

STL-Based Motion Planning and Uncertainty-Aware Risk Analysis for Human-Robot Collaboration with a Multi-Rotor Aerial Vehicle

Giuseppe Silano^{1,2*} · Amr Afifi³ · Martin
Saska¹ · Antonio Franchi^{3,4}

Received: date / Accepted: date

Abstract This paper presents a novel approach to motion planning and risk analysis for enhancing human-robot collaboration using a Multi-Rotor Aerial Vehicle (MRV). The proposed method uses Signal Temporal Logic (STL) to encode key mission objectives, such as safety, timing, and human preferences, with a strong focus on ergonomics and comfort. An optimization framework generates dynamically feasible trajectories while considering the MRV's physical constraints. Given the nonlinear and non-convex nature of the problem, smooth approximations and gradient-based techniques assist in handling the problem's computational complexity. Additionally, an uncertainty-aware risk analysis is incorporated to assess potential deviations from the mission specifications, providing insights into the likelihood of mission success under uncertain conditions. Further, an event-triggered replanning strategy is implemented to respond to unforeseen events and external disturbances. The approach is validated through MATLAB and Gazebo simulations, using an object handover task in a mock-up environment inspired by power line maintenance scenarios. The results highlight the method's effectiveness in achieving safe, efficient, and resilient human-robot collaboration.

Keywords Aerial Systems: Applications, Multi-Robot Systems, Human-Aware Motion Planning, Formal Methods in Robotics and Automation, Planning under Uncertainty.

Corresponding author: Giuseppe Silano

¹Giuseppe Silano and Martin Saska are with the Department of Cybernetics, Czech Technical University in Prague, 12135 Prague, Czech Republic (emails: {silangu, martin.saska}@fel.cvut.cz).

²Giuseppe Silano is with the Department of Power Generation Technologies and Materials, Ricerca sul Sistema Energetico (RSE) S.p.A., 20134 Milan, Italy (email: giuseppe.silano@rse-web.it).

³Amr Afifi and Antonio Franchi are with the Robotics and Mechatronics Department, Electrical Engineering, Mathematics, and Computer Science (EEMCS) Faculty, University of Twente, 7500 AE Enschede, The Netherlands (emails: {a.n.m.g.afifi, a.franchi}@utwente.nl).

⁴Antonio Franchi is with Department of Computer, Control and Management Engineering, Sapienza University of Rome, 00185 Rome, Italy (email: antonio.franchi@uniroma1.it).

1 Introduction

Aerial Robots (ARs), particularly Multi-Rotor Aerial Vehicles (MRAVs), have attracted increasing attention in recent years due to their agility, maneuverability, and ability to accommodate a wide range of onboard sensors [46, 54]. Their modular design and versatility make them suitable for a wide range of applications, including contactless interactions [5], physical engagements with the environment [64], wireless communications [36], aerial filming [32], and surveillance and search-and-rescue missions [50]. These capabilities have proven particularly advantageous in demanding environments such as high-altitude workspaces [3], wind turbine maintenance [57], large-scale construction sites [41], and power transmission line inspections [13]. These settings often require specialized personnel, costly equipment, and dedicated vehicles. Furthermore, they involve significant operational hazards, leading to labor-intensive tasks that are prone to human error.

Introducing ARs as *robotic co-workers* [29, 65] in such environments offers numerous benefits. These include the ability to assist with tool handling, monitor worker safety, and alleviate the physical and cognitive burden on human operators [9, 63]. However, the integration of ARs in these scenarios introduces critical safety challenges that must be addressed to enable seamless human-robot collaboration. Moreover, tasks performed at heights can impose ergonomic challenges due to restricted mobility and the need to maintain awkward postures. Consequently, ensuring both safety and human comfort in the design of control solutions for ARs, particularly multi-rotors, becomes a crucial consideration [67].

Despite extensive research on collaborative Human-Robot Interaction (HRI) involving ground-based robots, fewer studies have focused on aerial robots [4]. Previous works have explored manipulators assisting humans in handling heavy or bulky objects and in assembly tasks [27, 62]. Object handover, in particular, has been a key area of interest [31, 47]. However, to facilitate effective human-robot collaboration, especially in high-risk or hazardous environments, advanced motion planning techniques are essential. These techniques must address ergonomic, safety and comfort concerns, minimize the physical and cognitive demands on human operators, and ensure compliance with mission timing and specifications.

