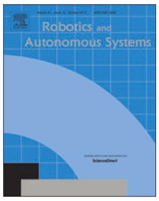
Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Dynamic obstacle avoidance for Multi-rotor UAV using chance-constraints based on obstacle velocity

Takumi Wakabayashi [a], Yukimasa Suzuki [a], Satoshi Suzuki [b,*]

[a] Department of Mechanical Engineering, Chiba University, 1-33 Yayoi-cho Inaga-ku, Chiba, 263-8522, Chiba, Japan
[b] Graduate School of Engineering, Chiba University, 1-33 Yayoi-cho Inaga-ku, Chiba, 263-8522, Chiba, Japan

ARTICLE INFO

Article history:
Received 17 February 2022
Received in revised form 18 September 2022
Accepted 23 November 2022
Available online 26 November 2022

Keywords:
Chance constraints
Velocity obstacles
Collision avoidance
Formation flight
Path planning

ABSTRACT

To ensure the safety of autonomous Multi-rotor UAVs flying in urban airspace, they should be capable of avoiding collisions with unpredictable dynamic obstacles, such as birds. UAVs must consider both relative position and relative velocity to avoid moving obstacles. Model predictive control (MPC) can consider the multiple collision avoidance constraints in a constrained optimisation framework. This study proposes a chance-constraints based on obstacle velocity (CCOV) method, which can be combined with previous positional chance constraint methods to account for uncertainty in both position and velocity. This effectively prevents collision with high-velocity obstacles, even in a noisy environment. The proposed method has been performed on a numerical simulation built in MATLAB.

1. Introduction

In the past, unmanned aerial vehicles (UAVs) have primarily served for military purposes. Currently, private enterprises often utilise them to reduce costs of commercial operations. In the future, small multirotor UAVs will likely be used in a wide range of fields, including inspection work in confined spaces, short-distance transportation, and monitoring of crop growth, due to their low cost and ease of handling. However, reducing the size of UAVs not only relatively increases friction from gears and bearings but also increases viscous losses due to a decrease in the Reynolds number [1]. Thus, flight times become considerably shorter. Therefore, small multirotor UAVs suffer from limitations, such as frequent battery replacement, sensor and camera installation, and limited flight endurance.

Formation flight is a potential method to overcome these limitations; it can improve flight time by enabling individual UAVs to temporarily exit the formation for battery replacement. The overall operation time can be reduced by attaching different sensors to each UAV. Furthermore, it improves fault tolerance by allowing UAVs to compensate for those that are no longer functional. Previous studies on formation flight established that motion planning in the presence of dynamic obstacles is a major hurdle in the operation of UAVs [2,3]. This is because the UAV must avoid external obstacles as well as the remaining UAVs within the formation, necessitating a more robust collision avoidance compared to that in a single UAV operation. In [4], the primary collision avoidance methods for UAVs are presented. An artificial potential field (APF) is designed by setting up a repulsive force for an avoidance target and an attractive force for a target position [5,6]. Although the UAV follows a collision-free path that minimises the cost, the presence of multiple obstacles can cause a phenomenon called deadlock, in which the control target cannot reach the target position. In another method, the problem of collision avoidance is considered as a non-convex optimisation problem, which is iteratively solved using sequential convex programming that approximates non-convex constraints using convex ones [7]. However, the feasibility of the trajectory obtained requires verification because it does not completely consider the UAV dynamics model.

Model predictive control (MPC) predicts the finite-time future using the control input sequence and UAV dynamics. Additionally, it predicts the next control input by solving an optimal control problem that minimises a given evaluation function. In the evaluation function, multiple collision avoidance constraints can be considered, in addition to the optimisation terms for target tracking and control inputs. Therefore, the trajectories generated by MPC are guaranteed to be feasible, and the approach is presumed to be less susceptible to deadlocks. Several studies have demonstrated dynamic obstacle avoidance using MPCs. In [8,9], collision avoidance was achieved by integrating attractive and repulsive forces defined in the APF into the constraints and models. In [10], a danger circle and minimal allowable distance were defined around an obstacle. When a UAV in a danger zone approaches the allowance distance, the cost is increased to encourage avoidance. [11,12] introduced a unique collision cost

[*] Corresponding author.
E-mail address: suzuki-s@chiba-u.jp (S. Suzuki).

https://doi.org/10.1016/j.robot.2022.104320

function as a constraint in the evaluation function by considering the relative positions of the UAV and obstacles, thereby altering the correlation of the cost and tuning parameter. Furthermore, a decentralised predictive flocking algorithm was proposed in [13].

However, these methods ignore the uncertainties prevalent in a real environment, such as the measurement and manoeuvre errors. In contrast, chance-constrained MPC (CCMPC) can account for these uncertainties using chance constraints [14]. Previously developed chance-constrained methods consider the uncertainty of position [15–17]. In addition, actual experiments based on CCMPC have been conducted using dynamic obstacle detection with on-board depth cameras [18]. Despite higher noise-resistance than deterministic collision avoidance methods, these methods ignore the uncertainty of the velocity, thereby heightening collision probability when the position and velocity data contain large amounts of noise. Moreover, UAVs are able to avoid obstacles in a wider variety of situations by accounting for both velocity and position.

The main contributions of this paper are:

- Formulation of the chance-constraints based on obstacle velocity (CCOV), which account for uncertainty in velocity.
- Combination of CCOV and positional chance constraints to account for both position and velocity uncertainty.
- Achievement of collision avoidance with high velocity obstacles, even in a noisy environment.

The rest of this paper is organised as follows: The primary problems encountered in the collision avoidance situation are stated in Section 2. In Section 3, the concept of velocity obstacles (VO), which are the basis of the CCOV method is briefly explained. Following that, the constraints used in the conventional CCMPC are defined, and the problems are discussed using the deterministic MPC in Section 4. Finally, the numerical simulation results of the proposed method are presented in Section 4, followed by the concluding statements in Section 5.

2. Problem statement

2.1. Preliminaries

The following conditions were applied in this study:

- The UAV can determine the position and velocity of all obstacles and UAVs, including itself. It must be noted that values obtained may contain noise, depending on the conditions.
- The UAVs and the obstacles are treated as spheres of radius r_i and r_o, respectively.
- The obstacles move in constant velocity linear motion.
- In the prediction interval of MPC, the avoidance targets are assumed to move in constant velocity linear motion from the positions predetermined at the start of the prediction.
- In a noisy environment, the position and velocity data of the obstacle and the UAV contain Gaussian noise with mean zero and standard deviation σ, which is a set parameter.

Here, the avoidance target refers to the obstacles and other UAVs that should be avoided.

The CCOV idea is based on the concept of VO [19]; the VO method can avoid fast moving obstacles because it uses the relative velocity of the obstacles for collision avoidance. The CCOV method constrains the probability of being judged as a collision with a VO to below a threshold. However, the approach ignores the location of obstacles, e.g., what is a safe distance between the UAV and an obstacle? Therefore, the combination of CCOV and chance constraint of position generates a robust and flexible trajectory.

2.2. UAV model

In this paper, vectors are denoted in bold ($\boldsymbol{x}$), matrices in capital (M), and sets in script ($\mathscr{A}$). An accented $\hat{x}$ denotes the mean of a random variable x. $Pr(\cdot)$ indicates the probability of an event, and a superscript denotes transposition (x^T).

