Abstract—Aerial package transportation often requires complex spatial and temporal specifications to be satisfied in order to ensure safe and timely delivery from one point to another. It is usually efficient to transport versatile payloads using multiple UAVs that can work collaboratively to achieve the desired task. The complex temporal specifications can be handled coherently by applying Signal Temporal Logic (STL) to dynamical systems. This paper addresses the problem of waypoint navigation of a multi-UAV payload system under temporal specifications using higher-order time-varying control barrier functions (HOCBFs). The complex nonlinear system of relative degree two is transformed into a simple linear system using input-output feedback linearization. An optimization-based control law is then derived to achieve the temporal waypoint navigation of the payload. The controller’s efficacy and real-time implementability are demonstrated by simulating a package delivery scenario inside a high-fidelity Gazebo simulation environment.

I. INTRODUCTION

Advancements in multi-UAV system research and technologies have led to the widespread use of multi-UAV systems in the aerial transportation of payloads from one point to another. It is evident that using multiple UAVs to deliver versatile payloads is more efficient than utilizing a single UAV of varying lift capacities. In addition, payloads need to be delivered to specific locations under some time constraints. For example, in the air delivery of packages, it is crucial to not only deliver the package at the correct location, but the package must also reach within the specified time. Thus it is essential to design control algorithms for multi-UAV payload systems that follow certain temporal specifications while enabling safe payload transportation.

Most of the literature focuses on developing trajectory-tracking controllers for multi-UAV payloads without considering temporal specifications. In [1], the authors propose a trajectory tracking controller using geometric control for a UAV fleet carrying the payload using cable suspension. In [2], a distributed model predictive control algorithm is developed for the purpose of trajectory tracking of payload using rigid link payload suspension. The authors in [3] develop an adaptive controller for a cable-suspended payload with user-specified safety constraints. For a rigid link suspended payload, a feedback linearized nonlinear controller is presented in [4].

Pushpak Jagtap is with the Robert Bosch Center for Cyber-Physical Systems, Nishanth Rao is with the Department of Aerospace Engineering, and Suresh Sundaram is with the Robert Bosch Center for Cyber-Physical Systems and the Department of Aerospace Engineering at the Indian Institute of Science, Bangalore, India. nishanthrao, vssuresh, pushpak@iisc.ac.in

Tasks that require complex spatial and temporal requirements can be handled effectively by applying Signal Temporal Logic (STL) [5] to dynamical systems. The authors in [6] use control barrier functions [7], [8] to design control inputs enforcing a class of STL specifications. In [9], a dynamically feasible STL-based planning and control algorithm is designed for a single UAV system that satisfies complex spatial and temporal requirements. An extension to multi-UAV fleets (without payload) is proposed in [10] that generates continuous trajectories for multiple UAVs to meet the temporal requirements.

While there have been works on designing controllers for multi-agent systems to satisfy STL specifications [11], there has been no work on developing controllers to satisfy complex STL specifications, particularly for a multi-UAV system carrying a payload. One of the reasons is that the dynamics of such a system is highly coupled and nonlinear in nature. This can lead to a complex state-space representation with large state and control vector dimensions. The curse of dimensionality can thus make it difficult to control such a system under STL specifications.

This paper focuses on developing a controller for the multi-UAV payload system to enforce the complex STL specifications using Higher-Order Time-Varying Control Barrier Functions (HOCBFs). The model complexity of the system is reduced by the application of input-output feedback linearization, which results in a linear system with a relative degree equal to 2. The STL specifications are captured using time-Varying HOCBFs, and an optimization-based control law is proposed. To validate the efficacy and real-time implementability of the proposed controller, a simulation environment is developed using the Gazebo simulator for the purpose of simulating a package delivery scenario under time specifications.

Section II discusses the necessary preliminaries and notations used in this paper. Section III proposes a control law for temporal waypoint navigation of the multi-UAV payload system. In Section IV, the proposed control law is validated in a high-fidelity Gazebo simulator that simulates a package-delivery scenario. Section V concludes the proposed work.

