constraints) For robot i, given the position distributions $\mathbf{p}_j^{0:N}$ of other robots $j \in \mathcal{I}, j \neq i$ and position distributions $\mathbf{p}_o^{0:N}$ of obstacles $o \in \mathcal{I}_o$, the initial state $\hat{\mathbf{x}}_i^0$ with uncertainty covariance Γ_i^0, the goal position $\mathbf{p}_{ig}$, and the collision probability thresholds δ_r, δ_o, the objective is to compute optimal trajectories and control inputs for the robot to progress from its initial state to its goal while the collision probability with each obstacle and robot is below given thresholds. The resulting optimization problem is*

$$\min_{\hat{\mathbf{x}}_i^{1:N}, \mathbf{u}_i^{0:N-1}} \quad \sum_{k=0}^{N-1} J_i^k(\hat{\mathbf{x}}_i^k, \mathbf{u}_i^k) + J_i^N(\hat{\mathbf{x}}_i^N) \tag{6a}$$

$$s.t. \quad \mathbf{x}_i^0 = \hat{\mathbf{x}}_i(0), \quad \hat{\mathbf{x}}_i^k = \mathbf{f}_i(\hat{\mathbf{x}}_i^{k-1}, \mathbf{u}_i^{k-1}), \tag{6b}$$

$$\Pr(\mathbf{x}_i^k \notin C_{ij}^k) \geq 1 - \delta_r, \forall j \in \mathcal{I}, j \neq i \tag{6c}$$

$$\Pr(\mathbf{x}_i^k \notin C_{io}^k) \geq 1 - \delta_o, \forall o \in \mathcal{I}_o \tag{6d}$$

$$\mathbf{u}_i^{k-1} \in \mathcal{U}_i, \quad \hat{\mathbf{x}}_i^k \in \mathcal{X}_i, \tag{6e}$$

$$\forall k \in \{1, \dots, N\}.$$

where J_i^k denotes the cost term of the robot at time k and J_i^N denotes the terminal cost.

Remark 1. *The positions of other robots and obstacles $\mathbf{p}_j^{0:N}$, $\mathbf{p}_o^{0:N}$ are assumed to follow a Gaussian distribution.*

Remark 2. *In Sec. V, we describe several assumptions to obtain the predicted positions $\mathbf{p}_j^{0:N}$ of other robots.*

E. Approximate Uncertainty Propagation

Evaluating the chance constraints (6c) and (6d) requires calculating the uncertainty covariance at each time step, i.e. uncertainty propagation. There are many methods to perform uncertainty propagation for nonlinear systems, for example the unscented transformation [19] and polynomial chaos expansions [20]. The readers can refer to [21] to get a comprehensive review. However, these methods are mostly computationally intensive and only outperform linearization methods when the propagation time is very long. In our case where the planning horizon is short, to achieve real time performance, we propagate uncertainties using a EKF-type update, i.e. $\Gamma_i^{k+1} = F_i^k \Gamma_i^k {F_i^k}^T + Q_i^k$, where Γ_i^k is the state uncertainty covariance at time k, Q_i^k is the process noise and $F_i^k = \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i}\Big|_{\hat{\mathbf{x}}_i^{k-1}, \mathbf{u}_i^k}$ is the state transition matrix of the robot. We further denote by Σ_i^k the 3x3 covariance matrix of the position $\mathbf{p}_i^k$, extracted from Γ_i^k.

Remark 3. *The covariance dynamics are dependent on the robot state and control inputs. Hence, it requires $\frac{N}{2}(n_x^2 + n_x)$ additional variables in the optimization Problem 1, which can increase the computation time greatly. In this paper, to avoid the need of additional variables, and similar to [22], we propagate the robot uncertainties based on its last-loop trajectory and control inputs.*

Remark 4. *If the initial state uncertainty is Gaussian, the predicted state uncertainties are Gaussian distributed when propagated using the linearized update with F_i^k computed from the last-loop trajectory and control inputs.*

IV. Chance Constraints Formulation

We now present the method to address the chance constraints of Eq. (6c) and (6d). The basic idea is to first linearize the collision conditions of Eq. (2) and (3) to get linear chance constraints and then reformulate them into deterministic constraints on the mean and covariance of the robot states.

A. Linear Chance Constraints

Consider a linear chance constraint in the form $\Pr(\mathbf{a}^T\mathbf{x} \leq b) \leq \delta$, where $\mathbf{x} \in \mathbb{R}^{n_x}$ is a random variable, $a \in \mathbb{R}^{n_x}, b \in \mathbb{R}$ are constants and δ is the level of confidence. Assuming that $\mathbf{x}$ follows a Gaussian distribution, the chance constraint can be transformed into a deterministic constraint [14].

Lemma 1. *Given a multivariate random variable $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \Sigma)$, then*

$$\Pr(\mathbf{a}^T\mathbf{x} \leq b) = \frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{b - \mathbf{a}^T\hat{\mathbf{x}}}{\sqrt{2\mathbf{a}^T\Sigma\mathbf{a}}}\right),$$

where $\text{erf}(\cdot)$ *is the standard error function defined as* $\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$.