The number of UAVs is n, and each UAV is denoted by a subscript $i \in \mathscr{I} = \{1, 2, \dots, n\} \subset \mathbb{N}$. The dynamics of UAV $i \in \mathscr{I}$ are described as follows:

$$\boldsymbol{x}_i^{k+1} = A\boldsymbol{x}_i^k + B\boldsymbol{u}_i^k + \boldsymbol{\omega}_i^k, \quad \boldsymbol{x}_i^0 \sim \mathscr{N}(\hat{\boldsymbol{x}}_i^0, \Gamma_i^0) \tag{1}$$

where $\boldsymbol{x}_i^k \in \mathbb{R}^{n_x}$ is the state vector at time k, and $\boldsymbol{u}_i^k \in \mathbb{R}^{n_u}$ is the control input. The initial state $\boldsymbol{x}_i^0$ is denoted by a Gaussian function: a random variable with mean $\hat{\boldsymbol{x}}_i^0$ and covariance matrix Γ_i^0. $\boldsymbol{\omega}_i^k$ denotes process noise, which is $\boldsymbol{\omega}_i^k \sim \mathscr{N}(0, Q_i^k)$. $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ are the linear dynamics, which are independent of each axis and represented as:

$$\begin{bmatrix} x_i^{k+1} \\ v_{ixs}^{k+1} \\ v_{ix}^{k+1} \\ a_{ix}^{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 - \frac{\Delta t}{T_d} & \frac{\Delta t}{T_d} & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & a_{1x}\Delta t & 1 + a_{2x}\Delta t \end{bmatrix} \begin{bmatrix} x_i^k \\ v_{ixs}^k \\ v_{ix}^k \\ a_{ix}^k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_{1x}\Delta t \end{bmatrix} \theta_{iref}^k \tag{2}$$

$$\begin{bmatrix} y_i^{k+1} \\ v_{iys}^{k+1} \\ v_{iy}^{k+1} \\ a_{iy}^{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 - \frac{\Delta t}{T_d} & \frac{\Delta t}{T_d} & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & a_{1y}\Delta t & 1 + a_{2y}\Delta t \end{bmatrix} \begin{bmatrix} y_i^k \\ v_{iys}^k \\ v_{iy}^k \\ a_{iy}^k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_{1y}\Delta t \end{bmatrix} \phi_{iref}^k \tag{3}$$

$$\begin{bmatrix} z_i^{k+1} \\ v_{iz}^{k+1} \\ a_{iz}^{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & a_{1z}\Delta t & 1 + a_{2z}\Delta t \end{bmatrix} \begin{bmatrix} z_i^k \\ v_{iz}^k \\ a_{iz}^k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b_{1z}\Delta t \end{bmatrix} u_{ith}^k \tag{4}$$

where $\boldsymbol{p}_i^k = [x_i^k \ y_i^k \ z_i^k]^T \in \mathbb{R}^3$, $\boldsymbol{v}_i^k = [v_{ix}^k \ v_{iy}^k \ v_{iz}^k]^T \in \mathbb{R}^3$, and $\boldsymbol{a}_i^k = [a_{ix}^k \, a_{iy}^k \, a_{iz}^k]^T \in \mathbb{R}^3$ are the position, velocity, and acceleration, respectively. Δt represents the sampling time, which in this study is set to 0.01 s. The sensor velocity $\boldsymbol{v}_{is}^k = [v_{ixs}^k \, v_{iys}^k]^T \in \mathbb{R}^2$ considers the delay time of the x and y axes acquired by the sensor. T_d is 0.04 s for the optical motion capture system, and 0.3 s for the global navigation satellite system (GNSS). $[\phi_{iref}^k \ \theta_{iref}^k \ u_{ith}^k]^T \in \mathbb{R}^3$ are the attitude command values for the roll and pitch angle and throttle input. The model parameters are listed in Table 1. An IBIS UAV from Liberaware is used as the control target, as shown in Fig. 1. In this study, the UAV is assumed to be a sphere of radius $r_i = 0.1$ m.

2.3. Obstacle model

The number of obstacles is n_o, and each obstacle is denoted as $o \in \mathscr{I}_o = \{1, 2, \dots, n_o\} \subset \mathbb{N}$. The position of the obstacle at time k is defined as the 3D space of $\boldsymbol{p}_o^k = [x_o^k, y_o^k, z_o^k]^T \in \mathbb{R}^3$,

Fig. 1. Overview of the IBIS.

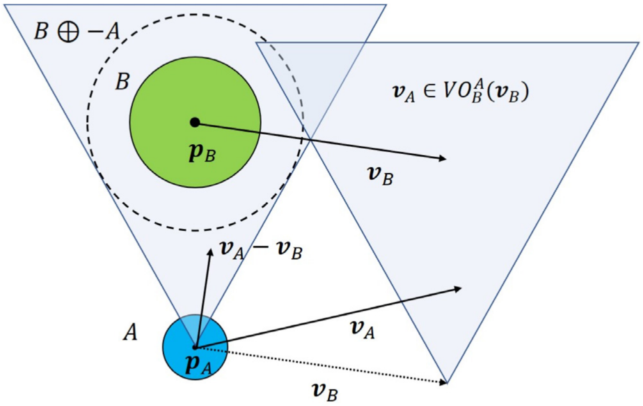


Fig. 2. Velocity obstacle $VO_B^A(\boldsymbol{v}_B)$ from an obstacle B to an agent A.

Table 1
Model parameters.

Parameter	Value	Parameter	Value	Parameter	Value
a_{1x}	−1.4	a_{2x}	−7.2	b_{1x}	0.12
a_{1y}	−1.4	a_{2y}	−7.2	b_{1y}	0.12
a_{1x}	−9.6	a_{2x}	−13.3	b_{1x}	−0.23

and the velocity is $\boldsymbol{v}_o^k = [v_{ox}^k, v_{oy}^k, v_{oz}^k]^T \in \mathbb{R}^3$. The obstacle moves in simulation space with a constant velocity linear motion as follows:

$$\begin{bmatrix} \boldsymbol{p}_o^{k+1} \\ \boldsymbol{v}_o^{k+1} \end{bmatrix} = \begin{bmatrix} I_3 & \Delta t I_3 \\ O & I_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_o^k \\ \boldsymbol{v}_o^k \end{bmatrix} \tag{5}$$

Because the obstacle model of radius r_i is also applied to other UAVs that should be avoided, they are predicted to be in constant linear motion at the measured velocity from the measured position.

2.4. Velocity obstacles

In this approach, the velocity space is defined as a plane, as shown in Fig. 2. Agent A is defined by the reference point $\boldsymbol{p}_A$ in the 2D plane, and the obstacle B is defined by the reference point $\boldsymbol{p}_B$. Agent A avoids collisions by selecting a velocity outside this space. To generalise, let $A \oplus B$ be the Minkowski sum of agent A and obstacle B, and let $-A = \{-\boldsymbol{a} \mid \boldsymbol{a} \in A\}$ denote agent A reflected at its reference point. Then:

$$A \oplus B = \{\boldsymbol{a} + \boldsymbol{b} \mid \boldsymbol{a} \in A, \boldsymbol{b} \in B\}, \quad -A = \{-\boldsymbol{a} \mid \boldsymbol{a} \in A\}. \tag{6}$$

Let $\lambda(\boldsymbol{p}, \boldsymbol{v})$ be the half-line in the direction of velocity $\boldsymbol{v}$ starting at position $\boldsymbol{p}$.

$$\lambda(\boldsymbol{p}, \boldsymbol{v}) = \{\boldsymbol{p} + t\boldsymbol{v} \mid t \geq 0\} \tag{7}$$

Hence, VO of B to A is defined as follows:

$$VO_B^A(\boldsymbol{v}_B) = \{\boldsymbol{v}_A \mid \lambda(\boldsymbol{p}_A, \boldsymbol{v}_A - \boldsymbol{v}_B) \cap B \oplus -A \neq 0\} \tag{8}$$

where $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ are the velocities of agent A and obstacle B, respectively.