II. PRELIMINARIES

In this paper, scalars and vectors are denoted by non-bold letters $a$, whereas matrices are denoted by bold letters $A$. We use $\mathbb{R}$, $\mathbb{R}^+$, and $\mathbb{R}^+_0$ to denote real, positive real, and nonnegative real numbers, respectively. $\mathbb{R}^n$ denotes an $n$-dimensional vector with real elements and $\mathbb{R}^{m \times n}$ denotes an $m \times n$ matrix with real elements. The vector $k = [0 \ 0 \ 1]^T \in \mathbb{R}^3$.
represents the unit z-axis. The matrix \( I_n \in \mathbb{R}^{n \times n} \) denotes the \( n \times n \) identity matrix. The Moore-Penrose pseudoinverse of a general matrix \( A \in \mathbb{R}^{m \times n} \) is denoted by \( A^+ \in \mathbb{R}^{n \times m} \). A real-valued \( n \times n \) dimensional signal is denoted by \( x : \mathbb{R}^+_0 \rightarrow \mathbb{R}^n \).

The notation \( x \) is abused by also referring to the trajectory of a dynamical system with \( n \) states. Thus, the meaning of \( x \) is assumed to be clear from the context. A function \( \alpha : \mathbb{R}^+_0 \rightarrow \mathbb{R}^+_0 \) is a class \( K \) function if it is continuous and strictly increasing, with \( \alpha(0) = 0 \). The Lie derivative of a vector field \( b \) along the vector field \( a \) is denoted by \( \mathcal{L}_a b = a \cdot \frac{\partial b}{\partial x} \). The function composition operator \( \circ \) is defined as \( f \circ g = f(g(x)). \)

### A. Payload-UAV System Description

There are, in general, \( N \) UAVs connected to the rigid massless rods, each of length \( l_i \), via spherical joints. These rods are rigidly connected to the payload and thus always remain vertical. The location of the centre of mass of the payload and the \( i^{\text{th}} \) UAV is denoted by \( r_{0i} \in \mathbb{R}^3 \) and \( r_{i} \in \mathbb{R}^3 \) respectively. The vector \( \rho_i \in \mathbb{R}^3 \) denotes the position of the attachment between the rod corresponding to the \( i^{\text{th}} \) UAV and the payload with respect to the centre of mass of the payload. The mass of the payload and the \( i^{\text{th}} \) UAV is denoted by \( m_0, m_i \in \mathbb{R}^+ \) respectively. The inertia matrix of the payload and the \( i^{\text{th}} \) UAV is indicated by \( J_0, J_i \in \mathbb{R}^{3 \times 3} \) respectively. The rotation matrix \( R_0 \in SO(3) \) and \( R_i \in SO(3) \) denotes the orientation of the payload and the \( i^{\text{th}} \) UAV respectively. This rotation matrix converts a vector in the body-fixed frame to the inertial frame. The payload-UAV system is modelled in the standard North, East and Down frame (NED), with the positive unit z-axis pointing downwards. The total thrust force produced by all the motors of the \( i^{\text{th}} \) UAV is denoted by \( F_i \in \mathbb{R}^+ \) and is related to the force vector \( F_i \in \mathbb{R}^3 \) as \( F_i = F_i R_i k \). The motors produce a torque vector in the body frame of the \( i^{\text{th}} \) UAV denoted by \( \tau_i = [\tau_{x_i}, \tau_{y_i}, \tau_{z_i}]^T \in \mathbb{R}^3 \). Thus, the overall control input for the multi-UAV system is \( \{F_i, \tau_i\} \) with \( i \in \{1, \ldots, N\} \).

### B. Payload-UAV Dynamics and State-space Representation

Using the description of the payload-UAV system discussed in Section II-A, the equations of motion for the multi-UAV payload system (shown in Fig. 1) are presented as follows. As the rigid links are always vertical, the position of the \( i^{\text{th}} \) UAV can be inferred as:

\[
\vec{r}_i = \vec{r}_0 + R_0 (\rho_i - l_i k).
\]

