Integrated Decision Control Approach for Cooperative Safety-Critical Payload Transport in a Cluttered Environment

NISHANTH RAO
SURESH SUNDARAM, Senior Member, IEEE
Indian Institute of Science, Bengaluru, India

In this article, the problem of coordinated transportation of heavy payload by a team of unmanned aerial vehicles (UAVs) in a cluttered environment is addressed. The payload is modeled as a rigid body and is assumed to track a precomputed global flight trajectory from a start point to a goal point. Due to the presence of local dynamic obstacles in the environment, the UAVs must ensure that there is no collision between the payload and these obstacles while ensuring that the payload oscillations are kept minimum. An integrated decision controller (IDC) is proposed that integrates the optimal tracking control law given by a centralized model predictive controller with safety-critical constraints provided by the exponential control barrier functions. The entire payload-UAV system is enclosed by a safe convex hull boundary, and the IDC ensures that no obstacle enters this boundary. To evaluate the performance of the IDC, the results for a numerical simulation as well as a high-fidelity `Gazebo` simulation are presented. An ablation study is conducted to analyze the robustness of the proposed IDC against practical dubieties like noisy state values, relative obstacle safety margin, and payload mass uncertainty. The proposed method is then compared with a baseline controller qualitatively, emphasizing the use of barrier functions in safety-critical applications. The results clearly show that the IDC achieves both trajectory tracking and obstacle avoidance successfully while restricting the payload oscillations within a safe limit.

Manuscript received 6 January 2023; revised 2 July 2023; accepted 25 August 2023. Date of publication 6 September 2023; date of current version 8 December 2023.

DOI. No. 10.1109/TAES.2023.3312065

Refereeing of this contribution was handled by Elisa Capello.

This work was supported in part by the Nokia CSR grant on Networked Robotics, and in part by the Ministry of Electronics and Information Technology (MeITy).

Authors' addresses: Nishanth Rao and Suresh Sundaram are with the Artificial Intelligence and Robotics Lab, Department of Aerospace Engineering, Indian Institute of Science, Bengaluru 560012, India, E-mail: (vssuresh@iisc.ac.in; nishanthrao@iisc.ac.in). *(Corresponding author: Nishanth Rao).*

I. INTRODUCTION

Recent advancements in unmanned aerial vehicles (UAVs) and sensor technology has created a widespread interest in their ability to solve payload transportation problems in the logistics sector. Collaborative interaction of UAVs can efficiently solve many transportation problems, especially when there are requirements for delivering versatile payloads that vary in size. This is because utilizing multiple UAVs to lift payloads are economically feasible and efficient, as opposed to utilizing a single UAV of a bigger form factor. Industrial missions [1], military expeditions [2], and medical search-rescue operations [3] demand that the UAVs have to ensure proper delivery of the payload along a set preplanned global trajectory, while at the same time, preventing collisions with the local dynamic obstacles present around it and maintaining minimal oscillations of the payload throughout the course. However, when there are multiple UAVs interacting with one another to transport a payload, existing path planning algorithms [4] fall short, as they do not consider the kinodynamic-maneuverability of the system and it is highly coupled complex dynamics.

Majority of existing literature [5], [6], [7] models the payload suspension as a point mass cable suspension. This results in simplified system dynamics, which is generally unrealistic in situations where there are requirements to transport heavy payloads. In [8], a payload suspension model is used, and formation control of UAVs are considered to negotiate a tight passage. However formation control is not practically possible for rigid link payload suspension with minimal oscillation constraints. Moreover, additional mechanisms like rack-pinion attachment points must be employed during formation control when rigid link suspensions are employed.

In [9] and [10], the payload is modeled as an external disturbance and its direction is along the cable connecting the payload and the UAV. This may lead to suboptimal results, as the control law cannot leverage the benefits of knowing the actual dynamics of the payload. For example, in cases when the payload is a rigid body unlike a point mass, it is important to know where the payload boundaries are, especially for designing obstacle avoidance algorithms while navigating in a cluttered environment. In addition, it is crucial to consider the payload orientation during controller design so that payload oscillations can be kept minimal. This is not possible when the payload is modeled as an external disturbance. Further, artificial potential fields (APFs) are considered for the purpose of obstacle avoidance in [9]. However, APFs can have limitations in certain scenarios like passing between closely spaced obstacles, exhibiting oscillations or being trapped in local minima as discussed in [11]. Moreover, control barrier functions (CBFs) are more suited for obstacle avoidance than APFs [12].

In contrast, [13] models the cable-payload suspension as a series of mass–spring–damper system. A null-space based controller is then designed for the purpose of trajectory tracking and obstacle avoidance. While cable suspension of light, point mass payload may be desirable in some circumstances, they are not desirable when the payload is heavy, as they may cause undesirable uncontrollable oscillations due to the cable elasticity. Further, with cable suspension, there can exist a point in the course of the mission where the cables can become slack, causing the payload to become massless, or even accelerate upwards. Controlling the states of the payload in these situations becomes complicated and may not be feasible in practical situations where the exact states of the payload like its relative position and linear acceleration are not known, but needs to be estimated based on the cable orientation and the tension in it. It can also lead to singularities in the dynamics of the system [14], resulting in extremely complex control laws. The aforementioned works address the problem of cooperative payload transport, it is evident that there is a need to model heavy payload as a rigid body, with rigid link suspension, which enables easy control over the states of the payload. Moreover, safety-critical payload transport must be ensured in a cluttered environment, with minimum deviation from the preset global trajectory while avoiding obstacles.

In this article, the payload-UAV system is modeled end-to-end, keeping in mind the necessity to control the states of the payload throughout the course of the mission. The dynamics of the system is modeled using Lagrangian mechanics as discussed in [15], and rigid, massless links are used in place of cable suspension, as demonstrated in [16]. This makes it relatively easier to calculate the payload states at all times. Moreover, using rigid links provides complete control authority to the UAVs to handle the payload as required. To ensure accurate tracking of the payload, a centralized model predictive controller (MPC) is employed. As of now, centralized MPC is shown to outperform distributed MPC at a lower sampling rate in [16]. Furthermore, the major advantage of using MPC based controllers is its inherent ability to take into consideration the state and control input constraints. The concept of receding horizon also ensures that future uncertainties and requirements are taken into consideration while obtaining the control law at the current time step. Inspired by the application of CBFs to safety-critical systems, an obstacle avoidance controller is developed for the entire payload-UAV system using exponential CBFs (ECBFs). Barrier functions naturally provide safety-critical constraints that can be leveraged in an optimization framework . The *integrated decision controller* (IDC) fuses the optimal tracking control law as provided by the MPC into an optimization-based obstacle avoidance decision controller. The IDC is designed in a way that allows any trajectory tracking controller to be used with the obstacle avoidance controller.

The major contributions of this article are as follows.

1) Developed an ECBF-based obstacle avoidance algorithm for the *entire* multi-UAV payload system.
2) Proposed an *integrated control approach* for simultaneous dynamic trajectory tracking and obstacle avoidance.

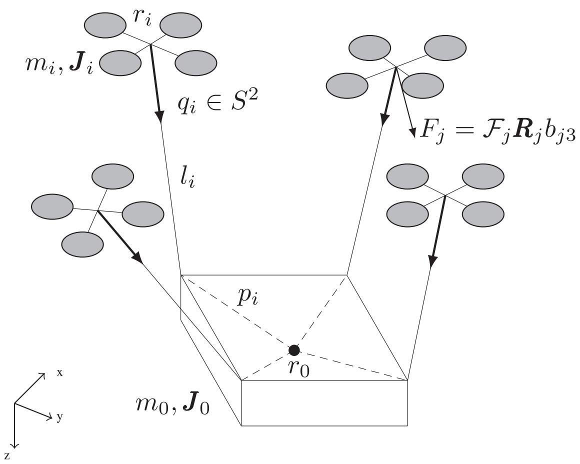


Fig. 1. Schematic diagram of N UAVs transporting a rigid body, attached via massless rigid links present in the NED coordinate frame is shown. The black dot at the center of the payload denotes its center of mass. r_0 and r_i denote the position of the payload and the ith UAV, respectively, and F_j denotes the force vector that the UAV produces.

3) The ablation study demonstrates the IDC's robustness to uncertainties like noisy payload state estimates, payload-mass uncertainty, and relative safety margins.
4) Highlighted the effectiveness of the IDC approach over state-of-art potential-field based obstacle avoidance *for multi-UAV payload* system.
5) Implemented and validated in a *real-time high-fidelity* `Gazebo` simulator to demonstrate its feasibility.

The rest of this article is organized as follows. Section II covers the system description and state space formulation. Section III covers the design and development of the IDC for the payload-UAV system, with brief discussion on model predictive control and exponential barrier functions. In Section IV, the proposed controller is evaluated, and the results for numerical simulations as well as a `Gazebo` simulation are presented along with an ablation study demonstrating the robustness of the IDC. A qualitative comparison with a standard baseline controller utilizing APFs is presented, to emphasize the importance of using barrier functions for safety-critical applications. Finally, Section V concludes this article.

II. System Model

In this section, a brief description of the equations of motion for the payload-UAV system is provided. The derivation of the system dynamics parallels the discussion presented in [15] except that the equations of motion are resolved in the payload frame of [16].

A. Payload-UAV System Description

There are N UAVs that are collaboratively transporting a rigid payload, as shown in Fig. 1. The UAVs are attached to the payload via massless rigid links. The rigid links are connected to the payload and the UAVs via spherical joints.

This way, the attitude dynamics of the UAVs are decoupled from the dynamics of the rest of the system.

Throughout this article, the North, East, Down (NED) coordinate system is used, with the positive z-axis pointing downward. The direction for positive x- and y- axes are then chosen on the basis of the right hand rule. The basis vectors for the inertial frame and the body-fixed frames are denoted by $\{e_1, e_2, e_3\}$ and $\{b_{i1}, b_{i2}, b_{i3}\}$, respectively. The variables that refer to the payload are subscripted by 0, and the variables that refer to the ith-UAV is subscripted by i.