$\boldsymbol{v}_A \in VO_B^A(\boldsymbol{v}_B)$ indicates that agent A contains all the velocities at which it will eventually collide. Reciprocal velocity obstacles (RVOs) can be used to resolve the oscillation problem that occurs in multi-agent navigation and enable collision avoidance between agents [20]. A hybrid reciprocal velocity obstacle (HRVO) combines the velocity space defined by RVO and VO, allowing for more efficient collision avoidance [21]. However, these approaches to collision-free path planning are computationally complex [22]. Although optimal reciprocal collision avoidance (ORCA) rapidly derives sufficient conditions to avoid collisions in a finite-time future [23], uncertainty is ignored.

3. Chance-constrained MPC

MPC obtains the optimal control input by sequentially solving an optimal control problem that minimises the evaluation function up to a finite-time future. The CCMPC is characterised by the use of chance constraints in the function. In the first part of this section, the constraints of the CCMPC proposed in previous studies are described, followed by a discussion of its advantages and disadvantages [15,24]. In the latter part, the formulation of CCOV and its implementation in CCMPC is presented.

3.1. Cost function for collision avoidance

MPC minimises the evaluation function presented as follows:

$$J = \varphi(\bar{\boldsymbol{x}}(t+T)) + \int_t^{t+T} L(\bar{\boldsymbol{x}}(t+\tau), \bar{\boldsymbol{u}}(t+\tau))d\tau \tag{9}$$

where T and τ denote the prediction time and the time variable from the time t when the prediction starts, respectively. $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$ denote the state and input within the evaluation interval predicted by the model, respectively. The first and second terms are referred to as terminal and stage costs, respectively, and are defined in this study as follows:

$$\varphi(\bar{\boldsymbol{x}}(t+T)) = \bar{\boldsymbol{x}}^T(t+T) S \bar{\boldsymbol{x}}(t+T) \tag{10}$$

$$\begin{aligned} L(\bar{\boldsymbol{x}}(t+\tau), \bar{\boldsymbol{u}}(t+\tau)) &= \bar{\boldsymbol{x}}^T(t+\tau) Q \bar{\boldsymbol{x}}(t+\tau) \\ &+ \bar{\boldsymbol{u}}^T(t+\tau) R \bar{\boldsymbol{u}}(t+\tau) + P(\bar{\boldsymbol{x}}(t+\tau)) + B(\bar{\boldsymbol{x}}(t+\tau)) \end{aligned} \tag{11}$$

where $P(\cdot)$ and $B(\cdot)$ denote the penalty and barrier functions, respectively, and S, Q, and R are the weight matrices. Using the costs in Eqs. (10) and (11), the states of the UAVs are guaranteed to converge to the origin. Thus, by setting the target position of each UAV at the origin, it is possible to use these costs to guide the UAV to the target position. In particular, in the case of formation flight, the origin is always set to the target point where the UAV should converge within the formation. The penalty function gradually increases the cost after a predetermined threshold is violated. In contrast, the barrier function steeply increases the cost just before the threshold is violated. Because the purpose of the evaluation function is to minimise the cost, collision avoidance can be considered as a scenario with the least cost.

3.2. Chance constraints with barrier function

The chance constraints that consider the uncertainty of the positions between the UAV and obstacle are briefly summarised in this section [15]. Hereafter, these constraints are referred to as position-chance constraints. In these constraints, a sphere, which is the sum of the UAV radius and the three axes of the avoidance target, is defined in the 3D space centred on the target. Next, a half-space perpendicular to the position vector is defined on the surface of the sphere, and the probability of entering this space is constrained to a value less than a certain threshold to enable collision avoidance. The constraints are expressed as follows:

$$\begin{aligned} g^{c}_{pih}(\boldsymbol{x}) &= \boldsymbol{a}^{T}_{ih}(\hat{\boldsymbol{p}}_i - \hat{\boldsymbol{p}}_h) - b_{ih} \\ &\quad - \operatorname{erf}^{-1}(1-2\delta_{ih})\sqrt{2\boldsymbol{a}^{T}_{ih}(\Sigma_i + \Sigma_h)\boldsymbol{a}_{ih}} \\ &\quad\ \forall h \in \mathscr{I} \cup \mathscr{I}_o, \quad h \neq i \end{aligned} \tag{12}$$

where $\boldsymbol{a}_{ih} = (\hat{\boldsymbol{p}}_i - \hat{\boldsymbol{p}}_h)/\|\hat{\boldsymbol{p}}_i - \hat{\boldsymbol{p}}_h\|$, $b_{ih} = r_i + r_h$, and δ_{ih} is the probability threshold. $\operatorname{erf}(\cdot)$ is the standard error function, defined as $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$. The constraints are introduced into the evaluation function as the barrier function.

$$\begin{aligned} B^{p}(\boldsymbol{x}) &= \frac{w_B}{g^{c}_{pih}(\boldsymbol{x})^2} \\ \forall h &\in \mathscr{I} \cup \mathscr{I}_o, \quad h \neq i \end{aligned} \tag{13}$$

where w_B is a coefficient that denotes the weight.

3.3. Deterministic constraints with penalty function

In the deterministic constraints, the UAV radius and the preset no-entry area are added to the three axes of the avoidance target, and an affine transformation is performed [24]. The UAV centre coordinates are restricted from entering the area, thus avoiding collisions. In this study, the constraints are referred to as deterministic position constraints and are expressed as follows:

$$\begin{aligned} g_{pih}(\boldsymbol{x}) &= 1 - \|\hat{\boldsymbol{p}}_i - \hat{\boldsymbol{p}}_h\|^{T}\,\Omega^{safe}_{ih}\,\|\hat{\boldsymbol{p}}_i - \hat{\boldsymbol{p}}_h\| \\ \forall h &\in \mathscr{I} \cup \mathscr{I}_o, \quad h \neq i \end{aligned} \tag{14}$$

where $\Omega^{safe}_{ih} = diag(1/(r_i + r_h + d^{a}_{ih})^2, 1/(r_i + r_h + d^{b}_{ih})^2, 1/(r_i + r_h + d^{c}_{ih})^2)$, and d^{a}_{ih}, d^{b}_{ih}, and d^{c}_{ih} denote the no-entry area defined in each axis direction. The constraints are introduced as a penalty function.

$$\begin{aligned} P^{p}(\boldsymbol{x}) &= \begin{cases} w_P \cdot g_{pih}(\boldsymbol{x}), & if\ g_{pih}(\boldsymbol{x}) \geq 0 \\ 0, & (otherwise) \end{cases} \\ \forall h &\in \mathscr{I} \cup \mathscr{I}_o, \quad h \neq i \end{aligned} \tag{15}$$

where w_P denotes the weight. The constraints by position are activated from the moment the UAV enters the area, and the cost increases with its proximity to the obstacle.

3.4. Advantages and disadvantages of the CCMPC considering position

The deterministic MPC using a no-entry area is a simple collision-avoidance method [11]. The control target can be distanced from obstacles based on the observed information. However, deterministic MPC is susceptible to oscillations and collisions due to the mutual interference of areas in a noisy environment. The numerical simulation results are shown in Fig. 3, where the blue spheres represent obstacles, and the red, green, and purple spheres represent the UAVs. The simulation was performed in MATLAB and contained Gaussian noise. The mean and two standard deviations are 0, 0.25 m, and 0.25 m/s, respectively. The three UAVs move from (3.0, 3.0, 0) m to (−3.0, −3.0, 0) m with the lead UAV as the centre coordinate of the formation, maintaining the triangle as much as possible. The two obstacles are 0.8 m and 1.3 m in diameter and move in a constant velocity linear motion with initial velocities of (0, 0.5, 0) m/s and (0.1, 0.3, 0) m/s. As shown in Fig. 3(a), the green UAV avoided the smaller obstacle, whereas the purple UAV collided with the larger obstacle, as shown in Fig. 3(b). This collision was caused by the approach of the green UAV, which avoided the obstacle and pushed the purple UAV out of the way. In a noisy environment, deterministic constraints may allow collision avoidance in simple situations, but can be ineffective in complex ones.