The kinematic and dynamic equations are given by:

\[
\begin{align*}
\dot{r}_0 &= R_0 v_0,
\dot{R}_0 &= R_0 \omega^X_0,
\dot{r}_i &= r_0 + R_0 \omega^X_0 (\rho_i - l_i k),
\dot{R}_i &= R_i \dot{\omega}^X_i,
\end{align*}
\]

where \( v_0 \) and \( \omega_0 \) are the translational and rotational speeds of the payload, \( \dot{\omega}^X_i \) is the angular velocity of the \( i^{\text{th}} \) UAV, and \( \dot{\rho}_i \) is the angular velocity of the \( i^{\text{th}} \) rod. The nonlinear state equations (2) - (8) can be obtained in the form \( \dot{x} = f(x, u) \) by rearranging (2) - (8), where the control input to the system is \( u \in \mathbb{R}^{12} \). The nonlinear state-space equations can be obtained in the form \( \dot{x} = f(x, u) \) by rearranging the kinematic equations (2) - (5):

\[
\begin{align*}
\dot{v}_0 &= \mathbf{J} \mathbf{\zeta},
\dot{\omega}_0 &= \mathbf{\xi}.
\end{align*}
\]

Fig. 1. A schematic diagram illustrating the payload transport using multiple UAVs. The support-frame together with the payload (shown lightly as a black cylinder) is modelled as a cuboid.

\[
m_T (\dot{v}_0 + \omega^X_0 v_0) + \sum_{i=1}^{N} m_i (-\rho^X_i \dot{\omega}_0 + (\omega^X_0)^2 \rho_i) = R_0^T \left[ m_T g k + \sum_{i=1}^{N} F_i \right],
\]

\[
\sum_{i=1}^{N} m_i \rho^X_i (\dot{v}_0 + \omega^X_0 v_0) + \mathbf{J}_0 \dot{\omega}_0 + \mathbf{J}_0 \dot{\omega}_0
\]

\[
= \sum_{i=1}^{N} \rho^X_i R_0^T (F_i + m_i g k),
\]

\[
\mathbf{J}_i \dot{\omega}_i + \omega^X_i \mathbf{J}_i \dot{\omega}_i = \tau_i,
\]

where the skew-symmetric operator \((\cdot)^\times : \mathbb{R}^3 \rightarrow SO(3)\) denotes the hat map which maps a vector to a skew-symmetric matrix, \( \mathbf{v}_0 \in \mathbb{R}^3 \) denotes the linear velocity of the payload in the payload-fixed frame, \( \omega_0 \in \mathbb{R}^3 \) and \( \dot{\omega}_0 \in \mathbb{R}^3 \) denotes the angular velocities of the payload and the \( i^{\text{th}} \) UAV, respectively. The quantity \( m_T = m_0 + \sum_{i=1}^{N} m_i \) is the total combined mass, and \( \mathbf{J}_0 = \left( \mathbf{J}_0 - \sum_{i=1}^{N} m_i (\rho^X_i)^2 \right) \) is the apparent moment of inertia of the payload. For a detailed derivation, please refer to [4]. In order to obtain a state-space representation of the payload-UAV dynamic model, define the state vector as:

\[
x = \begin{bmatrix} v_0^T & \mathbf{J}_0^T & \omega^X_0^T \end{bmatrix}^T, \quad i \in \{1, \ldots, N\},
\]

where \( \Theta_0, \Theta_i \in \mathbb{R}^3 \) are the ZYX Euler angle characterization of the attitudes \( R_0, R_i \) respectively. The nonlinear state-space equations can be obtained in the form \( \dot{x} = f(x, u) \) by rearranging (2) - (8), where the control input to the system is \( u = \{F_i, \tau_{x_i}, \tau_{y_i}, \tau_{z_i}\}^T \in \mathbb{R}^{12}, \quad i = 1, \ldots, N \) and are given below (in addition to the kinematic equations of (2) - (5)):

\[
\begin{align*}
\dot{v}_0 &= \mathbf{J} \mathbf{\zeta},
\dot{\omega}_0 &= \mathbf{\xi}.
\end{align*}
\]
where $\zeta \in \mathbb{R}^{6 \times 6}$, $\xi \in \mathbb{R}^{6 \times 1}$ and are given as follows:

$$
\zeta = \begin{bmatrix}
\frac{mv_I J_3}{\sum_{i=1}^{N} m_i \rho_i} - \sum_{i=1}^{N} m_i \rho_i x_i
\end{bmatrix}^{-1}
\begin{bmatrix}
P_{11} & P_{12} \\
P_{21} & P_{22}
\end{bmatrix}
$$