The location of the center of mass of the payload in the inertial frame is denoted by $r_0 \in \mathbb{R}^3$, and the location of the center of mass of the ith-UAV in the inertial frame is denoted by $r_i \in \mathbb{R}^3$. The position vector of the attachment point of the rigid link of length l_i corresponding to the ith-UAV in the payload frame is denoted by $p_i \in \mathbb{R}^3$. The unit vector along the rigid link corresponding to the ith-UAV, pointing toward the payload is denoted by $q_i \in \mathbb{S}^2$, where $\mathbb{S}^2 = \{q_i \in \mathbb{R}^3 \mid \|q_i\|_2 = 1\}$. The attitude of the payload is characterized by $\boldsymbol{R}_0 \in SO(3)$, the rotation matrix that rotates a vector in the payload-fixed frame to the inertial frame. The attitude of the ith-UAV is characterized by $\boldsymbol{R}_i \in SO(3)$, the rotation matrix that rotates a vector in the ith-UAV fixed frame to the inertial frame. Let $\boldsymbol{J}_0 \in \mathbb{R}^{3\times 3}$ denote the inertia matrix of the payload with mass m_0, and $\boldsymbol{J}_i \in \mathbb{R}^{3\times 3}$ denote the inertia matrix of the ith-UAV with mass m_i.

The force vector $F_i \in \mathbb{R}^3$ generated by the ith-UAV in the inertial frame is denoted by $F_i = \mathcal{F}_i \boldsymbol{R}_i b_{i3}$, where $\mathcal{F}_i \in \mathbb{R}$ denotes the thrust value. In addition to this, the torque vector generated by the ith-UAV in the body-fixed frame is denoted by $\tau_i \in \mathbb{R}^3$. Thus, the control inputs to the payload-UAV system is given by $\{\mathcal{F}_i, \tau_i\}$, $i \in \{1, ..., N\}$.

B. Payload-UAV System Dynamics

Based on the notations and the system description presented previously, the equations of motion are derived in the following section. As the connecting links are rigid and massless, the position vector r_i of the ith-UAV can be directly inferred as

$$r_i = r_0 + \boldsymbol{R}_0 \left(p_i - l_i q_i\right), \quad i = 1, \ldots, N. \tag{1}$$

The kinematic equations for the payload-UAV system are

$$\dot{r}_0 = \boldsymbol{R}_0 v_0 \tag{2}$$

$$\dot{\boldsymbol{R}}_0 = \boldsymbol{R}_0 \omega_0^{\times} \tag{3}$$

$$\dot{r}_i = \dot{r}_0 + \boldsymbol{R}_0 \left(\omega_0^{\times} p_i - l_i \Omega_i^{\times} q_i\right) \tag{4}$$

$$\dot{q}_i = \left(\Omega_i - \omega_0\right)^{\times} q_i \tag{5}$$

$$\dot{\boldsymbol{R}}_i = \boldsymbol{R}_i \omega_i^{\times}, \qquad i = 1, \ldots, N \tag{6}$$

where $(\cdot)^{\times} : \mathbb{R}^3 \to SO(3)$ is the skew-symmetric operator, $v_0 \in \mathbb{R}^3$ denotes the velocity of the payload in the payload-fixed frame, and w_0, $w_i \in \mathbb{R}^3$ denotes the angular velocities of the payload and the ith UAV, respectively. Let the angular velocities of the ith link be $\Omega_i \in \mathbb{R}^3$. Note that the orthogonal condition $q_i \cdot \Omega_i = 0$ must be satisfied for all time.

The kinetic energy $\mathcal{T}$ and the potential energy $\mathcal{U}$ of the system are given by

$$\mathcal{T} = \frac{1}{2} m_0 \|\dot{r}_0\|_2^2 + \frac{1}{2}\omega_0 \cdot \boldsymbol{J}_0 \omega_0 \sum_{i=1}^{N} \left(\frac{1}{2} m_i \|\dot{r}_i\|_2^2 + \frac{1}{2}\omega_i \cdot \boldsymbol{J}_i \omega_i\right) \tag{7}$$

$$\mathcal{U} = -m_0\, g\mathbb{k} \cdot r_0 - \sum_{i=1}^{N} m_i g\mathbb{k} \cdot r_i \tag{8}$$

where $\mathbb{k} = [0\ 0\ 1]^T$. To derive the equations of motion of the payload-UAV system, the *Lagrangian* $\mathcal{L}$ of the system must be obtained as

$$\mathcal{L} = \mathcal{T} - \mathcal{U}. \tag{9}$$

Using Lagrangian $\mathcal{L}$, the Euler–Lagrange equations can be obtained from the *Lagrange-d'Alembert Principle* as described in [15, Appendix] and are given as follows:

$$\frac{d}{dt}\boldsymbol{D}_{\dot{r}_0}\mathcal{L} - \boldsymbol{D}_{r_0}\mathcal{L} = \sum_{i=1}^{N} F_i \tag{10}$$

$$\frac{d}{dt}\boldsymbol{D}_{\omega_0}\mathcal{L} + \omega_0^{\times}\boldsymbol{D}_{\omega_0}\mathcal{L} - \boldsymbol{d}_{\boldsymbol{R}_0}\mathcal{L} = \sum_{i=1}^{N} p_i^{\times}\boldsymbol{R}_0^T F_i \tag{11}$$

$$q_i^{\times}\frac{d}{dt}\boldsymbol{D}_{\dot{q}_i}\mathcal{L} - q_i^{\times}\boldsymbol{D}_{q_i}\mathcal{L} = -l_i q_i^{\times} F_i \tag{12}$$

$$\frac{d}{dt}\boldsymbol{D}_{\omega_i}\mathcal{L} + \omega_i^{\times}\boldsymbol{D}_{\omega_i}\mathcal{L} = \tau_i \tag{13}$$

where $\boldsymbol{D}_a\mathcal{L}$ is the derivative of the Lagrangian $\mathcal{L}$, w.r.t., the vector a, and $\boldsymbol{d}_{\boldsymbol{R}_0}\mathcal{L}$ are the *left-trivialized derivatives* of the Lagrangian $\mathcal{L}$ [15]. Substituting for the derivatives of the Lagrangian, and simplifying the equations leads to the equations of motion of the payload-UAV system [16]

$$\begin{aligned} m_T\left(\dot{v}_0 + \omega_0^{\times} v_0\right) + \sum_{i=1}^{N} m_i\left(-p_i^{\times}\dot{\omega}_0 + l_i q_i^{\times}\dot{\Omega}_i + l_i q_i^{\times}\omega_0^{\times}\Omega_i\right) \\ + \sum_{i=1}^{N}\left(m_i\left(\omega_0^{\times}\right)^2 p_i + m_i l_i \|\Omega_i\|_2^2 q_i\right) = \boldsymbol{R}_0^T\left(m_T g\mathbb{k} + \sum_{i=1}^{N} F_i\right) \end{aligned} \tag{14}$$

$$\begin{aligned} \sum_{i=1}^{N} m_i p_i^{\times}\left(\dot{v}_0 + \omega_0^{\times} v_0 + l_i q_i \dot{\Omega}_i + l_i \|\Omega_i\|_2^2 q_i + l_i q_i^{\times}\omega_0^{\times}\Omega_i\right) \\ + \bar{\boldsymbol{J}}_0\dot{\omega}_0 + \omega_0^{\times}\bar{\boldsymbol{J}}_0\omega_0 = \sum_{i=1}^{N} p_i^{\times}\boldsymbol{R}_0^T\left(F_i + m_i g\mathbb{k}\right) \end{aligned} \tag{15}$$

$$\begin{aligned} m_i l_i^2\left(\dot{\Omega}_i + \omega_0^{\times}\Omega\right) - m_i l_i q_i^{\times}\left(\dot{v}_0 + \omega_0^{\times} v_0\right) + m_i l_i q_i^{\times} p_i^{\times}\dot{\omega}_0 \\ - m_i l_i q_i^{\times}\left(\omega_0^{\times}\right)^2 p_i = -l_i q_i^{\times}\boldsymbol{R}_0^T\left(F_i + m_i g\mathbb{k}\right) \end{aligned} \tag{16}$$

$$\boldsymbol{J}_i\dot{\omega}_i + \omega_i^{\times}\boldsymbol{J}_i\omega_i = \tau_i \tag{17}$$

where $m_T = (m_0 + \sum_{i=1}^{N} m_i)$ is the total mass of the UAVs with the payload and $\bar{\boldsymbol{J}}_0 = (\boldsymbol{J}_0 - \sum_{i=1}^{N} m_i (p_i^{\times})^2)$.