Lemma 2. *Given a multivariate random variable $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \Sigma)$ and a probability threshold $\delta \in (0, 0.5)$, then*

$$\Pr(\mathbf{a}^T\mathbf{x} \leq b) \leq \delta \iff \mathbf{a}^T\hat{\mathbf{x}} - b \geq c,$$

where $c = \text{erf}^{-1}(1 - 2\delta)\sqrt{2\mathbf{a}^T\Sigma\mathbf{a}}$.

Given the level of confidence, the corresponding error function and its inverse can be obtained by table look-up or using series approximation techniques.

B. Inter-robot Collision Avoidance Chance Constraints

We now consider the inter-robot collision avoidance constraints, Eq. (6c). For simplicity, we omit the superscript $\cdot^k$ in this section. Given positions and uncertainty covariances of the two robots $\mathbf{p}_i \sim \mathcal{N}(\hat{\mathbf{p}}_i, \Sigma_i)$, $\mathbf{p}_j \sim \mathcal{N}(\hat{\mathbf{p}}_j, \Sigma_j)$, the instantaneous collision probability of robot i with robot j is

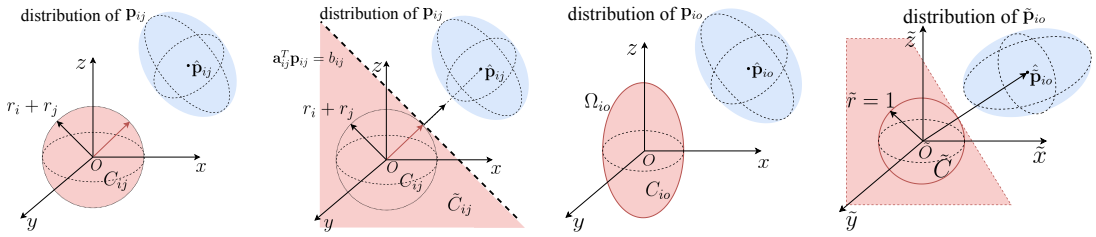
$$\Pr(\mathbf{x}_i \in C_{ij}) = \int_{\mathbb{R}^3} I_C(\mathbf{p}_i, \mathbf{p}_j) p(\mathbf{p}_i) p(\mathbf{p}_j) d\mathbf{p}_i d\mathbf{p}_j, \tag{7}$$

where I_C is the indicator function

$$I_C(\mathbf{p}_i, \mathbf{p}_j) = \begin{cases} 1, & \text{if } \|\mathbf{p}_i - \mathbf{p}_j\| \leq r_i + r_j; \\ 0, & \text{otherwise.} \end{cases}$$

We assume that $\mathbf{p}_i$ and $\mathbf{p}_j$ are independent Gaussian distributions, then $\mathbf{p}_i - \mathbf{p}_j$ is also a Gaussian distribution, i.e. $\mathbf{p}_i - \mathbf{p}_j \sim \mathcal{N}(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j, \Sigma_i + \Sigma_j)$. Hence, the collision probability defined by Eq. (7) can be written as

$$\Pr(\mathbf{x}_i \in C_{ij}) = \int_{\|\mathbf{p}_i - \mathbf{p}_j\| \leq r_i + r_j} p(\mathbf{p}_i - \mathbf{p}_j) d(\mathbf{p}_i - \mathbf{p}_j),$$



(a) Spherical region (b) Linearization of (a) (c) Ellipsoidal region (d) Linearization of (c)

Fig. 2: Chance constraints linearization. Red: collision region. Blue: confidence ellipsoid representation of the Gaussian distributed robot-robot/obstacle relative position. (a) Collision constraint with a sphere region; (b) Linearization with a half space; (c) Collision constraint with an ellipsoid region; (d) Transformation into a unit sphere region and linearization.

which is an integral of a multivariate Gaussian probability density function over a sphere, as illustrated in Fig. 2a.

However, there is no closed form to calculate the collision probability. But we can obtain an approximated upper bound by linearizing the collision condition. As shown in Fig. 2b, we enlarge the spherical collision region C_{ij} into a half space $\tilde{C}_{ij}$, which is defined as

$$\tilde{C}_{ij} \coloneqq \left\{ \mathbf{x} \,\middle|\, \mathbf{a}_{ij}^T(\mathbf{p}_i - \mathbf{p}_j) \leq b_{ij} \right\},$$

where $\mathbf{a}_{ij} = (\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j)/\left\|\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j\right\|$ and $b_{ij} = r_i + r_j$.

It is apparent that $C_{ij} \subset \tilde{C}_{ij}$, thus $\Pr(\mathbf{x}_i \in C_{ij}) \leq \Pr(\mathbf{x}_i \in \tilde{C}_{ij})$. Hence, following Lemma 1, we can obtain an upper bound of the collision probability between two robots:

$$\Pr(\mathbf{x}_i \in C_{ij}) \leq \frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{b_{ij} - \mathbf{a}_{ij}^T(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j)}{\sqrt{2\mathbf{a}_{ij}^T(\Sigma_i + \Sigma_j)\mathbf{a}_{ij}}} \right). \quad (8)$$