CCMPC with position chance constraints provides smoother motion planning than MPC in a noisy environment [15]. Fig. 4 shows the collision avoidance of CCMPC under identical conditions. As shown in Fig. 4(a), the green UAV avoided the smaller obstacle, as in the case of the deterministic MPC. In Fig. 4(b), the purple UAV avoided the outside of the obstacle in response to the avoidance of the green UAV, whereas the UAV with the deterministic MPC collided with the obstacle. This result indicates that chance constraints can be used to achieve greater flexibility in collision avoidance in a noisy environment requiring complex collision avoidance. However, because neither MPC considers the velocity, it is difficult to avoid obstacles with high velocity, regardless of the noise. Fig. 5 shows the simulation in which the UAV is required to avoid several obstacles with high velocity. The chance constraints were used for collision avoidance in the absence of noise. The UAV moved from the initial position (3.0, 0, 0) m to the target position (−3.0, 0, 0) m. The obstacles were all 1.2 m in diameter and had an initial velocity of 3.0 m/s in the direction of the x-axis. The initial positions were (−2.0, 0.6, 0) m, (−5.0, −1.0, 0) m, and (−5.0, −2.5, 0) m. As shown in Figs. 5(b) and 5(c), the UAV kept receding from the target position and failed to avoid two adjacent obstacles. This was because the risk of entering the area of the approaching obstacles was erroneously estimated as low. This problem can be addressed by considering obstacle velocity. CCOV is introduced into the conventional CCMPC to compensate for both the position and velocity uncertainties.

3.5. Chance-constraints based on obstacle velocity

CCOV provides the optimal control input that can achieve safe motion in a noisy environment. First, let ${}^{k}VO^{i}_{o}(\boldsymbol{v}_o)$ be the space defined by the VO of UAV i and obstacle o at time k, and the chance constraints between UAV i and obstacle o are expressed as follows:

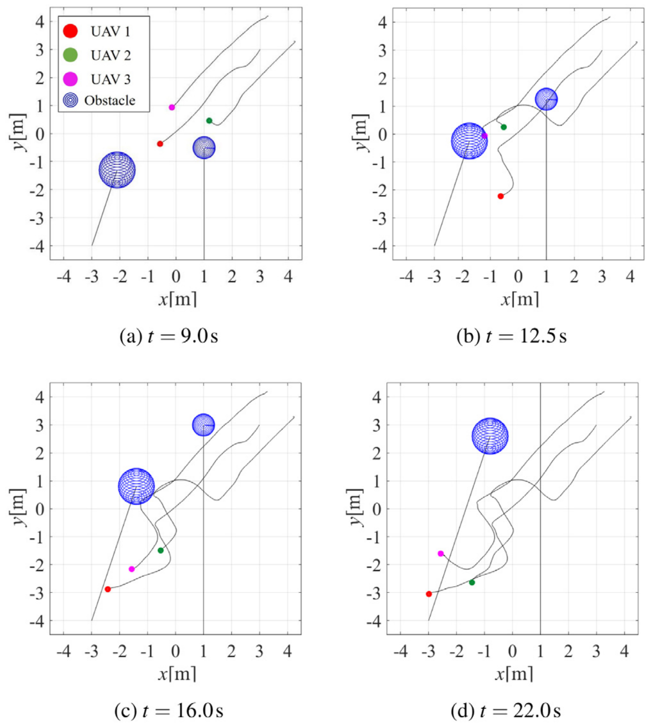
$$Pr({}^{k}\boldsymbol{v}_i \notin {}^{k}\,VO^{i}_{o}(\boldsymbol{v}_o)) \geq 1 - \delta_o, \qquad \forall o \in \mathscr{I}_o \tag{16}$$

where δ_o denotes the CCOV threshold. Similarly, the constraints between UAVs are expressed as follows:

$$Pr({}^{k}\boldsymbol{v}_i \notin {}^{k}\,VO^{i}_{j}(\boldsymbol{v}_j)) \geq 1 - \delta_r, \qquad \forall j \in \mathscr{I}, j \neq i \tag{17}$$

δ_r is the threshold between UAVs. In CCOV, the UAVs consider collision avoidance in the same plane.

This constraint restricts the probability of the velocity of a UAV entering the velocity space below the threshold. Between the UAV and an obstacle, a perpendicular line is first defined from the centre coordinate O of the obstacle to the half-line of ${}^{k}VO^{i}_{o}(\boldsymbol{v}_o)$, as shown in Fig. 6. On the extension of the line, an intersection point P with a half-line extending in the direction of relative velocity ${}^{k}\boldsymbol{v}_i - {}^{k}\,\boldsymbol{v}_o$ is determined, and thus collision avoidance is encouraged by suppressing the probability of that point entering the velocity space. For the example in Fig. 6, the velocity of the UAV is considered to be evaluated to ensure the extreme right



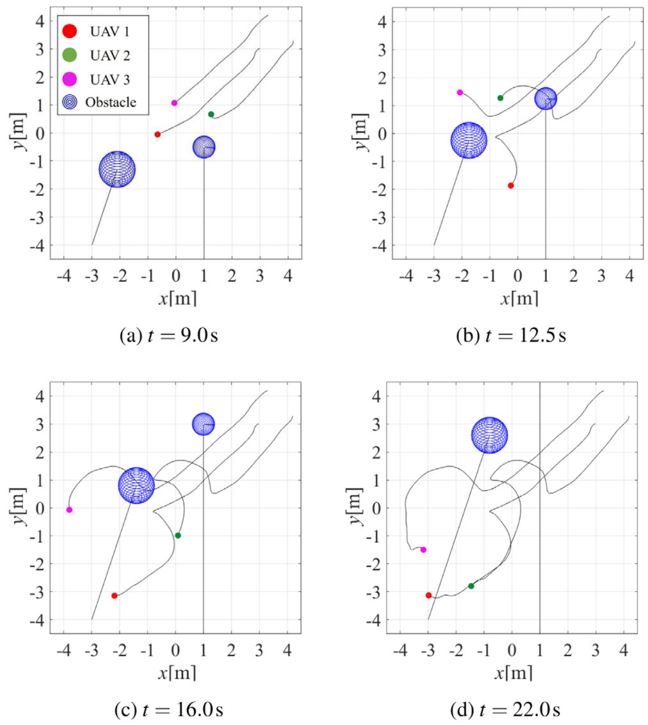
(a) $t = 9.0$ s

(b) $t = 12.5$ s

(c) $t = 16.0$ s

(d) $t = 22.0$ s

Fig. 3. Collision avoidance with Gaussian noise using deterministic MPC (standard deviation: 0.25 m, 0.25 m/s). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