C. State-Space Representation and Model Linearization

The equations of motion described earlier can be rearranged to obtain state-space representation of the payload-UAV system. Consider the state vector $x \in \mathbb{R}^{12+12N}$ as follows:

$$x = \left[\underbrace{r_0^T \;\; v_0^T \;\; \Theta_0^T \;\; \omega_{0^T}}_{\text{payload}\in\mathbb{R}^{12}} \;\; \underbrace{q_i^T \;\; \Omega_i^T \;\; \Theta_i^T \;\; \omega_i^T}_{\text{i}^{\text{th}}\text{UAV}\in\mathbb{R}^{12}} \right]^T, \; i \in \{1, \ldots, N\} \tag{18}$$

where $\Theta_0, \Theta_i \in \mathbb{R}^3$ are the ZYX Euler angle parameterization of $\boldsymbol{R}_0$ and $\boldsymbol{R}_i$, respectively. The state-space equations for $\dot{x}_a = [\dot{r}_0^T \; \dot{\Theta}_0^T \; \dot{q}_i^T \; \dot{\Theta}_i^T \, \dot{\omega}_i^T\,]^T$ can be obtained from (2)–(6) and (17). The equations for $\dot{x}_b = [\dot{v}_0^T \; \dot{\omega}_0^T \; \dot{\Omega}_i^T\,]^T$ can be obtained by rearranging (14)–(16) and is given by

$$\dot{x}_b = \boldsymbol{P}^{-1}\boldsymbol{Q} \tag{19}$$

where $\boldsymbol{P} =$

$$\left[\begin{array}{cc|ccc} m_T\boldsymbol{I}_3 & -\sum_{i=1}^N m_i p_i^\times & m_1 l_1 q_1^\times & \ldots & m_N l_N q_N^\times \\ \sum_{i=1}^N m_i p_i^\times & \bar{\boldsymbol{J}}_0 & m_1 l_1 p_1^\times q_1^\times & \ldots & m_N l_N p_N^\times q_N^\times \\ \hline m_1 l_1 q_1^\times & m_1 l_1 q_1^\times p_1^\times & m_1 l_1^2 \boldsymbol{I}_3 & \ldots & \boldsymbol{0}_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -m_N l_N q_N^\times & m_N l_N q_N^\times p_N^\times & \boldsymbol{0}_3 & \ldots & m_N l_N^2 \boldsymbol{I}_3 \end{array}\right] \tag{20}$$

and $\boldsymbol{Q} =$

$$\left[\begin{array}{c} -m_T \omega_0^\times v_0 - \sum_{i=1}^N \left\{ m_i \left(\omega_0^\times\right)^2 p_i + m_i l_i \left\|\Omega_i\right\|_2^2 q_i \right. \\ \left. +m_i l_i q_i^\times \omega_0^\times \Omega_i \right\} + m_T g \boldsymbol{R}_0^T \mathbb{k} + \sum_{i=1}^N \boldsymbol{R}_0^T F_i \\ \hline -\omega_0^\times \bar{\boldsymbol{J}}_0 \omega_0 - \sum_{i=1}^N m_i \left\{ p_i^\times \omega_0^\times v_0 + l_i p_i^\times q_i^\times \omega_0^\times \Omega_i \right. \\ \left. +l_i p_i^\times \left\|\Omega_i\right\|_2^2 q_i \right\} + \sum_{i=1}^N p_i^\times \boldsymbol{R}_0^T \left(F_i + m_i g \mathbb{k}\right) \\ \hline inem_1 l_1 \left\{ q_1^\times \left(\omega_0^\times\right)^2 p_1 - l_1 \omega_0^\times \Omega_1 + q_1^\times \omega_0^\times v_0 \right\} \\ -l_1 q_1^\times \boldsymbol{R}_0^T \left(F_1 + m_1 g \mathbb{k}\right) \\ \hline \vdots \\ \hline m_N l_N \left\{ q_N^\times \left(\omega_0^\times\right)^2 p_N - l_N \omega_0^\times \Omega_N + q_N^\times \omega_0^\times v_0 \right\} \\ -l_N q_N^\times \boldsymbol{R}_0^T \left(F_N + m_N g \mathbb{k}\right) \end{array}\right] \tag{21}$$

where $\boldsymbol{0}_n \in \mathbb{R}^{n\times n}$ is the $n \times n$ zero matrix and $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ is the $n \times n$ identity matrix. The control input to the system is defined by the vector $u = [\mathcal{F}_1 \;\; \tau_1^T \;\; .. \;\; \mathcal{F}_N \;\; \tau_N^T]^T \in \mathbb{R}^{4N}$. The equilibrium points of the system $\{x_e, u_e\}$ satisfy the condition $f(x_e, u_e) = 0$. One such equilibrium point (x_e, u_e) for the payload-UAV system is given in (22), and occurs when all the links are vertical, and the orientation of the payload and all the UAVs are zero, thus producing zero torques. In addition to this, the thrust generated by the UAVs must balance the overall weight they are subjected to, maintaining net zero acceleration

$$x_e = \left[r_0^T \; \mathbb{0}^T \; \mathbb{0}^T \; \mathbb{0}^T \; \mathbb{k}_i^T \; \mathbb{0}_i^T \; \mathbb{0}_i^T \; \mathbb{0}_i^T\right]^T, i \in \{1, .., N\} \tag{22a}$$

$$u_e = \left[\mathcal{F}_{e_i} \; \mathbb{0}_i^T\right]^T, i \in \{1, .., N\} \tag{22b}$$

where $\mathbb{0} \in \mathbb{R}^3$ is the zero vector and $\mathcal{F}_{e_i} = (m_i + m_0/N)g$. The nonlinear state-space equations can then be linearized about the equilibrium point (x_e, u_e). The resultant discretized equations of motion at time step k is given by

$$\Delta x_{k+1} = \boldsymbol{A}\Delta x_k + \boldsymbol{B}\Delta u_k \tag{23a}$$

$$\Delta y_k = \boldsymbol{C}\Delta x_k + \boldsymbol{D}\Delta u_k \tag{23b}$$

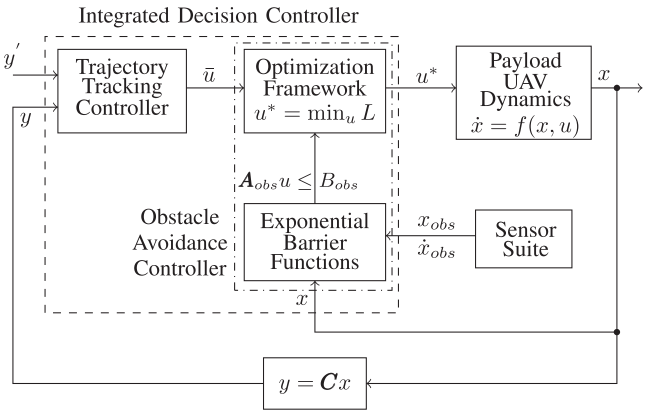


Fig. 2. Schematic diagram of the IDC. The IDC comprises of a trajectory tracking controller and an obstacle avoidance controller.

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the state matrix,[2] $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ is the input matrix,[1] $\boldsymbol{C} \in \mathbb{R}^{p\times n}$ is the output matrix, and $\boldsymbol{D} \in \mathbb{R}^{p\times m}$ is the input feedforward matrix. The discrete state vector $\Delta x_k \in \mathbb{R}^n$ and the control input vector $\Delta u_k \in \mathbb{R}^m$ is obtained as $\Delta x_k = x_k - x_e$, $\Delta u_k = u_k - u_e$, respectively. Due to the principle of receding horizon, it is assumed implicitly that the input u_k cannot affect the output y_k at the same time. Thus, throughout the discussion that follows, it is assumed that $\boldsymbol{D} = \boldsymbol{0}$.

III. IDC Design for Safety-Critical Navigation

In this section, an IDC is designed, that meets the requirements of trajectory tracking and obstacle avoidance. Particularly, the IDC must ensure that the payload oscillations are minimum, while tracking a set trajectory. Moreover, the IDC must be capable of providing safety critical decision control in order to avoid dynamic obstacles in the cluttered environment, while maintaining minimum deviation from the set trajectory. The schematic of the IDC is shown in Fig. 2. The structure of the IDC ensures powerful versatility in the sense that any tracking controller can be used in conjunction with the obstacle avoidance controller. In this work, an MPC is designed for the linear discretized system (23), with particular emphasis on the derivation of state and control input constraints. This way, the bounds on payload oscillations and tracking requirements can be automatically ensured in a single framework.

[1] Due to their high dimensionality, the matrices are uploaded as. csv files, and can be viewed at: https://drive.google.com/drive/folders/1ZIwZ4higpC-BZ5isK-Gr1uozPNqTscE7.

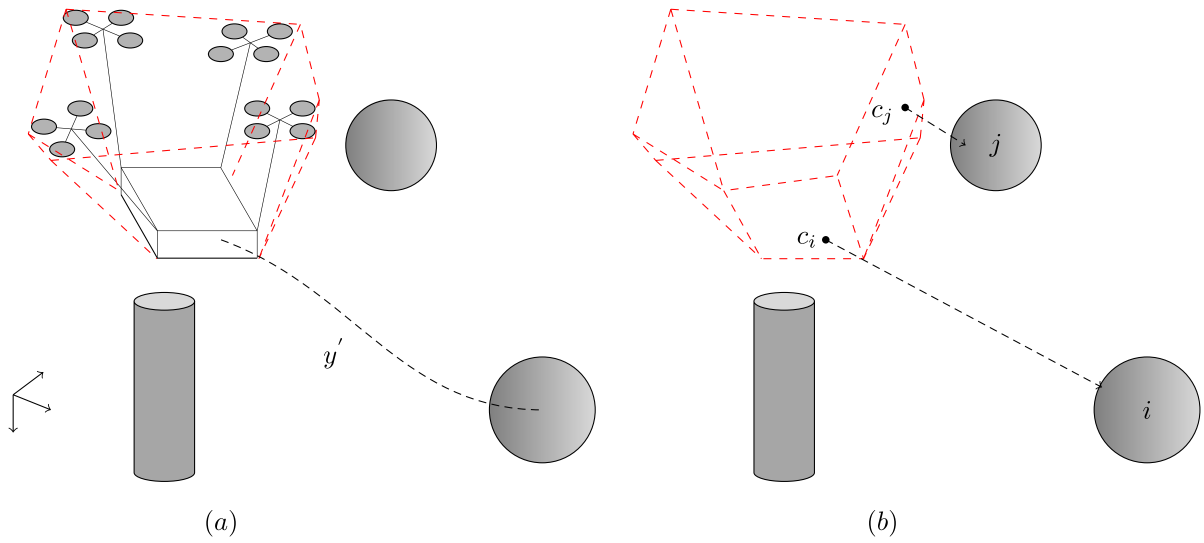


Fig. 3. Payload must track the desired trajectory y' amidst a cluttered environment. The cluttered environment consists of static and dynamic obstacles, which are shown by spheres and cylinders. (a) Formation of the convex hull (dashed lines) around the payload-UAV system. The convex hull encloses all the UAVs and the payload completely. (b) Points c_i and c_j are the points on the convex hull closest to the ith and the jth obstacle, respectively.