(11)

$$
\xi = \begin{bmatrix}
-\frac{mv_I^2 \omega_0 - \sum_{i=1}^{N} m_i (\omega_0^2) \rho_i + mv_I^2 R_{i1}^2 k + \sum_{i=1}^{N} R_{i1}^2 F_i}{-\omega_0^2 J_0 \omega_0 - \sum_{i=1}^{N} m_i \rho_i \omega_0^2 v_0 + \sum_{i=1}^{N} \rho_i R_{i1}^2 (F_i + m_i g_k)}
\end{bmatrix}
$$

(12)

An input-output feedback linearization of the position dynamics of the payload can be done, with relative degree 2, as seen from (10). Thus, we obtain a second-order linear system as given by:

$$
\begin{bmatrix}
\dot{y} \\
\ddot{y}
\end{bmatrix} = \begin{bmatrix}
\dot{v}_1 \\
\dot{v}_2 \\
\dot{v}_3
\end{bmatrix} = v,
$$

(13)

where $v$ is a new transformed control input. A detailed derivation of this procedure is provided in [4].

C. Signal Temporal Logic (STL)

Signal Temporal Logic (STL) [5] provides a formal framework to capture high-level specifications that can handle spatial, temporal, and logical constraints. It consists of a set of predicates $\mu$ that are evaluated based on their corresponding predicate function $b : \mathbb{R}^n \rightarrow \mathbb{R}$ as $\mu := \{ True \text{, if } h(x) \geq 0 \}$, $\mu := \{ False \text{, if } h(x) < 0 \}$. The syntax for an STL formula $\phi$ is given by:

$$
\phi := True \ | \ \neg \phi \ | \ \phi_1 \land \phi_2 \ | \ \phi_1 U[\alpha,\beta] \phi_2 \ | \ F[\alpha,\beta] \phi \ | \ G[\alpha,\beta] \phi,
$$

(14)

where $\alpha, \beta \in \mathbb{R}^+_0$ with $a \leq b$, $\phi_1$ and $\phi_2$ are STL formulas, $F$ denotes the eventually operator, $G$ denotes the always operator and $U$ denotes the until operator, each of which is defined below. The relation $(x, t) \models \phi$ indicates that the signal $x : \mathbb{R}^+_0 \rightarrow \mathbb{R}^n$ satisfies the STL formula $\phi$ at time $t$.

Definition 1. STL Semantics[5]: The STL semantics for a signal $x$ are recursively defined as follows:

$$(x, t) \models \mu \iff h(x) \geq 0$$

$$(x, t) \models \neg \phi \iff \neg ((x, t) \models \phi)$$

$$(x, t) \models \phi_1 \land \phi_2 \iff (x, t) \models \phi_1 \land (x, t) \models \phi_2$$

$$(x, t) \models \phi_1 U[\alpha,\beta] \phi_2 \iff \exists \exists t_1 \in [t + \alpha, t + \beta] \ s.t. (x, t_1) \models \phi_2$$

$$\land \forall t_2 \in [t, t_1), (x, t) \models \phi_1$$

$$(x, t) \models F[\alpha,\beta] \phi \iff \exists t_1 \in [t + \alpha, t + \beta] \ s.t. (x, t_1) \models \phi$$

$$(x, t) \models G[\alpha,\beta] \phi \iff \forall t_1 \in [t + \alpha, t + \beta], (x, t_1) \models \phi.$$  

D. Higher-order Time-varying Control Barrier Functions

Consider a nonlinear control-affine system:

$$\dot{x} = f(x) + g(x)u,$$

(16)

where $x \in \mathbb{R}^n$ is the state of the system and $u \in \mathbb{R}^m$ is the control input to the system. Let the relative degree of the system be $\eta > 1$. A time-varying set $C(t)$ is forward invariant if for a given control law $u$, there exists a unique solution $x(t)$ to the system (16) with $x(t_0) = x_0 \in C(t_0)$ such that $\forall t \in [t_0, t_f], x(t) \in C(t)$. For nonlinear affine systems (16) with relative degree $\eta > 1$, higher order control barrier functions (HOCBFs)[12][13] can be used to guarantee the forward invariance of the set $C(t)$.