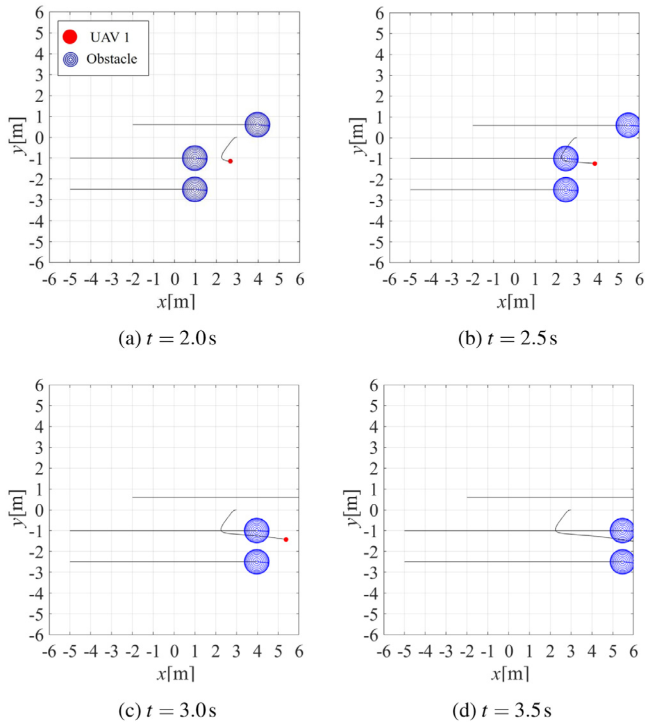
(a) $t = 9.0$ s

(b) $t = 12.5$ s

(c) $t = 16.0$ s

(d) $t = 22.0$ s

Fig. 4. Collision avoidance with Gaussian noise using CCMPC (standard deviation: 0.25 m, 0.25 m/s). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) $t = 2.0$ s

(b) $t = 2.5$ s

(c) $t = 3.0$ s

(d) $t = 3.5$ s

Fig. 5. Collision avoidance using CCMPC.

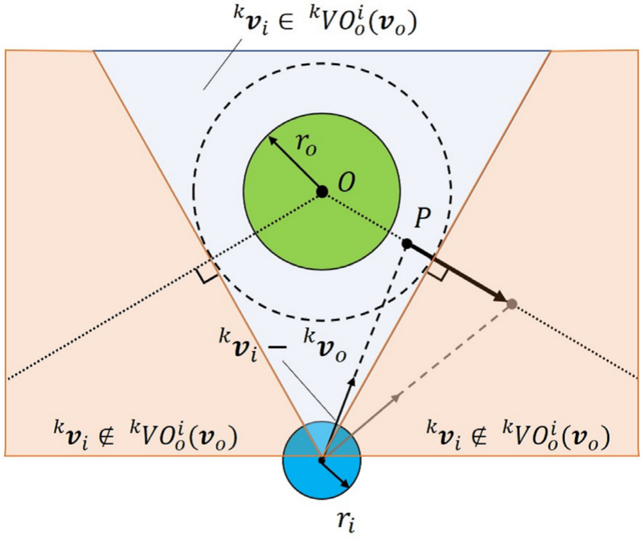

${}^k\boldsymbol{v}_i \in {}^kVO^i_o(\boldsymbol{v}_o)$
r_o
O
P
${}^k\boldsymbol{v}_i - {}^k\boldsymbol{v}_o$
${}^k\boldsymbol{v}_i \notin {}^kVO^i_o(\boldsymbol{v}_o)$
${}^k\boldsymbol{v}_i \notin {}^kVO^i_o(\boldsymbol{v}_o)$
r_i

Fig. 6. Chance-constraints based on obstacle velocity.

outer point is chosen. Let $\boldsymbol{r}_{io}$ be the vector from intersection P to the centre coordinate O. This process is expressed as follows [25]:

$$\begin{aligned} g^c_{vio}(\boldsymbol{x}) &= \boldsymbol{a}^T_{io}\,\hat{\boldsymbol{r}}_{io} - b_{io} \\ &\quad - \operatorname{erf}^{-1}(1 - 2\delta_o)\sqrt{2\boldsymbol{a}^T_{io}(\Sigma_i + \Sigma_o)\boldsymbol{a}_{io}} \end{aligned} \tag{18}$$

where $\boldsymbol{a}_{io} = \hat{\boldsymbol{r}}_{io}/\|\hat{\boldsymbol{r}}_{io}\|$, $b_{io} = r_i + r_o$, and Σ_i and Σ_o are the error covariance matrices of UAV i and obstacle o, respectively. The constraints between UAVs can be expressed as follows:

$$\begin{aligned} g^c_{vij}(\boldsymbol{x}) &= \boldsymbol{a}^T_{ij}\,\hat{\boldsymbol{r}}_{ij} - b_{ij} \\ &\quad - \operatorname{erf}^{-1}(1 - 2\delta_r)\sqrt{2\boldsymbol{a}^T_{ij}(\Sigma_i + \Sigma_j)\boldsymbol{a}_{ij}} \end{aligned} \tag{19}$$

where $\boldsymbol{a}_{ij} = \hat{\boldsymbol{r}}_{ij}/\|\hat{\boldsymbol{r}}_{ij}\|$ and $b_{ij} = 2r_i$. They are introduced into the evaluation function as the penalty functions as follows:

$$P^v_{io}(\boldsymbol{x}) = \begin{cases} w_{Po} \cdot g^c_{vio}(\boldsymbol{x}), & if\ g^c_{vio}(\boldsymbol{x}) \geq 0 \\ 0, & (otherwise) \end{cases} \tag{20}$$

$$P_{ij}^{v}(\boldsymbol{x}) = \begin{cases} w_{Pj} \cdot g_{vij}^{c}(\boldsymbol{x}), & if\ g_{vij}^{c}(\boldsymbol{x}) \geq 0 \\ 0, & (otherwise) \end{cases} \tag{21}$$

$$P^{v}(\boldsymbol{x}) = P_{io}^{v}(\boldsymbol{x}) + P_{ij}^{v}(\boldsymbol{x}) \tag{22}$$
$$\forall o \in \mathscr{I}_o, \quad \forall j \in \mathscr{I}, j \neq i$$

where w_{Po} and w_{Pj} are coefficients that represent the weights. The collision cost functions are expressed in the form of constraints described in Sections 3.2 and 3.3 with the incorporation of CCOV.

$$P(\boldsymbol{x}) = P^{p}(\boldsymbol{x}) + P^{v}(\boldsymbol{x}) \tag{23}$$

$$B(\boldsymbol{x}) = B^{p}(\boldsymbol{x}) \tag{24}$$

When using CCOV, it is necessary to measure the velocity of the target to be avoided. However, if the target is another UAV, the velocity can be easily obtained through wireless communication. Moreover, even if the target is a dynamic obstacle, the speed can be measured using a depth camera [18] or millimetre wave radar. Therefore, there is no need to equip UAVs with additional equipment in order to implement CCOV.

4. Numerical simulation

In this section, the simulation results of the proposed method are presented and compared with the results of conventional CCMPC and deterministic MPC. The control period was set to $T_s = 0.01$ s and C/GMRES was used for the MPC solver [26].

Moreover, it is assumed that UAVs in the environment can exchange absolute positions and velocities with each other and measure the positions and velocities of obstacles. The former can be easily achieved by transmitting the position and velocity of one UAV acquired by GNSS to another UAV via wireless communication, and the latter can be achieved by measuring obstacles using radar or cameras. However, these sensor data, especially the position and velocity of obstacles acquired by cameras, are expected to contain large errors. Therefore, in the latter part of the simulation, large errors are taken into account in the position and velocity data of UAVs and obstacles, and the robustness of the system is verified.

4.1. Comparison methods

In conventional CCMPC, the two constraints described in Sections 3.2 and 3.3 are introduced into the evaluation function. The penalty function and barrier functions are expressed as follows:

$$B(\boldsymbol{x}) = \frac{w_{Bc}}{g_{pih}^{c}(\boldsymbol{x})^2} \tag{25}$$

$$P(\boldsymbol{x}) = \begin{cases} w_{Pc} \cdot g_{pih}(\boldsymbol{x}), & if\ g_{pih}(\boldsymbol{x}) \geq 0 \\ 0, & (otherwise) \end{cases} \tag{26}$$
$$\forall h \in \mathscr{I} \cup \mathscr{I}_o, \quad h \neq i$$

$g_{pih}^{c}(\boldsymbol{x})$ and $g_{pih}(\boldsymbol{x})$ are constraints expressed in Eqs. (12) and (14). w_{Bc} and w_{Pc} are the weights.