Following Lemma 2, the collision chance constraint of Eq. (6c) can be transformed into a deterministic constraint,

$$\mathbf{a}_{ij}^T(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j) - b_{ij} \geq \text{erf}^{-1}(1 - 2\delta_r)\sqrt{2\mathbf{a}_{ij}^T(\Sigma_i + \Sigma_j)\mathbf{a}_{ij}}. \quad (9)$$

C. Robot-obstacle Collision Chance Constraints

For the collision avoidance constraints of Eq. (6d), by assuming that the positions of the robot and obstacle are independent random variables, the collision probability is

$$\Pr(\mathbf{x}_i \in C_{io}) = \int_{\|\mathbf{p}_i - \mathbf{p}_o\|_{\Omega_{io}} \leq 1} p(\mathbf{p}_i - \mathbf{p}_o) d(\mathbf{p}_i - \mathbf{p}_o), \quad (10)$$

where the collision region C_{io} described by Ω_{io} is an ellipsoid instead of a sphere, as shown in Fig. 2c.

To linearize the collision condition, we first do the affine coordinate transformation $\tilde{\mathbf{y}} = \Omega_{io}^{\frac{1}{2}}\mathbf{y}$. Then the collision region is transformed into a unit sphere $\tilde{C}_{io}$, as illustrated in Fig. 2d. The robot and obstacle positions are transformed to new Gaussian distributions, i.e. $\tilde{\mathbf{p}}_i \sim \mathcal{N}(\hat{\tilde{\mathbf{p}}}_i, \tilde{\Sigma}_i)$, $\tilde{\mathbf{p}}_o \sim \mathcal{N}(\hat{\tilde{\mathbf{p}}}_o, \tilde{\Sigma}_o)$, where

$$\begin{aligned} \hat{\tilde{\mathbf{p}}}_i &= \Omega_{io}^{\frac{1}{2}}\hat{\mathbf{p}}_i, \quad \tilde{\Sigma}_i = \Omega_{io}^{\frac{1}{2}T}\Sigma_i\Omega_{io}^{\frac{1}{2}}; \\ \hat{\tilde{\mathbf{p}}}_o &= \Omega_{io}^{\frac{1}{2}}\hat{\mathbf{p}}_o, \quad \tilde{\Sigma}_o = \Omega_{io}^{\frac{1}{2}T}\Sigma_o\Omega_{io}^{\frac{1}{2}}. \end{aligned} \quad (11)$$

In the new coordinate framework, let

$$\Pr(\tilde{\mathbf{x}}_i \in \tilde{C}_{io}) = \int_{\|\tilde{\mathbf{p}}_i - \tilde{\mathbf{p}}_o\| \leq 1} p(\tilde{\mathbf{p}}_i - \tilde{\mathbf{p}}_o) d(\tilde{\mathbf{p}}_i - \tilde{\mathbf{p}}_o),$$

then we have $\Pr(\mathbf{x}_i \in C_{io}) = \Pr(\tilde{\mathbf{x}}_i \in \tilde{C}_{io})$.

Now, we can use the same linearization method as for the sphere region with $\mathbf{a}_{io} = (\hat{\tilde{\mathbf{p}}}_i - \hat{\tilde{\mathbf{p}}}_o)/\|\hat{\tilde{\mathbf{p}}}_i - \hat{\tilde{\mathbf{p}}}_o\|$ and $b_{io} = 1$. The collision chance constraint of Eq. (6d) can thus be transformed into a deterministic constraint:

$$\begin{aligned} \mathbf{a}_{io}^T\Omega_{io}^{\frac{1}{2}}(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j) - b_{io} \geq \text{erf}^{-1}(1 - 2\delta_o) \\ \cdot \sqrt{2\mathbf{a}_{io}^T\Omega_{io}^{\frac{1}{2}}(\Sigma_i + \Sigma_j)\Omega_{io}^{\frac{1}{2}T}\mathbf{a}_{io}}. \end{aligned} \quad (12)$$

D. Comparison to Other Methods

We compare our method with several state-of-the-art collision probability approximation algorithms using a robot-obstacle proximity example. A point robot at position mean $(0.7, 0.7, 0.8)$ m with covariance diag(0.04,0.04,0.01) m^2 is close to an ellipsoid obstacle at origin with semi-principle axes $(0.6, 0.6, 2.2)$ m. See Table I for the collision probability computation results. The numerical integration result is the exact collision probability and gives a collision probability of 0.011. If we define the collision probability threshold to be $\delta = 0.03$ (thus confidence level 0.97), which corresponds to the 3σ confidence ellipsoid [23], then this configuration is feasible. However, when employing the enlarged bounding volume method [24], or the cube approximation [11], the configuration would be deemed infeasible. The center point PDF approximation approach [12] can give feasible checking results, but the resulting collision probability is significantly smaller than the real value, which may lead to unsafe trajectory planning. Our method thus provides a tighter bound.

V. Local Planning

We now present a tractable MPC formulation for each robot, followed by three approaches to obtain future position information of other robots and a theoretical discussion.