A. Safety Critical Payload Transport in a Cluttered Environment

This work focuses on ensuring the safety-critical transportation of a payload in a cluttered environment. In particular, there must be no collision of the payload with any surrounding obstacles, and the payload oscillations must be small

$$\begin{aligned} |\phi_0| &\le \delta_\phi && \text{(24a)}\\ |\theta_0| &\le \delta_\theta && \text{(24b)}\\ |\psi_0| &\le \delta_\psi \quad \forall t > t_0 && \text{(24c)}\end{aligned}$$

where $\Theta_0 = [\phi_0\ \theta_0\ \psi_0]^T$ is the roll–pitch–yaw (rpy) of the payload, and $\delta_\phi, \delta_\theta, \delta_\psi > 0$ are small positive constants. The cluttered environment consists of static and dynamic obstacles, whose position x_{obs} and velocity $\dot{x}_{\text{obs}}$ can be estimated by the onboard UAV sensor suite.

B. Trajectory Tracking Controller

A linear MPC (LMPC) can be designed for the system (23), such that the payload tracks a given reference trajectory y' as shown in in Fig. 3(a). The LMPC minimizes the cost function J at every time step k as

$$\begin{aligned} J\,(\Delta x_k, \Delta U_k) = {} & \left\|\bar{\boldsymbol{C}} X_e + \Delta Y_k - Y_k'\right\|^2_{\bar{\boldsymbol{Q}}} + \|\Delta U_k\|^2_{\bar{\boldsymbol{R}}} \\ & + \left\|y_{k+N_p} - y'_{k+N_p}\right\|^2_{\boldsymbol{Q}_f} \qquad (25)\end{aligned}$$

where $\bar{\boldsymbol{C}} = \text{diag}(\boldsymbol{C}, .., \boldsymbol{C})$, $\bar{\boldsymbol{Q}} = \text{diag}(\boldsymbol{Q}, .., \boldsymbol{Q})$, $\bar{\boldsymbol{R}} = \text{diag}(\boldsymbol{R}, .., \boldsymbol{R})$, $X_e = [x_e^T\ ..\ x_e^T]^T$, $\Delta Y_k = [\Delta y_k^T\ ..\ \Delta y_{k+N_p-1}^T]^T$, and $\Delta U_k = [\Delta u_k^T, .., \Delta u_{k+N_p-1}^T]^T$ are defined over the prediction horizon N_p. The matrices $\boldsymbol{Q}, \boldsymbol{Q}_f \ge 0$ and $\boldsymbol{R} > 0$ are the cost weighting matrices and $y_k = \boldsymbol{C}(x_e + \Delta x_k)$. The matrix $\boldsymbol{Q}_f$ is chosen to be equal to the solution of the discrete algebraic Riccati equation for the discrete system in (23) [17].

The cost function in (25) is subject to linear state and control input constraints. The nonlinear constraints derived in [16] needs to be linearized about every time step along with the model. This can be computationally demanding in a real-time implementation. A simplified set of constraints can be derived for the discrete linear system in (23). Let the upper and lower bounds for the input be u_{ub} and u_{lb}, respectively. Thus, $u_{lb} \le u_e + \Delta u_{k+i} \le u_{ub}$, which can also be expressed as

$$\begin{bmatrix} \boldsymbol{I}_m \\ -\boldsymbol{I}_m \end{bmatrix} \Delta u_{k+i} \le \begin{bmatrix} (u_{ub} - u_e) \\ (u_e - u_{lb}) \end{bmatrix} \qquad (26)$$

Similarly, let the upper and lower bounds for the state be x_{ub} and x_{lb}, respectively, for the set of states described by $\boldsymbol{c}_z x_{k+i}$, where the matrix $\boldsymbol{c}_z$ is a diagonal matrix, with ones for the states that require constraints, and zeros for the states that are free to take any values. The constraint $x_{lb} \le \boldsymbol{c}_z(x_e + \Delta x_{k+i}) \le x_{ub}$ can be expressed as

$$\begin{bmatrix} \boldsymbol{c}_z \\ -\boldsymbol{c}_z \end{bmatrix} \Delta x_{k+i} \le \begin{bmatrix} (x_{ub} - \boldsymbol{c}_z x_e) \\ (\boldsymbol{c}_z x_e - x_{lb}) \end{bmatrix}. \qquad (27)$$

The state vector Δx_{k+i} can be recursively substituted into (23), to obtain $\Delta X_k = [\Delta x_k^T\ ..,\ \Delta x_{k+N_p-1}^T]^T$, and thus, over the prediction horizon these constraint equations can be compactly written as

$$\boldsymbol{M}_U \Delta U_k \le \Delta U_b \qquad (28)$$

$$\boldsymbol{M}_x\,(\boldsymbol{F}\Delta x_k + \boldsymbol{H}\Delta U_k) \le \Delta Z_b \qquad (29)$$

where

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{0} & & & & \\ \boldsymbol{B} & \boldsymbol{0} & & & \\ \boldsymbol{AB} & \boldsymbol{B} & \boldsymbol{0} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \boldsymbol{A}^{N_p-2}\boldsymbol{B} & \boldsymbol{A}^{N_p-3}\boldsymbol{B} & \dots & \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}$$

$$\boldsymbol{M}_U = \begin{bmatrix} \begin{bmatrix} \boldsymbol{I}_m \\ -\boldsymbol{I}_m \end{bmatrix} & & \\ & \ddots & \\ & & \begin{bmatrix} \boldsymbol{I}_m \\ -\boldsymbol{I}_m \end{bmatrix} \end{bmatrix}$$

$$\boldsymbol{M}_x = \begin{bmatrix} \begin{bmatrix} \boldsymbol{c}_z \\ -\boldsymbol{c}_z \end{bmatrix} & & \\ & \ddots & \\ & & \begin{bmatrix} \boldsymbol{c}_z \\ -\boldsymbol{c}_z \end{bmatrix} \end{bmatrix}$$

$$\Delta U_b = \begin{bmatrix} \begin{bmatrix} (u_{ub} - u_e) \\ (u_e - u_{lb}) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (u_{ub} - u_e) \\ (u_e - u_{lb}) \end{bmatrix} \end{bmatrix}, \ \boldsymbol{F} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^{N_p-1} \end{bmatrix}$$

$$\Delta Z_b = \begin{bmatrix} \begin{bmatrix} \left(x_{ub} - \boldsymbol{c}_z x_e\right) \\ \left(\boldsymbol{c}_z x_e - x_{lb}\right) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \left(x_{ub} - \boldsymbol{c}_z x_e\right) \\ \left(\boldsymbol{c}_z x_e - x_{lb}\right) \end{bmatrix} \end{bmatrix}.$$

Equation (29) can be rearranged, to obtain the constraints in terms of ΔU_k, as in (28). Once these constraints are obtained, the cost function in (25) can be solved as

$$\boxed{\begin{aligned} \Delta \bar{U}_k &= \underset{\Delta U_k}{\arg\min}\ J(\Delta x_k, \Delta U_k) \\ \text{s.t}\ \begin{bmatrix} \boldsymbol{M}_U \\ \boldsymbol{M}_x \boldsymbol{H} \end{bmatrix} \Delta U_k &\leq \begin{bmatrix} \Delta U_b \\ \Delta Z_b - \boldsymbol{M}_x \boldsymbol{F} \Delta x_k \end{bmatrix} \end{aligned}} \quad \begin{aligned} &(30a) \\ &(30b) \end{aligned}$$

and the optimal tracking control input over N_p horizon $\bar{U}_k$ is given by $\bar{U}_k = U_e + \Delta\bar{U}_k$, where $U_e = [u_e^T \dots u_e^T]^T$. The convergence proof for the quadratic programming (QP) formulation of (30) can be found in [17] .

The constraints include bounds on the control input $u = [\mathcal{F}_1\ \tau_1^T \dots \mathcal{F}_N\ \tau_N^T]^T$, and bounds on states such as the z-coordinates of the payload and the UAVs, which cannot be positive (due to the NED convention) and the set of constraints in (24). The reference trajectory y' contains the desired position of the payload, the desired orientation of the payload to be zero ensuring minimal oscillations and the desired orientation of links to be vertical, ensuring that the UAVs do not collide with each other, and the system stays close to the equilibrium point (x_e, u_e) as described in (22). Thus, the optimal tracking control input $\bar{u}_k$ at the current step k is obtained as $\bar{u}_k = [\boldsymbol{I}\ \boldsymbol{0}\ \dots\ \boldsymbol{0}]\bar{U}_k$.

C. Obstacle Avoidance Controller

The obstacles are dynamic in nature, and it is assumed that the position and velocity of the obstacle are available at every time step k. Due to the spatial structure of the payload-UAV system, it must be ensured that the rigid links and the propellers of the UAVs do not collide with the surrounding obstacles. This is ensured by constructing a *safe* convex hull around the payload-UAV system such that it fully encloses it, as shown in Fig. 3(a). The collision avoidance problem is solved using the ECBFs [18]. In general, the CBF is defined over the state space, and provides safety-critical constraints, that can be leveraged in an optimization framework. The ECBF naturally allows extension of CBFs to higher relative-degree systems, such as the payload-UAV system described in Section II.