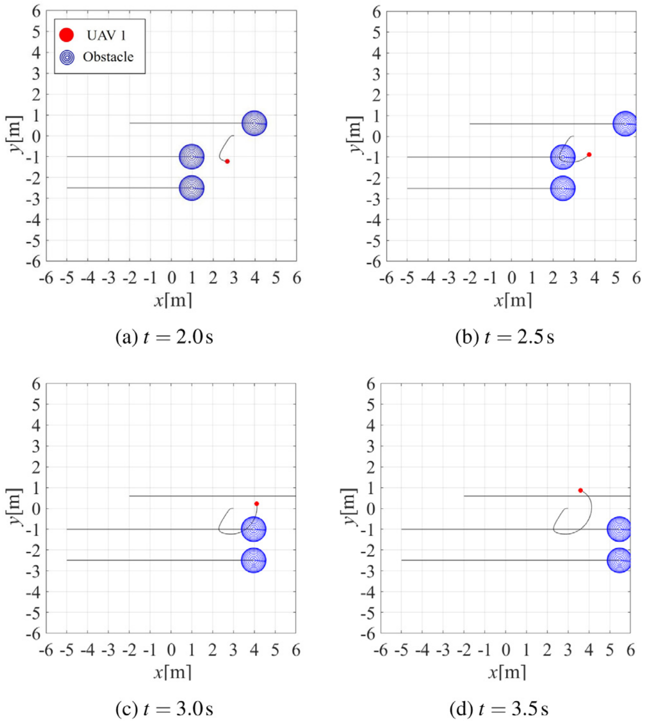
The following constraints are used for deterministic MPC, which ignores the uncertainty.

$$g_{ih}^{d} = d_{safe}^{2} - (\boldsymbol{p}_i - \boldsymbol{p}_h)^2 \tag{27}$$
$$\forall h \in \mathscr{I} \cup \mathscr{I}_o, \quad h \neq i$$

The constraints are introduced as the barrier function; thus, the cost rises steeply when UAV i attempts to enter the no-entry area d_{safe}. Nominal value of simulation parameters are listed in Table 2. These values will be used in following simulations.

Table 2
Simulation parameters.

Description	Symbol	Value
Prediction time	T	1.0 s
Collision probability threshold	$\delta_{\bullet}$, $\bullet = r, o, ih$	0.003
Weight for CC	w_{Bc}, w_B	0.5
Weight for CCVO for obstacle	w_{Po}	5×10^5
Weight for CCVO for other UAVs	w_{Pj}	5×10^3
Weight for deterministic constraints	w_P, w_{Pc}	6×10^6



(a) $t = 2.0$ s

(b) $t = 2.5$ s

(c) $t = 3.0$ s

(d) $t = 3.5$ s

Fig. 7. Collision avoidance using CCMPC considering obstacle velocity.

4.2. Collision avoidance of high-velocity obstacles

The proposed method was performed under conditions identical to those in the simulation, as shown in Fig. 5 and compared to conventional CCMPC. The CCOV results are shown in Fig. 7. As shown in Fig. 7(a), the UAV moved along the negative y-axis to avoid the leading obstacle, as in the case of the conventional method. However, as shown in Fig. 7(b), the UAV moved along the positive y-axis immediately and did not approach the riskier area where the two obstacles approached. Subsequently, the UAV was able to correctly head to the target position, as shown in Figs. 7(c) and 7(d). Fig. 8 shows the attitude commands for the roll and pitch angles. For the period from 2.0 to 2.5 s, large control inputs approaching 50 degrees can be observed. However, these were instantaneous corrections implemented to avoid a critical collision. Hence, this behaviour is expected to be feasible in a real environment. By considering not only the relative position but also the velocity, more flexibility can be achieved in collision avoidance. Furthermore, it is noteworthy that the UAV did not exhibit movement along the z-axis.

4.3. Formation flight in a noisy environment

For this scenario, a situation identical to that in the simulations shown in Figs. 3 and 4 was set up. The standard deviation of Gaussian noise for positions and velocities were set to 0.45 m

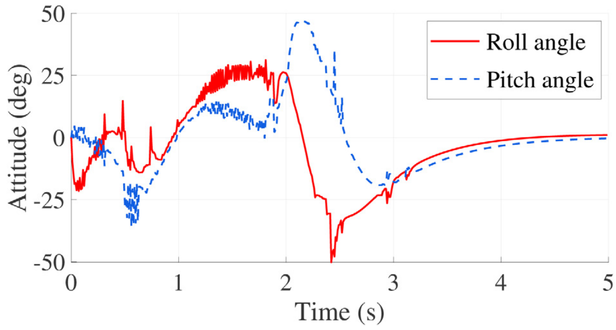


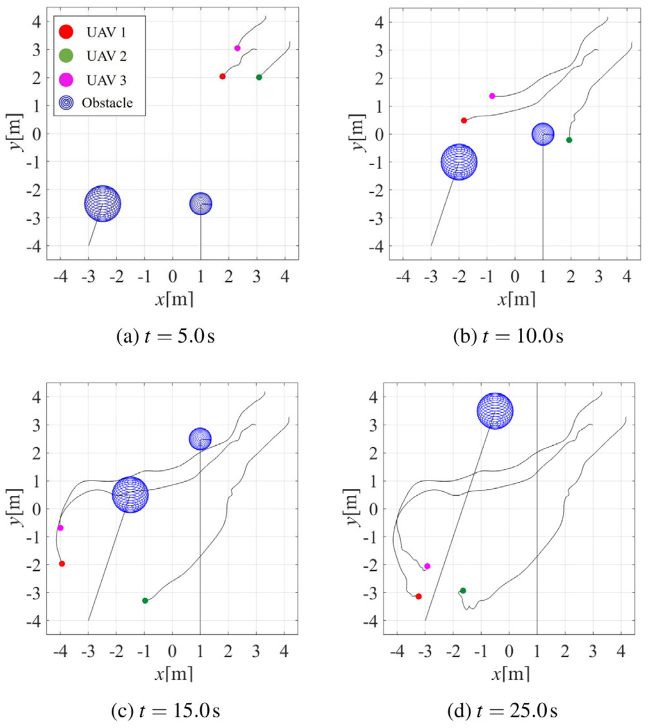
Fig. 8. Roll and pitch command value.

and 0.45 m/s, respectively. A single point positioning GNSS is used for the UAV's localisation in the outdoor environment, with a position error of several tens of centimetres, and a velocity error of approximately 0.2 to 0.3 m/s. In addition, the error in measuring the position of obstacles using cameras and radar is considered to be within 10% of the distance from the obstacle, which is several tens of centimetres considering the size of the UAV assumed in this study. Therefore, the measurement error set in this simulation is sufficiently large compared to the assumed sensor error, and is considered sufficient for robustness verification. The simulation results of the proposed method are shown in Fig. 9, and those of the conventional CCMPC are shown in Fig. 10. The three UAVs with CCOV split into a Y-shape and avoided the obstacles, as shown in Fig. 9(b). In contrast, the green UAV using conventional CCMPC collided with the obstacle at $t = 9.5$ s, as shown in Fig. 10(b). Furthermore, the purple UAV collided with the obstacle at $t = 15.0$ s, as shown in Fig. 10(c). In a noisier environment, merely the position chance constraints are insufficient to ensure a safe distance from dynamic obstacles. By introducing CCOV, early collision avoidance can be achieved even in a noisy environment.