Definition 2. HOCBF[13] A function $b : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}$ is a candidate higher order control barrier function (HOCBF) of relative degree $\eta$ for the system defined in (16) if there exists differentiable class $K$ functions $\alpha_i, i \in \{1, 2, \ldots, \eta\}$ such that

$$
\sup_{u \in \mathcal{U}} \left[ \mathcal{L}_b^\eta b(x, t) + \mathcal{L}_b^{\eta-1} b(x, t) u + \frac{\partial \eta b(x, t)}{\partial \eta} + \alpha_{\eta-1}(x, t) \right] \geq 0,
$$

(17)

where the functions $\gamma_i$ is defined as:

$$
\gamma_i(x, t) = \gamma_{i-1}(x, t) + \alpha_i(\gamma_{i-1}(x, t)), \quad i = \{1, \ldots, \eta\}
$$

(18)

with $\gamma_0$ chosen as $\gamma_0(x, t) = b(x, t)$. Define a sequence of sets $C_i(t)$ associated with the functions $\gamma_i$ as follows:

$$
C_i(t) = \{ x \in \mathbb{R}^n, \gamma_i(x, t) \geq 0 \}
$$

(19)

such that $C(t) = C_1(t) \cap C_2(t) \cap \ldots \cap C_\eta(t)$.

Theorem 1. (see [13, 14]) The set $C(t) = C_1(t) \cap \ldots \cap C_\eta(t)$ is forward invariant for the system defined in (16), if $b(x, t)$ is a HOCBF according to Definition 2.

III. TEMPORAL WAYPOINT NAVIGATION OF MULTI-UAV PAYLOAD SYSTEM

A. HOCBFs for STL Tasks

This subsection discusses how higher-order time-varying control barrier functions can be used for STL tasks. Consider the following STL fragment:

$$
\psi := True \ | \ \neg \mu \ | \ \psi_1 \land \psi_2,
$$

(20)

$$
\phi := G[\alpha,\beta] \psi \ | \ F[\alpha,\beta] \psi \ | \ \psi_1 U[\alpha,\beta] \psi_2 \ | \ \phi_1 \land \phi_2,
$$

(21)

where $\psi_1, \psi_2$ are STL formulas for class $\psi$ functions defined in (20) and $\phi_1, \phi_2$ are STL formulas for class $\phi$ functions defined in (21). The barrier functions are required to be less than or equal to their corresponding predicate function between the desired time interval. For tasks of the form $F[\alpha,\beta] \mu$ with predicate function $h(x)$, choose $b(x, t)$ such that $b(x, t') \leq h(x)$ for some $t' \in [a, b]$. For tasks of the form $F[\alpha,\beta] \mu$ with the predicate function $h(x)$, choose $b(x, t)$ such that $b(x, t') \leq h(x)$ for all $t' \in [a, b]$. When there is a conjunction of STL formulas as in (20) or (21), the following smooth-approximation for the minimum operator is used for a set of $M$ barrier functions with $j = \{1, \ldots, M\}$:

$$
\min_j b_j(x, t) \approx -\ln \left( \sum_j \exp (-b_j(x, t)) \right).
$$

(22)
Thus, for tasks of the form $F_{[a,b]}\psi$ where $\psi := \mu_1 \land \mu_2$ with the predicates $h_1(x)$ and $h_2(x)$ respectively, choose $b(x,t) := -\ln(\exp(-b_1(x,t)) + \exp(-b_2(x,t)))$ such that $h_1(x) \leq b_1(x,t)$ and $h_2(x) \leq b_2(x,t)$ for some $t' \in [a,b]$. Similarly, for tasks of the form $G_{[a,b]}\psi$ with the predicates $h_1(x)$ and $h_2(x)$ respectively, choose $b(x,t) := -\ln(\exp(-b_1(x,t)) + \exp(-b_2(x,t)))$ such that $h_1(x) \leq b_1(x,t)$ and $h_2(x) \leq b_2(x,t)$ for all $t' \in [a,b]$, and so on. More details on using barrier functions for STL properties can be found in [6].

The STL controller can be formulated as a convex quadratic programming (CQP) problem as follow:

$$\min_u u^T Qu \quad \text{s.t.} \quad Pu \leq H.$$  \hfill (23a)

At each time step, the state vector $x$ is obtained and the vectors $P, H$ are computed using (17), which results in a linear inequality $Pu \leq H$.