Suppose there exists an r-times continuously differentiable function $h(x) : \mathcal{D} \to \mathbb{R},\ \mathcal{D} \subset \mathbb{R}^n$, that is a function of the state vector $x \in \mathcal{D}$. Define a super level set $\mathcal{C}$ over h as

$$\begin{aligned} \mathcal{C} &= \{x \in D \subset \mathbb{R}^n : h(x) \geq 0\} && \text{(31a)} \\ \partial\mathcal{C} &= \{x \in D \subset \mathbb{R}^n : h(x) = 0\} && \text{(31b)} \\ \text{Int}\,(\mathcal{C}) &= \{x \in D \subset \mathbb{R}^n : h(x) > 0\} && \text{(31c)} \end{aligned}$$

where $\partial\mathcal{C}$ represents the boundary of the set $\mathcal{C}$ and Int($\mathcal{C}$) represents the interior of the set $\mathcal{C}$. The set $\mathcal{C}$ is said to be *forward invariant*, if for every $x(t_0) = x_0 \in \mathcal{C}$, the state trajectory $x(t) \in \mathcal{C}\ \forall\ t > t_0$, i.e., the trajectory $x(t)$ never leaves $\mathcal{C}$ if it starts in $\mathcal{C}$. For a system whose dynamics can be written as $\dot{x} = f(x) + g(x)u$, the relative degree of $h(x)$ is defined as the number of times h must be differentiated before the input u appears explicitly. Thus, the function h is said to have a degree r if $\mathfrak{L}_g\mathfrak{L}_f^{r-1}h(x) \neq 0$ and $\mathfrak{L}_g\mathfrak{L}_f h(x) = \mathfrak{L}_g\mathfrak{L}_f^2 h(x) = \dots = \mathfrak{L}_g\mathfrak{L}_f^{r-2}h(x) = 0$, where $\mathfrak{L}_a b(x) = a \cdot \frac{\partial b}{\partial x}$ denotes the lie derivative of the vector field b and a.

THEOREM 1 ([18]) Given a set $\mathcal{C} \subset \mathbb{R}^n$ defined as a super level set of an r-times continuously differentiable function $h : \mathcal{D} \to \mathbb{R}$, then h is an *ECBF* if there exists a row vector $K \in \mathbb{R}^r$ such that

$$\inf_u \left[\mathfrak{L}_f^r h(x) + \mathfrak{L}_g\mathfrak{L}_f^{r-1}h(x)u + K\eta(x)\right] \geq 0 \qquad (32)$$

$\forall\ x \in \mathcal{C}$, where $\eta(x) = [h(x)\ \dot{h}(x)\ \dots\ h^{(r-1)}(x)]^T$. The set $\mathcal{C}$ is then a *forward invariant* set.

Consider the safe set $\mathcal{C} = \mathbb{R}^3 - \mathcal{O}$, where the obstacle set $\mathcal{O}$ is the space occupied by all the N_{obs} obstacles. If it is ensured that the safe set $\mathcal{C}$ is *forward invariant* for all time t, i.e., if the trajectory of the payload-UAV system starts in $\mathcal{C}$, and forever be trapped inside $\mathcal{C}$, then obstacle avoidance is guaranteed. To ensure that the set $\mathcal{C}$ is forward invariant, construct the ith ECBF $h_i(.) : \mathbb{R}^3 \to \mathbb{R}$ as

$$h_i(x_c) = \left\|x_c - x_{\text{obs}_i}\right\|_2^2 - R_{o_i}^2,\ i = 1, \dots, N_{\text{obs}} \qquad (33)$$

where x_{obs_i} and R_{o_i} are the position and the radius of the ith obstacle, respectively, and x_c is the position of a point c_i on the surface of the convex hull around the payload-UAV system closest to the ith obstacle. It should be noted that the form of $h_i(.)$ is dependent on the ith obstacle shape.

Equation (33) holds for spherical obstacles only. However, (33) can be modified to include cylindrical pole obstacles (see Section IV-B), by extending z-axis linearly in both positive and negative axis, and the vectors in the (33) only include x- and y-coordinates. It can be easily verified that $h_i(.)$ satisfies the properties described in (31). At this point, an assumption is made that the convex hull (and thus, all the points on its surface) translates with a velocity equal to the translational velocity of the payload. This assumption is easily justified as long as the payload-UAV system remains close to the equilibrium point of (22). From (23), it can be inferred that the dynamics for the payload is actually a double integrator system, i.e., the control inputs appear only in the acceleration equations of the payload.[2] This results in the relative degree of $h_i(.)$ to be 2. Thus, (32) reduces to

$$\ddot{h}_i \geq -K \left[h_i \;\; \dot{h}_i\right]^T \tag{34}$$

where

$$\dot{h}_i = 2\left(x_c - x_{\text{obs}_i}\right)^T \left(\dot{x}_c - \dot{x}_{\text{obs}_i}\right) \tag{35a}$$

$$\ddot{h}_i = 2\left(x_c - x_{\text{obs}_i}\right)^T \left(\ddot{x}_c - \ddot{x}_{\text{obs}_i}\right) + 2\left\|\dot{x}_c - \dot{x}_{\text{obs}_i}\right\|_2^2 \tag{35b}$$

and $K \in \mathbb{R}^2$ is a row vector, chosen such that the poles of the system (34) are all negative. Substituting acceleration equations of the payload for $\ddot{x}_c$ and substituting (35) into (34) and rearranging the equations results in a set of N_{obs} constraints in Δu_k, of the form $\boldsymbol{A}_{\text{obs}}\Delta u_k \leq B_{\text{obs}}$.[3]

Thus, the obstacle avoidance controller can be designed as

$$\boxed{\begin{aligned} &\Delta u_k^* = \underset{\Delta u_k}{\arg\min}\ \tfrac{1}{2}\left\|\Delta u_k - \Delta\bar{u}_k\right\|^2_{\boldsymbol{Q}_{\text{obs}}} && (36a)\\ &\text{s.t} \quad \boldsymbol{A}_{\text{obs}}\Delta u_k \leq B_{\text{obs}} && (36b)\\ &\begin{bmatrix}\boldsymbol{I}_m\\ -\boldsymbol{I}_m\end{bmatrix}\Delta u_k \leq \begin{bmatrix}(u_{ub} - u_e)\\ (u_e - u_{lb})\end{bmatrix} && (36c)\end{aligned}}$$

where $\boldsymbol{Q}_{\text{obs}} > 0$ is a diagonal weighting matrix. The bounds on the control input are reapplied, along with the barrier function constraints. The optimal control law Δu_k^* produced by the IDC is thus equal to the optimal tracking control law $\Delta\bar{u}_k$ in the absence of obstacles, but while encountering the obstacles, the optimal control law Δu_k^* is different, ensuring that the set of constraints in (36c) is satisfied.

Note that the problem of (36) is a QP formulation, with quadratic cost function and linear affine constraints, defined similar to [18, Problem (26)]. The convergence proof for (36) follows [18, eq. (26)].

IV. SIMULATION RESULTS

In this section, the implementation details and simulation results are discussed to evaluate the performance of the IDC. The LMPC is implemented in C++ using the ACADO toolkit [19] code generator. *Multiple Shooting* discretization technique is employed to obtain a discrete system analogous to (23). The prediction horizon for the LMPC is chosen to be 20 time steps. The optimization problems in both (25) and (36) is solved using the *online active set strategy* [20] as implemented in the qpOASES library [21]. A major advantage of using active-set strategies is that the QP problem can be easily "warm-started," i.e., a good estimate of the optimal active set is used to start the algorithm. This is very useful in cases where a sequence of quadratic programs are to be solved, which is exactly what needs to be done in MPC problems. More details on this strategy and its time complexity can be found in [20]. To construct the convex hull, and find the point on the surface of the convex hull closest to the obstacle, *polytope distance* algorithms [22] are used as implemented in the CGAL library [23]. The advantage of detaching the obstacle avoidance controller from the MPC is that it enables the user to use preexisting, state-of-art MPC implementations that are highly optimized for nonlinear systems, and semidefinite hessian matrices that occur in the MPC optimization stage [19].

TABLE I
Parameter Values for the Payload-UAV System

	m	J_{xx}	J_{yy}	J_{zz}
Payload UAVs	3.1 0.7	0.29 0.01	0.29 0.01	0.55 0.01
		p_i		l_i
UAV1		$[0.5,\ 0.5,\ -0.25]^T$		3.2
UAV2		$[0.5,\ -0.5,\ -0.25]^T$		3.2
UAV3		$[-0.5,\ -0.5,\ -0.25]^T$		3.2
UAV4		$[-0.5,\ 0.5,\ -0.25]^T$		3.2

A. Numerical Results

The parameters considered for the payload-UAV system is shown in Table I, and the constants δ_ϕ, δ_θ, and δ_ψ of (24) are set to 5°. It is observed that assigning a value lower than 5° to these constants leads to infeasibility of (30), especially during obstacle avoidance.

1) *Trajectory Tracking:* For a numerical evaluation, four UAVs are considered to transport a rigid payload, as shown in Fig. 1. The desired trajectory for the payload is chosen as a figure-eight ∞ contour, as described by

$$r_{0\,d_x} = 6\sin(0.5t) \tag{37a}$$

$$r_{0\,d_y} = -6\sin(0.5t)\cos(0.5t) \tag{37b}$$

$$r_{0\,d_z} = -5u(t) \tag{37c}$$

where $u(t)$ is the unit step function. From Figs. 4 and 5, it can be seen that the LMPC controller successfully stabilizes the payload, while ensuring that it tracks the desired trajectory. The attitude of the payload is limited to within 5°during the abrupt take-off, but dies down quickly once stabilized. The yaw angles of the payload and UAVs are highly penalized to ensure that the optimization problem remains strictly convex [16]. A higher cost is placed on the attitude and the position of the payload, and a slightly lower cost is placed on the attitude of the links and UAVs in the weighting

[2] One can substitute the A and B matrices given in the footnote 1 in (23) to obtain the equations for the payload-UAV system.

[3] The analytical forms of the matrix $\boldsymbol{A}_{\text{obs}}$ and B_{obs} can be found here.