4.4. Study of simulation parameters and comparison of computational load

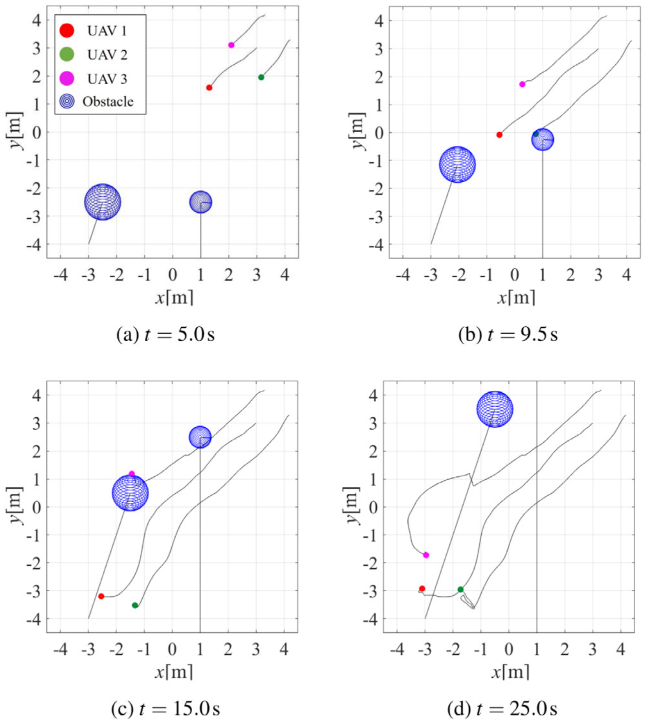
In this section, the various simulation parameters are examined and present comparative results of the computational load. First, multiple simulations of formation flight identical to those in Section 4.3 were performed, and collision rate was evaluated for each value of the simulation parameters. The collision rate is the percentage of collisions that occurred in multiple simulations. It is defined as 100% when all three UAVs collided with obstacles or other UAVs in all simulations and 0% when no collisions occurred. Here, five simulations were performed for one parameter value. The collision rates for all simulation patterns are listed in Table 3. The results are compared for the proposed method (PM), conventional CCMPC (CCMPC), and deterministic MPC (DMPC).

The first row of Table 3 shows the results when the noise gain is varied. The row labelled "Variable" displays the results when the noise gain varies with time, as shown in Fig. 11. In all cases, including the case where the noise gain is varied, the collision rate of the proposed method is the lowest, indicating the effectiveness of the proposed method in noisy environments. The second and subsequent rows of Table 3 show the results when the simulation parameters are changed. Here, the noise gains for the simulations after the second row are set to 0.5 m and 0.5 m/s. The simulations are performed in a fairly noisy environment. This table shows that for all parameters, the proposed method performs best for the nominal values shown in Table 2 in the previous section. This shows the validity of the nominal values set.



(a) $t = 5.0$ s (b) $t = 10.0$ s (c) $t = 15.0$ s (d) $t = 25.0$ s

Fig. 9. Collision avoidance with Gaussian noise using CCMPC considering obstacle velocity (standard deviation: 0.45 m, 0.45 m/s). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) $t = 5.0$ s (b) $t = 9.5$ s (c) $t = 15.0$ s (d) $t = 25.0$ s

Fig. 10. Collision avoidance with Gaussian noise using CCMPC (standard deviation: 0.45 m, 0.45 m/s). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Finally, the computation time for each method were compared. The comparison results are shown in Table 4. In this comparison, the average computation time required for one step

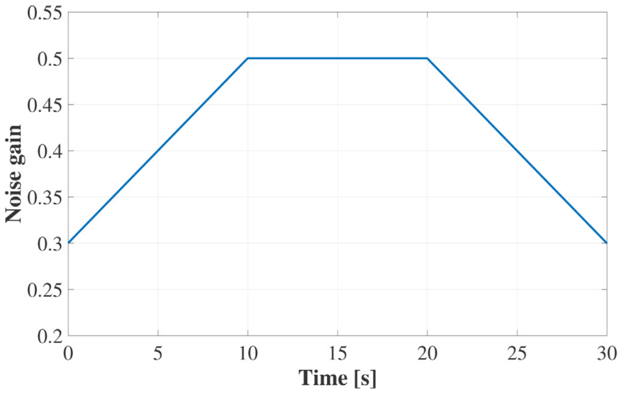


Fig. 11. Profile of noise gain in simulation.

Table 3
Comparison of collision rate for all simulation patterns.

Parameters	Value	PM	CCMPC	DMPC
Noisegain	0.35	**0.0**	20.0	26.7
	0.45	**0.0**	26.7	66.7
	0.5	**33.3**	46.7	53.3
	Variable	**6.7**	60.0	60.0
T	0.5	46.7	60.0	**40.0**
	1.0	**33.3**	53.3	60.0
	2.0	46.7	**40.0**	60.0
w_{Bc}, w_B	0.25	**13.3**	53.3	–
	0.5	**33.3**	53.3	–
	1.0	**13.3**	53.3	–
$\delta_\bullet$	0.003	**33.3**	53.3	–
	0.03	**26.7**	53.3	–
	0.3	**46.7**	**46.7**	–
w_{Po}	5×10^3	66.7	–	–
	5×10^5	**33.3**	–	–
	5×10^7	60.0	–	–

Table 4
Comparison of computational load.

Parameters	Value	PM	CCMPC	DMPC
T	0.5	4.46	4.36	1.0
	1.0	5.82	5.82	1.0
	2.0	7.36	7.40	1.0

of control calculation for each method was measured and expressed as a factor based on the computation time for deterministic MPC. The results are also shown for each prediction time, since it is expected that the computation time increases as the prediction time increases. The table shows that the computation time of CCMPC is several times longer than that of deterministic MPC, indicating that it is computationally more demanding. This tendency becomes more pronounced as the prediction time increases. However, when comparing the conventional CCMPC with the proposed method, there is almost no change in the magnification compared to deterministic MPC. It was clear that adding the probabilistic constraints considering obstacle velocity does not increase the computation time. Furthermore, we measured the computation time of one step of control calculation for the deterministic MPC implementation used in the flight experiments in the previous study [11]. We found that the average computation time was about 0.974 ms. This result indicates that the computation time is sufficiently fast compared to the set control period of 0.01 s, even when the largest prediction time is selected.

The above simulations demonstrate the effectiveness of the proposed method. In this study, collision avoidance was performed only on a 2D plane, but it can be easily extended to 3D collision avoidance because the shapes of UAVs and obstacles are assumed to be 3D spheres. However, in the case of rotary-wing UAVs, such as multirotor helicopters, avoidance in the vertical direction causes instability because the downwash of the upper UAV affects the UAV below it. Therefore, when performing 3D collision avoidance, it is necessary to set the virtual airframe shape to an oblate ellipsoid with its long axis in the vertical direction.

5. Conclusion

The main contribution of this study is the formulation of a CCOV that considers the uncertainty of velocity. Because it was combined with position-chance constraints, robust collision avoidance has been achieved for obstacles with high velocities in a noisy environment. The main limitation of this study is that it assumes that the computation of the optimisation will be completed within the control period.

In future work, the effects of increasing obstacles and constraints on the computation load need to be considered. The proposed method will be extended to 3D collision avoidance, and its effectiveness will be demonstrated through actual flight experiments.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

[1] D. Floreano, R. Wood, Science, technology and the future of small autonomous drones, Nature 521 (2015) 460–466, http://dx.doi.org/10.1038/nature14542.