**Example 1.** In this example, the calculation of $P, H$ vectors for a system with relative degree $\eta = 2$ is shown. One can simply expand (23) for $\eta = 2$ to obtain as follows:

$$- (L_0\dot{L}_f b) + \frac{\partial^2 \dot{b}}{\partial t^2} + 2bL_f \dot{b} + 2b\frac{\partial b}{\partial t} + (L_f b)^2 + \frac{\partial b}{\partial t}^2 + b^4 + 2(L_f b) \frac{\partial b}{\partial t} + 2b^2L_f \dot{b} + 2b^2\frac{\partial b}{\partial t} \geq 0$$

where the class $\mathcal{K}$ functions $\alpha_i$’s are chosen as $\alpha_i(s) = s^2$ for all $s \in \mathbb{R}_0^+$ and $\gamma_0 = b$. Thus, at every time step, the CQP of (23) is solved to obtain the optimal control input $u$ to the system.

**B. Temporal Waypoint Navigation Problem**

Consider a series of $M$ waypoints. During the course of waypoint navigation, it is, in general, desirable that the payload reaches a particular waypoint location $\Lambda_j \in \mathbb{R}^3$ where $j = \{1, \ldots, M\}$ within a certain threshold radius $R_0 \in \mathbb{R}$, while meeting some time specifications before moving on to the next waypoint. This requires that the controller complies with both the spatial as well as the temporal requirement of the task at hand. For this purpose, a set of time-varying HOCBFs can be constructed to meet the required specifications. For the payload-UAV system, the structure of the HOCBFs is chosen as follows:

$$b_j(x,t) = \Gamma_j(t) - \left(\|r_0 - \Lambda_j\|_2\right)^2,$$

where $\Gamma_j$ is a temporal function. For example, without any loss of generality, let the payload-UAV system start at the origin. The specifications to be met are as follows:

$$\phi := \phi_1 \land \phi_2$$

where $\phi_1 := F_{[0,14]} \left(\|r_0 - [2\ 2\ 2]^T\|_2 \leq 0.1\right)$ and $G_{[14,25]} \left(\|r_0 - [2\ 2\ 2]^T\|_2 \leq 3\right)$. This means that the payload must reach the waypoint location $\Lambda = [2\ 2\ 2]^T$ within a radius of $R_0 = 0.1m$ in the time interval $t \in [0,14]$.

Once it reaches the specified waypoint, it must remain at the waypoint location $\Lambda = [2\ 2\ 2]^T$ within a radius of $R_0 \approx 3m$ in the time interval $t \in [14, 25]$. For this specification, the predicate function $h_1$ corresponding to $\phi_1$ is given by $h_1(x) = \left(0.1 - \|r_0 - [2\ 2\ 2]^T\|_2\right)^2$. Similarly, the predicate function $h_2$ corresponding to $\phi_2$ is given by $h_2(x) = \left(3 - \|r_0 - [2\ 2\ 2]^T\|_2\right)^2$. The barrier function $b_1$ corresponding to $\phi_1$ must satisfy the requirement $b_1(x,t) \leq h_1(x)$ for some $t' \in [0,14]$. Similarly, the barrier function $b_2$ corresponding to $\phi_2$ must satisfy the requirement $b_2(x,t) \leq h_2(x)$ for all $t' \in [14,25]$. The candidate barrier functions can be chosen as follows:

$$b_1(x,t) = \left(50.1 - \frac{50}{14}\right) - \|r_0 - [2\ 2\ 2]^T\|_2$$

and

$$b_2(x,t) = \left(347.93 \cdot e^{-0.418t} + 2\right) - \|r_0 - [2\ 2\ 2]^T\|_2$$

This can then be extended to a series of $M$ waypoints that require both spatial as well as temporal specifications. The final barrier function $b(x,t)$ obtained from (26c) can then be plugged into (24) to generate the HOCBF constraints. Finally, the CQP problem defined in (23) can be solved to obtain control input $v$ in an online manner for the feedback-linearized system defined in (13) that ensures that the payload-UAV system meets the temporal waypoint navigation specifications. This optimal control input $v$ must then be converted to the original control input to the nonlinear system. This process is explained in detail in [4].