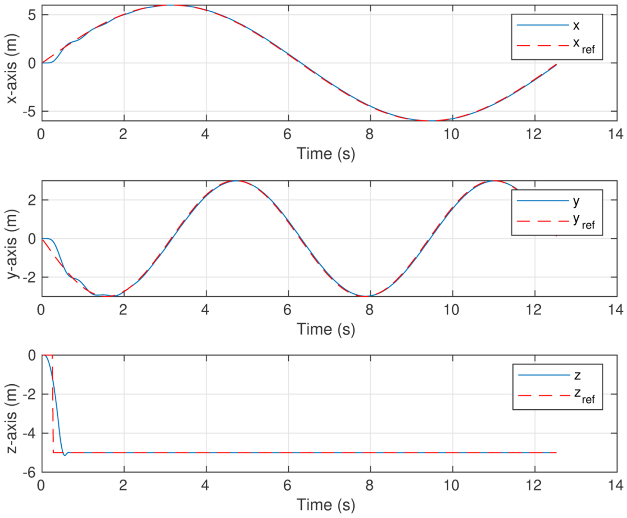


Fig. 4. Position of the payload is shown in solid line, and the desired trajectory is shown in dashed lines (no obstacles).

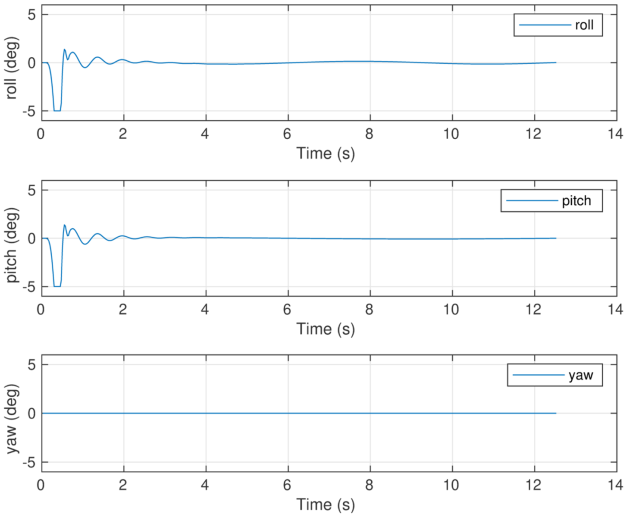
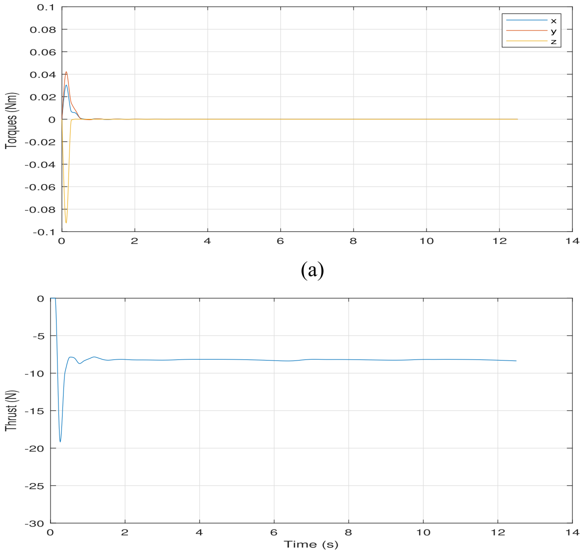


Fig. 5. Attitude of the payload (no obstacles).

matrix $\boldsymbol{Q}$ of (25). This ensures that the system can still make slight deviations from its equilibrium points, making the system slightly more flexible. The control inputs for UAV1 are shown in Fig. 6. The control inputs for the rest of the UAVs are available here. A snapshot of the 3-D simulation[4] is shown in Fig. 7, where the complete scenario is shown.

2) *Obstacle Avoidance:* The numerical simulation consists of both static and dynamic obstacles. It is assumed that the obstacles can be enclosed in sphere of radius R_o. For demonstration purposes, obstacle radius is set as 0.5 m and a safety margin is set as 0.5 m. Thus, $R_0 = 1$ m. The dynamic obstacle is modeled as a simple harmonic oscillator, oscillating about its mean position $(6, 0, -5)$m. The mean position of the dynamic obstacle coincides with a point on the desired trajectory, and the obstacle timing is set in such a way that it directly confronts the payload-UAV system at its mean position. The static obstacle is placed at $(-6, 0, -5)$m. Due to the dynamic nature of the obstacle, the constraint matrices $(\boldsymbol{A}_{\text{obs}}, B_{\text{obs}})$ of (36b) are time varying in nature.



(a)

(b)

Fig. 6. UAV1 inputs $\mathcal{F}_1$, τ_1 versus time.

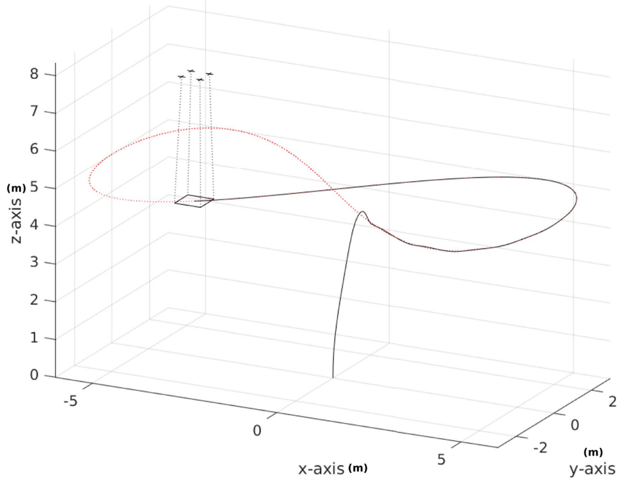


Fig. 7. Snapshot of the numerical simulation. The reference trajectory is shown in red dotted lines, and the payload trajectory is shown in black line. For the purpose of demonstration, the coordinates are converted to ENU from the NED coordinates.

It can be observed from Fig. 8 that the trajectory only changes along the z-axis, where the payload goes above the obstacle, as this is the most feasible solution. As seen in Fig. 9, the payload oscillates while avoiding the obstacles. This oscillation is bounded under a small deviation of 5° and occurs due to the fact that the payload-UAV system is avoiding the obstacle while in motion, without slowing down. Due to the high cost placed on the yaw angles of the payload and UAVs, there is almost no change in the yaw configuration of the system throughout the course of motion. The control input curves for UAV1 are shown

[4] The video demonstrations for both the numerical simulations as well as `Gazebo` simulations can be found at: https://youtu.be/AMVlMNYQCLw

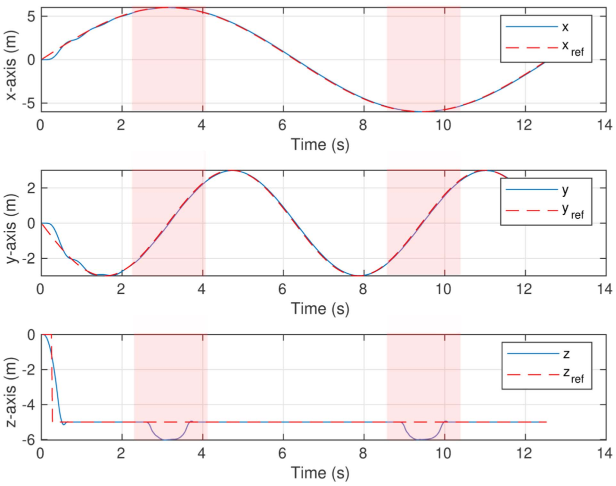


Fig. 8. Trajectory tracking with two surrounding obstacles. The instants of obstacle avoidance is shown in red shaded area.

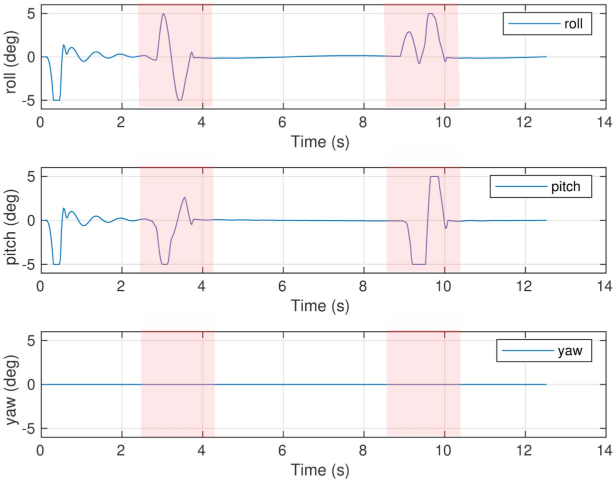
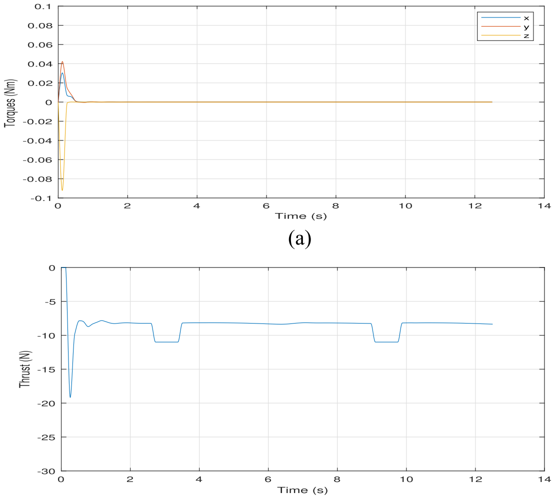


Fig. 9. Attitude of the payload with two surrounding obstacles with shared obstacle avoidance region.



(a)

(b)

Fig. 10. UAV1 inputs $\mathcal{F}_1$, τ_1 versus time with obstacles.

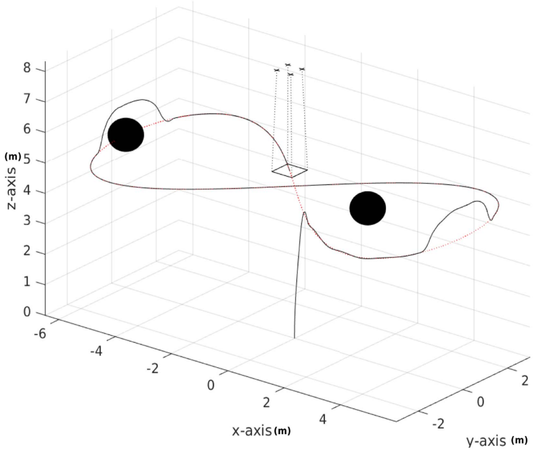


Fig. 11. Snapshot of the obstacle avoidance simulation. The sphere on the left is a static obstacle and the sphere on the right is a dynamic obstacle.

in Fig. 10. The control input curves for the rest of the UAVs are available here. A bird's eye snapshot of the 3-D obstacle avoidance simulation[5] is shown in Fig. 11. The spherical obstacle on the right oscillates along the x-axis, and periodically meets the payload-UAV system at the point $(6, 0, -5)$, while the spherical obstacle on the left remains stationary at the point $(-6, 0, -5)$.