TABLE I: Comparison of collision probability algorithms

Algorithms	Collision probability	Computation time (ms)	Feasible?
Numerical integral	0.011	258.665	Yes
Bounding volume [5]	1	0.011	No
Center point [12]	3.6E-18	0.016	Yes
Cube approx. [11]	0.100	0.044	No
Our method	0.017	0.011	Yes

A. Deterministic MPC Formulation

Let $\mathbf{p}_{ig}$ be the goal position of robot i, we minimize the displacement between its terminal position at the planning horizon and its goal. To this end, the terminal cost is

$$J_i^N(\hat{\mathbf{x}}_i^N) = l_i^N \left\|\mathbf{p}_{ig} - \hat{\mathbf{p}}_i^N\right\| / \left\|\mathbf{p}_{ig} - \hat{\mathbf{p}}_i^0\right\|, \tag{13}$$

where l_i^N is the weight coefficient.

The stage cost is to minimize the robot control inputs

$$J_{i,u}^k(\mathbf{u}_i^k) = l_{i,u}^k \left\|\mathbf{u}_i^k\right\|, \quad k = \{0, 1, \ldots, N-1\}, \tag{14}$$

where $l_{i,u}^k$ is the weight coefficient.

We also introduce a potential field cost to increase the separation between robots and obstacles. Denote by $d_{ij} = \|\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j\|_{\Omega_{ij}}$ the distance between robot i and robot/obstacle j. For time step k the potential field cost is $J_{i,c}^k(\hat{\mathbf{x}}_i^k) = \sum_{j \in \mathcal{I} \cup \mathcal{I}_o, j \neq i} J_{i,j,c}^k(\hat{\mathbf{x}}_i^k)$, with $J_{i,j,c}^k(\hat{\mathbf{x}}_i^k) = l_{i,c}^k(d_{ij}^{\text{safe}} - d_{ij})$ if $d_{ij} < d_{ij}^{\text{safe}}$ or $J_{i,j,c}^k(\hat{\mathbf{x}}_i^k) = 0$ otherwise, where d_{ij}^{safe} is the safe potential field distance and $l_{i,c}^k$ is the weight coefficient.

By transforming the chance constraints into the deterministic constraints presented in Sec. IV and utilizing the above cost terms, the following tractable deterministic MPC formulation for Problem 1 can be derived:

$$\begin{aligned}
\min_{\hat{\mathbf{x}}_i^{1:N}, \mathbf{u}_i^{0:N-1}} \quad & J_i^N(\hat{\mathbf{x}}_i^N) + \sum_{k=0}^{N-1} J_{i,u}^k(\mathbf{u}_i^k) + \sum_{k=1}^{N} J_{i,c}^k(\hat{\mathbf{x}}_i^k) \\
\text{s.t.} \quad & \mathbf{x}_i^0 = \hat{\mathbf{x}}_i(0), \quad \hat{\mathbf{x}}_i^k = \mathbf{f}_i(\hat{\mathbf{x}}_i^{k-1}, \mathbf{u}_i^{k-1}), \\
& g_{ij}^k(\hat{\mathbf{x}}_i^k, \hat{\mathbf{p}}_j^k, \Sigma_i^k, \Sigma_j^k, \delta_r) \leq 0, \\
& g_{io}^k(\hat{\mathbf{x}}_i^k, \hat{\mathbf{p}}_o^k, \Sigma_i^k, \Sigma_o^k, \delta_o) \leq 0, \\
& \mathbf{u}_i^{k-1} \in \mathcal{U}_i, \quad \hat{\mathbf{x}}_i^k \in \mathcal{X}_i, \\
& \forall j \neq i \in \mathcal{I}; \forall o \in \mathcal{I}_o; \forall k \in \{1, \ldots, N\}.
\end{aligned} \tag{15}$$

where g_{ij}^k and g_{io}^k denote the deterministic constraints of Eq. (9) and (12) for probabilistic inter-robot and robot-obstacle collision avoidance respectively, and the position uncertainty covariances Σ_i^k are computed as discussed in Remark 3.

B. Multi-robot Planning

In the CCNMPC formulation the position distribution for all other robots $j \neq i$, given by $\hat{\mathbf{p}}_j^{0:N}$ and $\Sigma_j^{0:N}$, is assumed known. Next we discuss three methods to obtain these values, but the CCNMPC formulation is general and other coordination approaches could be devised.

1) Constant velocity model without communication: By regarding all other robots as dynamic obstacles and employing a constant velocity model, one robot can predict other robots future behaviors based on onboard measurements. Hence, each robot can plan its own trajectory independently and without communication, which leads to a distributed planning scheme for multi-robot collision avoidance.

Given the current position and velocity distribution $\hat{\mathbf{p}}_j^0, \hat{\mathbf{v}}_j^0$ and $\Sigma_{j,pv}^0$ of robot j, we compute

$$\begin{aligned}
&[\hat{\mathbf{p}}_j^k, \hat{\mathbf{v}}_j^k]^T = F_j^k [\hat{\mathbf{p}}_j^{k-1}, \hat{\mathbf{v}}_j^{k-1}]^T, \\
&\Sigma_{j,pv}^k = F_j^k \Sigma_{j,pv}^{k-1} {F_j^k}^T + Q_{j,pv}^k,
\end{aligned} \tag{16}$$

where the state transition matrix $F_j^k = \begin{bmatrix} I_3 & \Delta t I_3 \\ O & I_3 \end{bmatrix}$, Δt is the time step for prediction, $Q_{i,pv}^k$ is the additive process noise of the model. The position uncertainty covariance is $\Sigma_j^k = \Sigma_{j,pv}^k(1:3, 1:3)$.