**IV. Simulation Results**

In order to validate the proposed STL controller in (23), a high-fidelity Gazebo simulation is conducted. The rotorS MAV simulator package [15] is used to spawn four UAVs that are connected to a rectangular payload via four rigid links. The software stack consists of custom Gazebo plugins that simulate GPS and IMU sensors. The entire payload-UAV model is spawned using .xacro files, and the proposed control algorithm is implemented inside a ROS C++ script. The control algorithm runs at a sampling rate of 50Hz. The simulation is conducted on a 3.2GHz Intel i7 processor with 8GB RAM and NVIDIA 1050Ti graphics driver.

The multi-UAV payload system is shown in Fig. 3. It consists of four Hummingbird UAVs that are connected to the four vertical rods via spherical joints (modeled as two revolute joints in .xacro file). The vertical rods are rigidly connected to the payload and, thus, cannot move. The payload carries two packages that are to be delivered at specific locations with certain time specifications.

Consider a typical package delivery scenario where the UAVs must collaboratively transport the payload carrying
Fig. 2. Figure shows the simulation environment in the Gazebo simulator. The mailboxes represent the location at which the package must be delivered. The multi-UAV payload system starts at the origin carrying two packages. Once both packages are delivered, the payload must reach the hub (shown as a white cylindrical platform) for refilling purposes.

Fig. 3. Figure shows 4 Hummingbird UAVs connected to the vertical links through spherical joints. There are two packages on the payload, which the UAVs must transport to the required destination under certain time specifications.

multiple packages from one point to another. For the sake of simulation purposes, consider that there are two packages on the payload that must be delivered at different locations with some temporal specifications. The payload must arrive at \( \Lambda_1 = (10, 10, 5) \) to deliver the first package between the time interval \([0, 18]\) and stay there till \( t = 30s \) within a tolerance radius of \( R_0 = 1.0 \). The payload must then arrive at \( \Lambda_2 = (25, 10, 5) \) to deliver the second package between the time interval \([30, 48]\) and stay there till \( t = 60s \) within the tolerance radius of \( R_0 = 1.0 \). Finally, the empty payload must land at the hub located at \( \Lambda_3 = (40, 20, 1) \) for refilling within the time interval \([60, 78]\) and stay there till \( t = 90s \) within the tolerance radius of \( R_0 = 1.0 \). These requirements can be captured in the following STL specification \( \phi := \phi_1 \land \phi_2 \land \phi_3 \land \phi_4 \) where,

\[
\phi_1 := F_{[0,18]} \left( \left\| r_0 - [10 10 5]^T \right\|_2 \leq 1.0 \right) \quad \land \quad G_{[17,30]} \left( \left\| r_0 - [10 10 5]^T \right\|_2 \leq 1.0 \right), \tag{27}
\]

\[
\phi_2 := F_{[30,48]} \left( \left\| r_0 - [25 10 5]^T \right\|_2 \leq 1.0 \right) \quad \land \quad G_{[47,60]} \left( \left\| r_0 - [25 10 5]^T \right\|_2 \leq 1.0 \right), \tag{28}
\]

\[
\phi_3 := F_{[60,78]} \left( \left\| r_0 - [40 20 1]^T \right\|_2 \leq 1.0 \right) \quad \land \quad G_{[77,90]} \left( \left\| r_0 - [40 20 1]^T \right\|_2 \leq 1.0 \right), \tag{29}
\]

\[
\phi_4 := G_{[0,90]} \left( \left\| r_0 \right\|_\infty \leq 50 \right). \tag{30}
\]

The specification \( \phi_4 \) in (30) ensures that the payload always remains inside a safe cuboid of side \( 50m \) centred at the origin.

The entire simulation environment is shown in Fig. 2. It consists of two mailboxes positioned at \( \Lambda_1 \) and \( \Lambda_2 \), respectively. While delivering the packages, the payload’s centre of mass must be within the green translucent cylinders that have a radius equal to \( R_0 \), which is the tolerance radius in the horizontal plane. Once delivered, the UAVs must land the payload on the white circular hub located at \( \Lambda_3 \). The simulation video is available here\(^1\).