B. Gazebo Simulation

The obstacle avoidance control algorithm is validated by testing it inside a high-fidelity simulation environment. The `Gazebo` simulator [24] is used to create an obstacle-avoidance environment course. For spawning the payload-UAV system, the entire model is written in a `urdf` file, and the `RotorS` package [25] is used for spawning four hummingbird drones. This is shown in Fig. 12. The control algorithm is implemented in a `C++` script, which communicates with the `Gazebo` simulator via the `ROS` library. The inputs to the model in the simulation are rotor revolution per minute values, which can be easily obtained from the thrust force and torques via a linear transformation matrix. The environment in `Gazebo` consists of many obstacles including walls and poles that are static and spherical obstacles that are dynamic. The desired trajectory that the payload must track goes in between the walls, and then, to the goal point behind the right-most sphere (see Fig. 12). This ensures that the payload-UAV system does not go all the way around the walls, and then, to the goal point.

The position of the leftmost and rightmost spherical obstacle is at $(22, 13, 5)$ and $(22, 37, 5)$, respectively. The cylindrical poles are placed at $(22, 20, 5)$, $(22, 26, 5)$, $(22, 30, 5)$ from left to right. It must be noted that the coordinate conventions in the `Gazebo` simulator is East-North-Up (ENU). Because of this, there is a coordinate conversion from NED to ENU at the interface of the `ROS C++` script and the `Gazebo` APIs.

It can be observed from Fig. 13 that the payload successfully avoids both the spherical obstacles, as well as the cylindrical pole obstacles. Unlike for spherical obstacles where the payload dodges by going above them, due to the

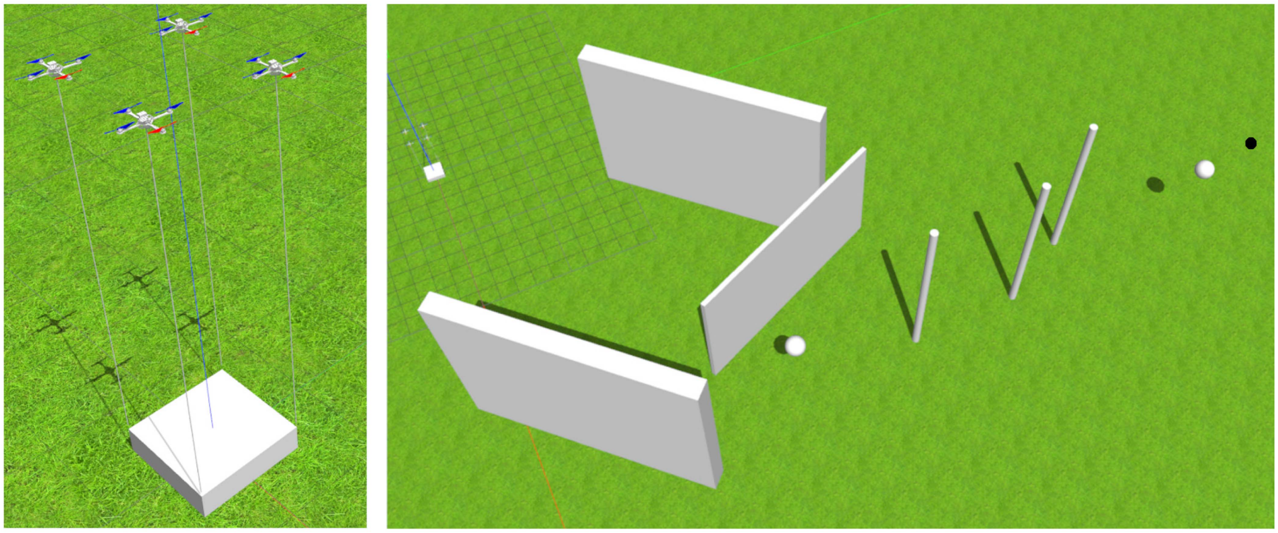


Fig. 12. Figure on the left shows the payload-UAV system, and the figure on the right shows the obstacle-avoidance environment course, that consists of walls, poles and spherical obstacles. The goal point for the payload is shown by a small black circle behind the rightmost sphere.

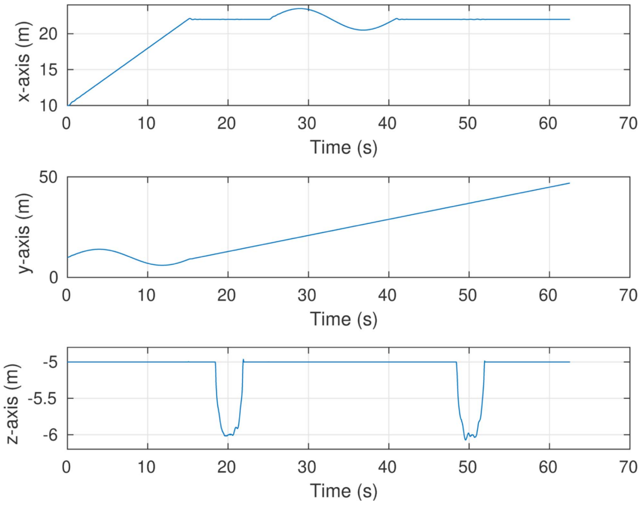


Fig. 13. Payload Trajectory for the obstacle environment course in `Gazebo`.

height of the pole obstacles, the payload goes around them by maintaining a safe distance from these poles. This is inferred better[5] from Fig. 14, where the purple line shows the trajectory history of the payload. The sphere near the walls is a static sphere, where as the sphere toward the end is a dynamic sphere oscillating about its mean position $(22, 37, 5)$. The control algorithm is able to run at a real-time frequency of 20 Hz, providing optimal revolution per minute values to the UAVs in the simulation.

C. Ablation Study

To evaluate the robustness of the IDC, some of the practical uncertainties are explored in the following paragraphs.

1) *Mass Uncertainty:* In practice, there is some uncertainty between the reported mass of the payload and the actual mass of the payload. A ±10% variation is considered here. From Table I, the value for the payload mass is taken as 3 kg during the controller design, but the payload mass in the plant equations is chosen to be either 2.7 kg or 3.3 kg. It is observed that the performance of the controller does not change, however the control effort $\|u\|_2$ required varies by ±5%. This is justified, as the UAVs need to compensate for the change in the weight, while meeting the tracking requirements.

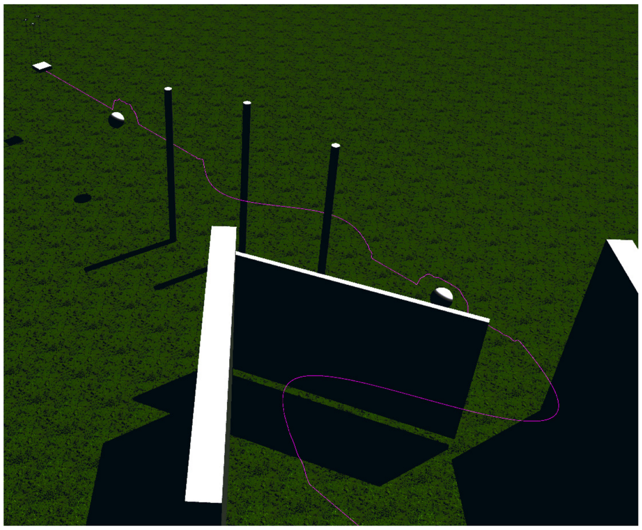
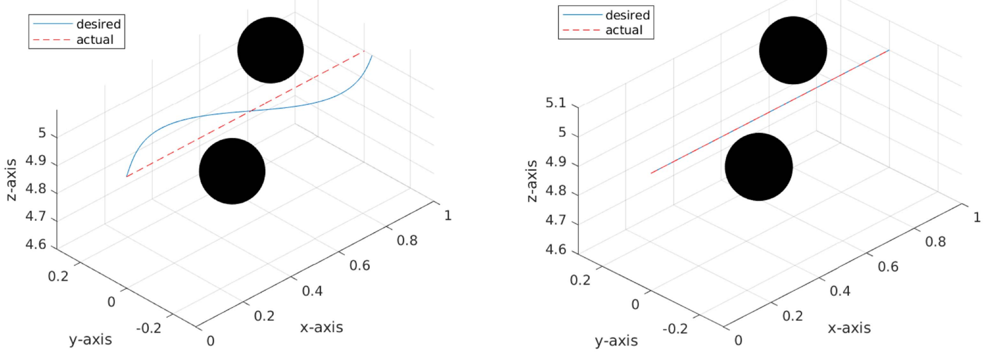


Fig. 14. Purple line shows the trajectory of the payload. The image is darkened to make the purple line more prominent and noticeable.

2) *Noisy Payload States:* The linear velocity and angular rates of the payload in (18) is subjected to zero mean Gaussian noise $\mathcal{N}(0, \sigma)$. This is done to mimic the actual payload state values obtained from a noisy rate sensor. For now, the UAV states are ignored, due to the presence of onboard estimation algorithms that can accurately infer the states. The goal is to analyze the robustness of the proposed IDC to the noisy state of the payload, i.e., to find the maximum value of the σ (m/s for linear velocity and rad/s for angular velocity), such that the payload oscillations remain under a certain threshold. The results



(a)

(b)

Fig. 15. (a) Trajectory of the payload (red dotted line) along with two spherical obstacles in case of potential fields approach. (b) Trajectory of the payload (red dotted line) along with two spherical obstacles in the case of barrier functions approach. The desired trajectory is shown in blue solid line.