2) Sequential planning with communication: If the team of robots is centrally controlled, or a fast communication channel is available, higher coordination can be achieved by planning trajectories sequentially, i.e., each robot plans a trajectory that avoids the trajectories of all other robots and then communicates its trajectory (given by $\hat{\mathbf{p}}_i^{0:N}$ and $\Sigma_i^{0:N}$).

Denote by $\mathcal{T}_i^t = \{\hat{\mathbf{p}}_i^{0:N}, \Sigma_i^{0:N}\}|_t$ the trajectory for robot i planned at time t. At the initial time $t = 0$ robot i avoids only the plans $\mathcal{T}_j^0$ of other robots with $j < i$, in a priority scheme. In subsequent time steps, robot i plans a trajectory $\mathcal{T}_i^t$ that avoids $\mathcal{T}_j^t$ for all $j < i$ and $\mathcal{T}_j^{t-\Delta t}$ for all $j > i$.

3) Distributed planning with communication: Robots communicate their planned trajectories. At every time step, every robot avoids the planned trajectories of all other robots in the previous time-step. That is, at time t, robot i plans a trajectory $\mathcal{T}_i^t$ that avoids $\mathcal{T}_j^{t-\Delta t}$ for all $j \neq i \in \mathcal{I}$.

C. Theoretical Discussion

1) Collision avoidance: Our formulation imposes, by construction, that the probability of collision with respect to each obstacle and at every stage of the plan is less or equal than δ_o under a constant velocity assumption for moving obstacles (Sec. III-B) and a simplified propagation model (Sec. III-E). For collision avoidance with other robots in the team, guarantees vary according to the coordination methods (and the associated assumptions) described in Sec. V-B.

2) Probability of collision with any given obstacle: From V-C1, the probability of collision of robot i at time step k with respect to *any* given obstacle can be bounded by

$$\Pr(\mathbf{x}_i^k \in \bigcup_{o=1}^{n_o} C_{io}^k) \leq \sum_{o=1}^{n_o} \Pr(\mathbf{x}_i^k \in C_{io}^k) = n_o \delta_o,$$

where n_o is the number of obstacles. By choosing $\delta_o = \delta_{all}/n_o$, one may specify a joint threshold of collision δ_{all}.

3) Probability of collision for the planned trajectory: From V-C1, at all stages the probability of collision with any given obstacle is less or equal than the specified threshold δ_o. The probability of collision for the whole trajectory of robot i with respect to each obstacle can be bounded by

$$\Pr(\bigvee_{k=1}^{N} (\mathbf{x}_i^k \in C_{io}^k)) \leq \sum_{k=1}^{N} \Pr(\mathbf{x}_i^k \in C_{io}^k).$$

In our case this bound would be $N\delta_o$, but it is over conservative in practice. We argue that, in the context of online receding horizon planning it is beneficial to impose a probability of collision of δ_o for each individual stage - instead of for the whole trajectory - thanks to the fast re-planning and relatively small displacement between stages.

Furthermore, our formulation is consistent with a stochastic formulation of the MPC problem where the chance constraint is defined as a discounted sum of violation probabilities in the finite horizon, as proposed for example by [25]. The

rationality with this formulation is also that by penalizing violation probabilities close to the initial time and relaxing the penalty of violation probabilities in the far future, feasibility of the online optimization is enabled.

The discounted chance constraint with respect to an obstacles is defined as:

$$\sum_{k=1}^{N}(\gamma)^k \Pr(\mathbf{x}_i^k \in C_{io}^k) \leq \delta_o, \tag{17}$$

where $\gamma \in (0,1)$ is the discounting factor.

Lemma 3. *Our formulation provides an upper bound in the discounted probability of collision, i.e. Equation (17) is satisfied, if the discounting factor $\gamma < 0.5$.*

Proof. Our formulation guarantees that $\Pr(\mathbf{x}_i^k \in C_{io}^k) \leq \delta_o, \forall k = 1, \cdots, N$. Hence, the discounted probability of collision satisfies

$$\sum_{k=1}^{N}(\gamma)^k \Pr(\mathbf{x}_i^k \in C_{io}^k) \leq \delta_o \sum_{k=1}^{N}(\gamma)^k = \frac{\gamma(1-\gamma^N)}{1-\gamma}\delta_o.$$

Given $\gamma < 0.5$, we have $\gamma(1-\gamma^N)-(1-\gamma) = 2\gamma-1-\gamma^{N+1} < 0$. Thus, $\frac{\gamma(1-\gamma^N)}{1-\gamma} < 1$. Hence, $\sum_{k=1}^{N}(\gamma)^k \Pr(\mathbf{x}_i^k \in C_{io}^k) \leq \delta_o$. □

In this proof we also employ the conservative bound on the joint probability of collision. Future works should look at obtaining tighter bounds on the joint probability of collision over the whole trajectory.

4) Feasibility: Due to unmodeled dynamics, disturbances, or deviations from the simplifying assumptions, the optimization problem may become infeasible. In those rare situations, our approach is to command the MAVs to decelerate. Typically, the problem becomes feasible again after a small number of steps (below half a second, see Section VI-C).