\(^1\)The video demonstration is available on this link if the pop-up doesn’t work: https://youtu.be/T1MYlKHaMkw

| Parameters for the multi-UAV payload system |
|--------------------------------------------|
| \( m \) | \( J_{xx} \) | \( J_{yy} \) | \( J_{zz} \) |
| Payload | 1.0 | 0.556 | 0.556 | 0.556 |
| UAVs | 0.68 | 0.029 | 0.029 | 0.055 |
| \( \rho_i \) | \( l_i \) |
| UAV1 | \([-0.25, 0.25, 0.125]^T\) | 3.2 |
| UAV2 | \([-0.25, -0.25, 0.125]^T\) | 3.2 |
| UAV3 | \([1.0, 0.25, 0.125]^T\) | 3.2 |
| UAV4 | \([-0.25, 0.25, 0.125]^T\) | 3.2 |

---

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
The payload trajectory in the individual axis is shown in Fig. 4 along with the time envelopes of the respective temporal functions $\Gamma_j(t)$. It can be seen that the controller satisfies all the temporal specifications while navigating through each of the waypoints. The values for time-varying barrier function $b(x, t)$ are provided in Fig. 5.

VI. CONCLUSIONS

In this paper, the problem of temporal waypoint navigation for multi-UAV payload systems is addressed. The $N$ UAVs are attached to rigid rods via spherical joints. These rods are connected to the payload rigidly and thus remain vertical always. This system’s complex nonlinear state-space equations are simplified by applying input-output feedback linearization to the payload’s translational dynamics. The Signal Temporal Logic Specifications are captured using higher-order time-varying control barrier functions to formulate an optimization-based control law. The controller’s efficacy and real-time implementability are demonstrated by simulating a package delivery scenario in a high-fidelity Gazebo simulator.

VI. ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support from the Nokia CSR grant on Network Robotics.

REFERENCES

[1] T. Lee, “Geometric control of quadrotor UAVs transporting a cable-suspended rigid body,” IEEE Transactions on Control Systems Technology, vol. 26, no. 1, pp. 255–264, 2017.

[2] J. Webbeh, S. Rahman, and I. Sharf, “Distributed model predictive control for UAVs collaborative payload transport,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 11 666–11 672.

[3] X. Jin and Z. Hu, “Adaptive cooperative load transportation by a team of quadrotors with multiple constraint requirements,” IEEE Transactions on Intelligent Transportation Systems, pp. 1–14, 2022.

[4] N. Rao and S. Sundaram, “An input-output feedback linearization based exponentially stable controller for multi-UAV payload transport,” arXiv preprint arXiv:2207.04375, 2022.

[5] O. Maler and D. Nickovic, “Monitoring temporal properties of continuous signals,” in Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems. Springer, 2004, pp. 152–166.

[6] L. Lindemann and D. V. Dimarogonas, “Control barrier functions for signal temporal logic tasks,” IEEE control systems letters, vol. 3, no. 1, pp. 96–101, 2018.

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

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

[9] Y. V. Pant, H. Yin, M. Arcak, and S. A. Seshia, “Co-design of control and planning for multi-rotor UAVs with signal temporal logic specifications,” in 2021 American Control Conference (ACC). IEEE, 2021, pp. 4209–4216.

[10] Y. V. Pant, H. Abbas, R. A. Quaye, and R. Mangharam, “Fly-by-logic: Control of multi-drone fleets with temporal logic objectives,” in 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICPPS). IEEE, 2018, pp. 186–197.

[11] L. Lindemann and D. V. Dimarogonas, “Barrier function based collaborative control of multiple robots under signal temporal logic tasks,” IEEE Transactions on Control of Network Systems, vol. 7, no. 4, pp. 1916–1928, 2020.

[12] X. Xu, “Constrained control of input-output linearizable systems using control sharing barrier functions,” Automatica, vol. 87, pp. 195–201, 2018.

[13] W. Xiao and C. Belta, “High order control barrier functions,” IEEE Transactions on Automatic Control, 2021.

[14] ———, “Control barrier functions for systems with high relative degree,” in 2019 IEEE 58th conference on decision and control (CDC). IEEE, 2019, pp. 474–479.

[15] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, Robot Operating System (ROS): The Complete Reference (Volume 1). Cham: Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26054-9_23