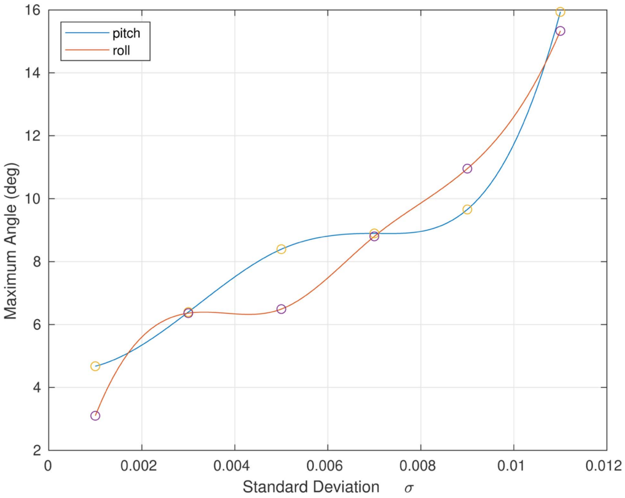


Fig. 16. Maximum roll and pitch angle versus the noise standard deviation σ (m/s and rad/s for linear and angular velocity).

are shown in Fig. 16. It can be observed that the maximum roll and pitch angles of the payload increase monotonically as σ increases. Beyond a standard deviation of 0.012, the obstacle avoidance controller becomes unstable, while the tracking controller continues to track the trajectory (with a noisy performance) when assessed independently. If the maximum threshold for payload oscillations is set at 10°, the standard deviation must not be above 0.008. In practice, most of the IMU's standard deviation falls below or occurs close the nominal value of 0.008.[5] It is also observed that the maximum roll and pitch angles occur only at the time of avoiding the obstacles.

3) *Relative Safety Margin:* The IDC ensures that there is minimum deviation from the reference trajectory while avoiding the obstacles. Earlier, a safety margin of 0.5 m was considered. It is observed that by changing the safety margin the controller performance does not change; it only determines the relative proximity between the payload-UAV system and the obstacle. In practice, the relative distance to the obstacles may be underestimated or overestimated. Thus, considering the right safety margin is crucial. Setting a higher safety margin may result in a higher deviation from the reference trajectory, which in some cases may render the tracking controller unstable, whereas setting a lower safety margin may be undesirable, as the payload-UAV may come too close to the obstacle, which could result in a collision. After various experiments, a thumb rule of choosing the safety margin equal to the obstacle radius is found to provide the best results.

D. Comparison With Potential Field Methods

In order to highlight the advantages of using barrier functions over potential field methods in the obstacle avoidance controller, a typical scenario of a cluttered environment is considered with two spherical obstacles as shown in Fig. 15. The payload must be transported through these obstacles considering the fact that there is sufficient space between them, and thus, requiring no deviation from the global trajectory.

The *repulsive potential fields* are designed as in [9] that originate from the obstacle's center of mass. Due to this reason, one can see that though there is sufficient space between the two obstacles, the trajectory of the payload deviates slightly due to the repulsive fields as seen in Fig. 15(a), which is unnecessary and requires more control effort. This issue is mitigated with the proposed IDC as seen in Fig. 15(b) that uses a safe convex hull around the payload-UAV system. As there is sufficient space for the system to pass through the gap between the obstacles, the barrier function constraints remain inactive, and thus, requires no deviation from the global trajectory.

V. CONCLUSION

In this article, an integrated decision control-based obstacle avoidance controller is presented to solve the problem of collaborative payload transportation in a cluttered

[5] A list of parameters for a few commonly used rate sensors are available at: https://github.com/rpng/kalibr_allan

environment. The IDC fuses the optimal tracking control provided by the MPC, with the safety-critical constraints provided by the ECBFs in an optimization framework. The structure of the ECBFs depend on the shape of the obstacle, which can be static or dynamic in nature. To ensure that the UAVs or the payload do not collide with the surrounding obstacles, a safe convex hull is computed around the entire system at each time instant, and the point on the convex hull closest to the surrounding obstacle is used to generate the ECBF constraints. Numerical simulations are conducted on the proposed controller to demonstrate its functionality, and eventually, the algorithm is deployed in a real time, high fidelity simulation using `Gazebo`. An ablation study is conducted to demonstrate the robustness of the IDC to handle practical unforeseen situations like payload mass uncertainties, noisy payload states, and choosing an optimal relative safety margin around the obstacles.

In the scope of future work, nonlinear control laws will be explored that can handle the problem of tracking and obstacle avoidance in a single framework. Moreover, external disturbances and state estimation techniques can be used to make the controller more robust to sensor noise and model uncertainties. Further, with the help of additional supporting mechanisms like rack-pinion joints, formation control of the UAVs for rigid payload can be made possible.

REFERENCES

[1] H. Y. Jeong, B. D. Song, and S. Lee, "The flying warehouse delivery system: A quantitative approach for the optimal operation policy of airborne fulfillment center," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7521–7530, Dec. 2021.

[2] Z. Xiaoning, "Analysis of military application of UAV swarm technology," in *Proc. 3rd Int. Conf. Unmanned Syst.*, 2020, pp. 1200–1204.

[3] R. D. Arnold, H. Yamaguchi, and T. Tanaka, "Search and rescue with autonomous flying robots through behavior-based cooperative intelligence," *J. Int. Humanitarian Action*, vol. 3, no. 1, pp. 1–18, 2018.

[4] C. Pei, F. Zhang, P. Huang, and H. Yu, "Trajectory planning for collaborative transportation by tethered multi-UAVs," in *Proc. IEEE Int. Conf. Real-time Comput. Robot.*, 2021, pp. 769–775.

[5] D. K. Villa, A. S. Brandao, and M. Sarcinelli-Filho, "A survey on load transportation using multirotor UAVs," *J. Intell. Robot. Syst.*, vol. 98, no. 2, pp. 267–296, 2020.

[6] K. Klausen, T. I. Fossen, T. A. Johansen, and A. P. Aguiar, "Cooperative path-following for multirotor UAVs with a suspended payload," in *Proc. IEEE Conf. Control Appl.*, 2015, pp. 1354–1360.

[7] Y. Liu, F. Zhang, P. Huang, and X. Zhang, "Analysis, planning and control for cooperative transportation of tethered multi-rotor UAVs," *Aerosp. Sci. Technol.*, vol. 113, 2021, Art. no. 106673.

[8] T. Lee, "Collision avoidance for quadrotor UAVs transporting a payload via Voronoi tessellation," in *Proc. Amer. Control Conf.*, 2015, pp. 1842–1848.

[9] I. H. B. Pizetta, A. S. Brandão, and M. Sarcinelli-Filho, "Avoiding obstacles in cooperative load transportation," *ISA Trans.*, vol. 91, pp. 253–261, 2019.

[10] I. H. B. Pizetta, A. S. Brandao, and M. Sarcinelli-Filho, "Control and obstacle avoidance for an UAV carrying a load in forestal environments," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2018, pp. 62–67.

[11] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1991, pp. 1398–1404.

[12] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 8129–8136.

[13] J. Gimenez, D. C. Gandolfo, L. R. Salinas, C. Rosales, and R. Carelli, "Multi-objective control for cooperative payload transport with rotorcraft UAVs," *ISA Trans.*, vol. 80, pp. 491–502, 2018.

[14] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Proc. Robot., Sci. Syst. IX*, 2013.

[15] T. Lee, "Geometric control of quadrotor UAVs transporting a cable-suspended rigid body," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1, pp. 255–264, Jan. 2018.

[16] J. Wehbeh, S. Rahman, and I. Sharf, "Distributed model predictive control for UAVs collaborative payload transport," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 11666–11672.

[17] P. Ru and K. Subbarao, "Nonlinear model predictive control for unmanned aerial vehicles," *Aerospace*, vol. 4, no. 2, 2017, Art. no. 31.

[18] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *Proc. Amer. Control Conf.*, 2016, pp. 322–328.

[19] B. Houska, H. Ferreau, and M. Diehl, "ACADO toolkit–An open source framework for automatic control and dynamic optimization," *Optimal Control Appl. Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[20] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.

[21] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, no. 4, pp. 327–363, 2014.

[22] S. Hert and S. Schirra, "3D convex hulls," in *CGAL User and Reference Manual*, 5.3.1 ed. CGAL Editorial Board, 2021. [Online]. Available: https://doc.cgal.org/5.3.1/Manual/packages.html#PkgConvexHull3

[23] The CGAL Project, *CGAL User and Reference Manual*, 5.3.1 ed. CGAL Editorial Board, 2021. [Online]. Available: https://doc.cgal.org/5.3.1/Manual/packages.html

[24] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, An open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, 2004, pp. 2149–2154.

[25] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "RotorS—A modular Gazebo MAV simulator framework," in *Robot Operating System (ROS): The Complete Reference*, vol. 1. Cham, Switzerland:Springer, 2016, pp. 595–625, doi: 10.1007/978-3-319-26054-9_23.

Nishanth Rao received the B.Tech. degree in electronics and communcation engineeing from PES University, Bangalore, India, in 2020. He is currently working toward the master's degree in robotics with the University of Pennsylvania, Philadelphia, PA, USA.

He is currently working as a Research Assistant under Dr. S. Sundaram with the Artificial Intelligence and Robotics Laboratory, Department of Aerospace Engineering, Indian Institute of Science, Bengaluru, India. His research interests include system identification and nonlinear safety-critical control and learning of multiagent systems.

Suresh Sundaram (Senior Member, IEEE) received the Ph.D. degree in aerospace engineering from the Indian Institute of Science, Bengaluru, India, in 2005.

From 2010 to 2018, he was an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He is currently an Associate Professor with the Department of Aerospace Engineering, Indian Institute of Science. His research interests include flight control, unmanned aerial vehicle design, machine learning, optimization, and computer vision.