VI. RESULTS

In this section we describe our implementation of the proposed method and evaluate it in experiments and simulations. A video demonstrating the results accompanies this paper can be found at https://youtu.be/P7SUFEKUP9Q.

A. Experimental Setup

Our experimental platform is the Parrot Bebop 2 quadrotor. The radius of each quadrotor is set as 0.3 m. An external motion capture system (OptiTrack) is used to measure the pose of each quadrotor, which is regarded as the "real" pose. We then add Gaussian noise to the data to simulate the localization uncertainties. The added measurements noise is zero mean with covariance $\Sigma = \text{diag}(0.06 \text{ m}, 0.06 \text{ m}, 0.06 \text{ m}, 0.4 \text{ deg}, 0.4 \text{ deg})^2$. Taking the noisy measurements as inputs, an UKF is employed to estimate the state of quadrotors. Based on our experimental data, the average resulted state estimation error is $\|\hat{\mathbf{p}} - \mathbf{p}\| = 0.05$ m in terms of the quadrotors' position. We use an Intel i7 CPU@2.6GHz computer for the planner and use Robot Operating System (ROS) to send commands to the quadrotors. We rely on the solver Forces Pro [26] to generate optimized NMPC code. The collision probability thresholds are set to $\delta_r = 0.03$ and $\delta_o = 0.03$. By default, the time step used in the NMPC is $\Delta t = 0.05$ s and the total number of steps is $N = 20$. This planning horizon, of one second, is based on the experience and analysis of our previous work [3], [17] and works well in practice in our scenarios.

B. Trajectory Safety and Efficiency Comparisons

In this scenario, we compare our method with a bounding volume MPC approach [5] and a deterministic MPC approach [3]. For all three methods we compute trajectories sequentially and the only difference is the way in which the uncertainties are treated. In the experiment, two quadrotors, initially at $(-1.6, 0, 1.2)$ m and $(1.6, 0, 1.2)$ m, are required to swap their positions. For each approach, we performed the experiment 50 times under three levels of measurements noise: $1/4\Sigma$, Σ and 4Σ. The corresponding average state estimation error for the position, i.e. $\|\hat{\mathbf{p}} - \mathbf{p}\|$, was 0.03 m, 0.05 m and 0.09 m respectively. We measured the minimum distance between the two quadrotors as a safety metric and the total trajectory length and duration as efficiency metrics.

The results of the three approaches are shown in Table II. Under measurements noise of Σ, the purely deterministic approach succeeded in 64% of the trials. With the larger noise level of 4Σ its performance deteriorated to a success rate of only 36%. The two probabilistic approaches succeeded in all runs. However, thanks to a tighter bound for the collision probability approximation, our method achieves the same level of safety as [5] but with more efficient collision avoidance, i.e., the trajectory length and duration are shorter. This efficiency is more apparent when the measurements noise is larger, e.g. with covariance 4Σ.

C. Collision Avoidance in Dynamic Environments

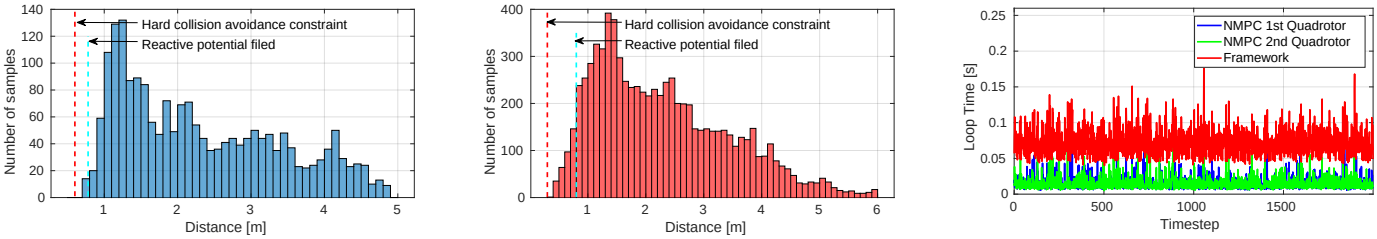
Fig. 1 showed a snapshot from our experiment. In Fig. 3a we cumulate the distance between the two drones. They maintained a safe distance of 0.6 m over the entire run. In Fig. 3b we cumulate the distance between each drone and each moving human. The distance is computed as the closest distance between the quadrotor's position and the ellipsoid's surface. In all instances a minimum safe separation of 0.3 m was achieved. Close distances between robots and obstacles are observed, since they share a quite confined space. In Fig. 3c we show the computation time of each NMPC solver and the central sequential planning framework. The mean computation time of the NMPC solver is 14.3 ms and that of the total framework is 71.3 ms. The framework includes state estimation, uncertainty propagation, obstacles' prediction, communication and solving both NMPC problems. Among all NMPC solutions over the entire run, the percentage of infeasible solutions was 2.8% and the longest infeasible period was 9 time steps (corresponding to 0.45 s).

D. Comparison of Multi-robot Planning Strategies

We evaluate our method in simulation with multiple quadrotors exchanging their initial positions, and compare the three

TABLE II: Trajectory safety and efficiency comparisons of planning algorithms with different levels of measurements noise. The values are computed only from successful runs. ($d_{\min}$: average minimum distance (m); l: average trajectory length (m); T: average trajectory duration (s); sr: success rate.)