[2] M. Saffarian, F. Fahimi, Control of helicopters' formation using non-iterative nonlinear model predictive approach, in: Proc. of American Control Conference, 2008, pp. 3707–3712, http://dx.doi.org/10.1109/ACC.2008.4587070.

[3] J. Shin, H.J. Kim, Nonlinear model predictive formation flight, IEEE Trans. Syst. Man, Cybern. A: Syst. Humans 39 (5) (2009) 1116–1125, http://dx.doi.org/10.1109/TSMCA.2009.2021935.

[4] S. Huang, R.S.H. Teo, K.K. Tan, Collision avoidance of multi unmanned aerial vehicles: A review, Annu. Rev. Control 48 (2019) 147–164, http://dx.doi.org/10.1016/j.arcontrol.2019.10.001.

[5] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in: Proc. IEEE Int. Conf. Robot. Autom, 1985, pp. 500–505, http://dx.doi.org/10.1109/ROBOT.1985.1087247.

[6] D.H. Kim, H.O. Wang, Guohua Ye, Seiichi Shin, Decentralized control of autonomous swarm systems using artificial potential functions: analytical design guidelines, in: Proc. 43rd IEEE Conf. on Decis. and Control, Vol. 1, 2004, pp. 159–164, http://dx.doi.org/10.1109/CDC.2004.1428623.

[7] F. Augugliaro, A.P. Schoellig, R. D'Andrea, Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach, in: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., 2012, pp. 1917–1922, http://dx.doi.org/10.1109/IROS.2012.6385823.

[8] D.H. Shim, H.J. Kim, S. Sastry, Decentralized nonlinear model predictive control of multiple flying robots, in: Proc. 42nd IEEE Int. Conf. Decis. and Control, Vol. 4, 2003, pp. 3621–3626, http://dx.doi.org/10.1109/CDC.2003.1271710.

[9] R.A. Abdellatif, A.A. El-Badawy, Artificial potential field for dynamic obstacle avoidance with MPC-based trajectory tracking for multiple quadrotors, in: Proc. 2nd Novel Intell. Leading Emerg. Sci. Conf., 2020, pp. 497–502, http://dx.doi.org/10.1109/NILES50944.2020.9257973.

[10] Z. Chao, L. Ming, Z. Shaolei, Z. Wenguang, Collision-free UAV formation flight control based on nonlinear MPC, in: Proc. Int. Conf. Electron. Commun. Control, 2011, pp. 1951–1956, http://dx.doi.org/10.1109/ICECC.2011.6066578.

[11] Y. Aida, S. Suzuki, Y. Fujisawa, K. Iizuka, T. Kawamura, Y. Ikeda, Collision-free guidance control for multiple small helicopters, in: Proc. IEEE Int. Conf. Robot. Autom., 2014, pp. 5537–5543, http://dx.doi.org/10.1109/ICRA.2014.6907673.

[12] M. Kamel, J. Alonso-Mora, R. Siegwart, J. Nieto, Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control, in: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., 2017, pp. 236–243, http://dx.doi.org/10.1109/IROS.2017.8202163.

[13] D. Huang, Q. Yuan, X. Li, Decentralized flocking of multi-agent system based on MPC with obstacle/collision avoidance, in: Proc. Chin. Control Conf, 2019, pp. 5587–5592, http://dx.doi.org/10.23919/ChiCC.2019.8865184.

[14] A.T. Schwarm, M. Nikolaou, Chance constrained model predictive control, AIChE J. 45 (8) (1999) 1743–1752, http://dx.doi.org/10.1002/aic.690450811.

[15] H. Zhu, J. Alonso-Mora, Chance-constrained collision avoidance for MAVs in dynamic environments, IEEE Robot. Autom. Lett. 4 (2) (2019) 776–783, http://dx.doi.org/10.1109/LRA.2019.2893494.

[16] M. Castillo-Lopez, P. Ludivig, S.A. Sajadi-Alamdari, J.L. Sanchez-Lopez, M.A. Olivares-Mendez, H. Voos, A real-time approach for chance-constrained motion planning with dynamic obstacles, IEEE Robot. Autom. Lett. 5 (2) (2020) 3620–3625, http://dx.doi.org/10.1109/LRA.2020.2975759.

[17] X. Zhang, J. Ma, Z. Cheng, S. Huang, S.S. Ge, T.H. Lee, Trajectory generation by chance-constrained nonlinear MPC with probabilistic prediction, IEEE Trans. Cybern. 51 (7) (2021) 3616–3629, http://dx.doi.org/10.1109/TCYB.2020.3032711.

[18] J. Lin, H. Zhu, J. Alonso-Mora, Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments, in: Proc. IEEE Int. Conf. Robot. Autom., 2020, pp. 2682–2688, http://dx.doi.org/10.1109/ICRA40945.2020.9197481.

[19] P. Fiorini, Z. Shiller, Motion planning in dynamic environments using velocity obstacles, Int. J. Robot. Res. 17 (7) (1998) 760–772, http://dx.doi.org/10.1177/027836499801700706.

[20] J. van den Berg, M. Lin, D. Manocha, Reciprocal velocity obstacles for real-time multi-agent navigation, in: Proc. IEEE Int. Conf. Robot. Autom., 2008, pp. 1928–1935, http://dx.doi.org/10.1109/ROBOT.2008.4543489.

[21] J. Snape, J. van den Berg, S.J. Guy, D. Manocha, The hybrid reciprocal velocity obstacle, IEEE Trans. Robot. 27 (4) (2011) 696–706, http://dx.doi.org/10.1109/TRO.2011.2120810.

[22] J.A. Douthwaite, S. Zhao, L.S. Mihaylova, A comparative study of velocity obstacle approaches for multi-agent systems, in: Proc. UKACC 12th Int. Conf. Control, CONTROL, 2018, pp. 289–294, http://dx.doi.org/10.1109/CONTROL.2018.8516848.

[23] J. van den Berg, S.J. Guy, M. Lin, D. Manocha, Reciprocal n-body collision avoidance, in: Robot. Res., Springer, Berlin Heidelberg, 2011, pp. 3–19, http://dx.doi.org/10.1007/978-3-642-19457-3_1.

[24] T. Wakabayashi, Y. Nunoya, S. Suzuki, Dynamic obstacle avoidance of multi-rotor UAV using chance constrained MPC, in: Proc. 21st IEEE Int. Conf. Control Autom. Syst., 2021, pp. 412–417, http://dx.doi.org/10.23919/ICCAS52745.2021.9649942.

[25] L. Blackmore, M. Ono, B.C. Williams, Chance-constrained optimal path planning with obstacles, IEEE Trans. Robot. 27 (6) (2011) 1080–1094, http://dx.doi.org/10.1109/TRO.2011.2161160.

[26] T. Ohtsuka, A continuation/GMRES method for fast computation of nonlinear receding horizon control, Automatica 40 (4) (2004) 563–574, http://dx.doi.org/10.1016/j.automatica.2003.11.005.

Takumi Wakabayashi received his B.E. degree from Chiba University, Japan, in 2020. He is now a master course student of Chiba University. His research interests include control of autonomous vehicle.

Yukimasa Suzuki received his B.E. degree from Chiba University, Japan, in 2021. He is now a master course student of Chiba University. His research interests include control of UAVs.

Satoshi Suzuki received his M.E. and Ph.D. (Eng.) degrees from Chiba University, Japan, in 2006 and 2008, respectively. Between 2009 and 2018, he worked as an assistant professor and associate professor at Shinshu University. In 2019, he joined Chiba University, where he is currently an Associate Professor of the Graduate School of Engineering. His research interests include navigation, guidance, and control of autonomous flying vehicles.