Measure noise / Algorithm	$1/4\Sigma$				Σ				4Σ			
	$d_{\min}$	l	T	sr	$d_{\min}$	l	T	sr	$d_{\min}$	l	T	sr
Our method	0.74	6.77	2.63	100%	0.81	7.08	2.72	100%	0.86	7.21	3.06	100%
Bounding volume [5]	0.74	6.84	2.91	100%	0.87	7.09	2.95	100%	1.10	8.18	3.13	100%
Deterministic MPC [3]	0.63	6.75	2.60	68%	0.64	6.74	2.63	64%	0.61	6.88	2.62	36%



(a) Measured inter-drone distance (b) Distance to moving obstacles (c) NMPC and framework loop time

Fig. 3: Experimental results of two quadrotors following predefined paths while avoiding two walking humans.

multi-robot coordination strategies described in Section V-B, with a noise level of Σ. Figures 4a-4c show the trajectories of six quadrotors, where the only difference is the coordination strategies. Table III shows the minimum distance among quadrotors and statistics of their trajectories. We report the average computation time and the trajectory length for all six quadrotors (minimum, maximum, mean value and standard deviation to compare cooperativeness).

We observe that the minimum distance when using the constant velocity model (0.56 m) is smaller than the safe distance (0.6 m). Thus, collisions happened due to the mismatch between the predicted trajectories (constant velocity) and the executed trajectories by the quadrotors. This indicates that the 97% confidence level is not enough when the constant velocity model is employed and should be increased. Instead, sequential planning (SP) and distributed planning with communication (DC) can achieve safe navigation. While SP showed better performance, it suffers from a computation burden due to its centralized scheme (the computational cost grows linearly with the number of robots). The DC approach performs well at a much lower computational cost.

Since the DC approach is scalable, in Fig. 4d we show the trajectories of sixteen quadrotors exchanging antipodal positions on the circle. We note that the computational time of solving the CCMPC for each robot does increase with the number of obstacles and robots, due to the larger number of constraints. In our experiments, the average computation time of a CCMPC planning step was 14.3 ms for two robots, 14.4 ms for four robots, 16.2 ms for six robots and 24.7 ms for sixteen robots. This indicates that the DC approach scales well with the number of robots.

TABLE III: Statistics for coordination strategies with six drones. CV: constant velocity model; SP: sequential planning; DC: distributed planning with communication.

Coordination strategies	Min. dist (m)	Trajectory length (m)				Av.comp. time (ms)
		min.	max.	av.	std	
CV	0.56	4.82	7.09	5.72	0.89	15.2
SP	0.70	4.31	4.54	4.43	0.09	115.3
DC	0.70	4.18	4.80	4.51	0.24	16.2

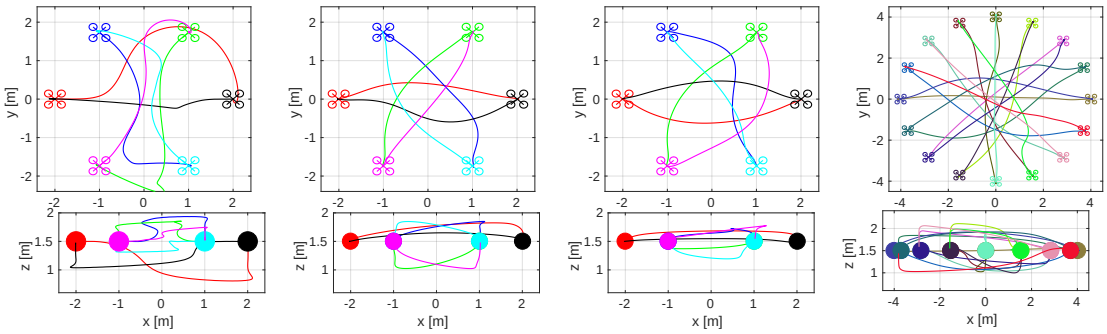
VII. Conclusion

In this paper we showed that robust probabilistic collision avoidance among robots and obstacles can be achieved via chance constrained nonlinear model predictive control when the obstacles are modeled as ellipsoids. By assuming that the uncertainties are Gaussian distributed, we developed a tight bound for approximation of collision probability between each robot and obstacle. In experiments with two quadrotors we showed that our method can generate more efficient trajectories for the robots while maintaining the same level of safety compared with the bounding volume approach. In simulations with six quadrotors we showed that the strategies where the planned trajectories are exchanged outperform the constant velocity model. Furthermore, while distributed planning with communication is less cooperative than sequential planning, it scales well with the number of robots. Future works shall explore more elaborated approaches for multi-robot coordination and deadlock avoidance, which may occur since the method is local. By combining our method with a global planner these problems might be resolved.

Appendix

Based on the Parrot Bebop2 SDK, the control inputs to the quadrotor are given by $\mathbf{u} = [\phi_c, \theta_c, v_{z_c}, \dot{\psi}_c]^T \in \mathbb{R}^4$, where ϕ_c and θ_c are commanded roll and pitch angles, v_{z_c} is the commanded velocity in vertical z direction and $\dot{\psi}_c$ is the commanded angular velocity around the z-body axis. The state $\mathbf{x} \in \mathbb{R}^9$ was defined in Sec. III-A. We use a first order low-pass Euler approximation of the quadrotor dynamics [27], where the dynamics of the state velocity vector are

$$\begin{cases} \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \end{bmatrix} = R_Z(\psi) \begin{bmatrix} \tan\theta \\ -\tan\phi \end{bmatrix} g - k_D \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \\ \dot{v}_z = \frac{1}{\tau_{v_z}}(k_{v_z} v_{z_c} - v_z), \end{cases}$$



(a) Constant velocity (b) Sequential planning (c) Distributed planning (d) Distributed planning

Fig. 4: Simulation results of multiple quadrotors exchanging positions. Solid lines represent the trajectories executed by the quadrotors. The upper and lower plots show the top view (X-Y) and side view(X-Z) respectively.

where $g = 9.81$ m/s^2 is the earth's gravity, $R_Z(\psi) \in SO(2)$ is the rotation matrix along the z-body axis, k_D is the drag coefficient, k_{v_z} and τ_{v_z} are the gain and time constant of vertical velocity.

As in [5], the attitude dynamics of the quadrotor are

$$\dot{\phi} = \frac{1}{\tau_\phi}(k_\phi \phi_c - \phi), \quad \dot{\theta} = \frac{1}{\tau_\theta}(k_\theta \theta_c - \theta), \quad \dot{\psi} = \dot{\psi}_c,$$

where k_ϕ, k_θ and τ_ϕ, τ_θ are the gains and time constant of roll and pitch angles respectively.

REFERENCES

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

[2] D. Shim, H. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *IEEE Conference on Decision and Control*, vol. 4, 2003, pp. 3621–3626.

[3] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges, "Real-time planning for automated multi-view drone cinematography," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–10, 2017.

[4] B. Gopalakrishnan, A. K. Singh, M. Kaushik, K. M. Krishna, and D. Manocha, "Chance constraint based multi agent navigation under uncertainty," in *Proceedings of the Advances in Robotics*. ACM, 2017, p. 53.

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

[6] D. Lyons, J. Calliess, and U. D. Hanebeck, "Chance constrained model predictive control for multi-agent systems with coupling constraints," in *American Control Conference (ACC)*. IEEE, 2012, pp. 1223–1230.

[7] A. Gray, Y. Gao, J. K. Hedrick, and F. Borrelli, "Robust predictive control for semi-autonomous vehicles with an uncertain driver model," in *Proceedings IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 208–213.

[8] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: a modular framework for fast and guaranteed safe motion planning," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 1517–1522.

[9] A. Lee, Y. Duan, S. Patil, J. Schulman, Z. McCarthy, J. van den Berg, K. Goldberg, and P. Abbeel, "Sigma hulls for gaussian belief space planning for imprecise articulated robots amid obstacles," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5660–5667.

[10] C. Park, J. S. Park, and D. Manocha, "Fast and bounded probabilistic collision detection for high-DOF trajectory planning in dynamic environments," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2018.

[11] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 913–929, 2013.

[12] N. E. Du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 809–815, 2011.

[13] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[14] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.

[15] W. Liu and M. H. Ang, "Incremental sampling-based algorithm for risk-aware planning under motion uncertainty," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2051–2058.

[16] D. Morgan, G. P. Subramanian, S. J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *International Journal of Robotics Research*, vol. 35, no. 10, pp. 1261–1285, 2016.

[17] T. Nageli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges, "Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1696–1703, 2017.

[18] A. Y. Uteshev and M. V. Goncharova, "Point-to-ellipse and point-to-ellipsoid distance equation analysis," *Journal of Computational and Applied Mathematics*, vol. 328, pp. 232–251, 2018.

[19] A. Völz and K. Graichen, "Stochastic model predictive control of nonlinear continuous-time systems using the unscented transformation," in *European Control Conference (ECC)*. IEEE, 2015, pp. 3365–3370.

[20] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz, "Stochastic nonlinear model predictive control with probabilistic constraints," in *American Control Conference (ACC)*. IEEE, 2014, pp. 2413–2419.

[21] Y. Z. Luo and Z. Yang, "A review of uncertainty propagation in orbital mechanics," *Progress in Aerospace Sciences*, vol. 89, pp. 23–39, 2017.

[22] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious nmpc with gaussian process dynamics for miniature race cars," in *European Control Conference (ECC)*. IEEE, 2018, pp. 1341–1348.

[23] M. Bensimhoun, "N-dimensional cumulative function, and other useful facts about gaussians and normal densities," Jerusalem, Israel, Tech. Rep., 2009.

[24] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1928–1935.

[25] S. Yan, P. Goulart, and M. Cannon, "Stochastic model predictive control with discounted probabilistic constraints," *arXiv preprint arXiv:1807.07465*, 2018.

[26] A. Domahidi and J. Jerez, "Forces professional. embotech gmbh (http://embotech. com/forces-pro)," 2014.

[27] S. Stevsic, T. Nägeli, J. Alonso-Mora, and O. Hilliges, "Sample efficient learning of path following and obstacle avoidance behavior for quadrotors," *IEEE Robotics and Automation Letters*, 2018.