Temporal Logic (TL) offers a powerful mathematical framework for defining complex mission specifications that combine temporal and logical constraints [8]. In particular, Signal Temporal Logic (STL) [43] is advantageous because it introduces a metric known as *robustness*. This robustness metric not only determines whether a system satisfies given specifications but also quantifies the degree to which the specifications are fulfilled. This feature allows for the formulation of an optimization problem that maximizes the robustness score, resulting in the generation of optimal, dynamically feasible trajectories that meet mission objectives.

In this paper, we present a motion planning framework that leverages STL specifications to encode collaborative missions involving humans and ARs, with a strong emphasis on improving ergonomic and comfortable collaboration. To illustrate the effectiveness of this approach, we use the task of object handover [22] in a power line maintenance scenario as a motivating example (see Figure 1). The mission requirements are expressed through STL formulae, and the objective is to maximize

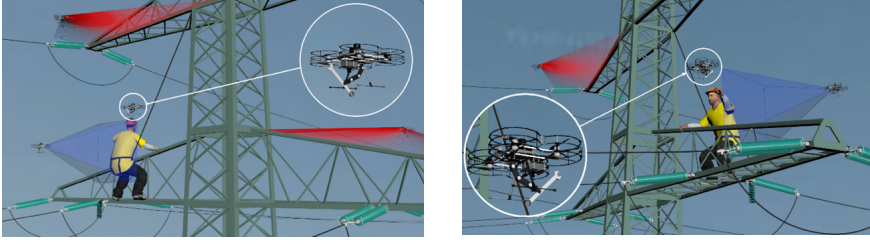


Fig. 1: Illustration of an MRAV facilitating tool delivery to a human worker in a power line scenario.

the robustness of these specifications by solving a nonlinear, non-smooth, and non-convex optimization problem. This approach generates optimal trajectories that account for vehicle dynamics and actuation constraints. Additionally, we introduce a risk-aware analysis that incorporates uncertainties in human pose, providing a systematic framework to assess and quantify the risks associated with deviations from STL specifications. An event-driven replanning strategy is also integrated to handle unforeseen events and external disturbances, ensuring mission continuity.

1.1 Related work

To facilitate collaboration between ARs and humans in shared workspaces, it is essential to integrate HRI principles into the motion planner design. As outlined in [14, 60, 61], an ideal planner should address three critical aspects: (i) *safety*, ensuring that the robot's motion does not endanger humans in the environment; (ii) *reliability and effectiveness*, ensuring the robot's motion considers its dynamics constraints and effectively complete tasks; and (iii) *social acceptability*, integrating behaviors that align with human preferences and social norms to promote smooth, cooperative interactions.

These elements are crucial for developing a successful HRI planner that fosters seamless collaboration and coexistence in shared spaces. For example, even in relatively straightforward tasks like object handovers, the AR must be capable of *reaching* the handover location and *approaching* the human in a controlled, easily understandable, and comfortable manner. This approach enhances the human operator's comfort while promoting efficient and safe collaboration, ensuring that the robot's intentions are easily recognizable and acceptable to the operator. Ergonomics considerations are key to improving the overall interaction experience.

While the field of aerial HRI is growing [31, 67], many human-aware navigation planners are still rooted in proxemics-based criteria [33, 52]. Some alternative methods include the application of Aerial Social Force Model (ASFM), which allow robots to approach humans safely [25, 26]. Additionally, numerous efforts focus on ensuring physical safety of ARs operating in close proximity to humans. Safety mechanisms include imposing constraints on motion planners [45], fine-tuning controller parameters [10], implementing control barrier functions [17], or applying formal verification techniques [7]. However, these solutions generally overlook comfort and ergonomics

Table 1: Comparison of addressed features in related papers and our proposed approach: *included* (✓) and *not included* (✗).

Reference	Features					
	System Dynamics	Ergonomics	Comfort	Time Requirements	Safety	Physical Limits
[7,10,26,33]	✗	✗	✓	✗	✓	✗
[49,59]	✗	✓	✓	✗	✓	✗
[3,16]	✓	✓	✗	✗	✓	✓
[34,66]	✗	✗	✓	✓	✓	✗
Ours	✓	✓	✓	✓	✓	✓

concerns and often simplify robot dynamics, of both which are critical in human-AR collaboration frameworks, especially in high-risk environments.

Human ergonomics and comfort have been explored in several studies, particularly for ground-based robots. For instance, [59] introduced a manipulation planner that considers factors such as ergonomics and the human’s field of view. Similarly, [49] proposed a method for computing human joint torques based on a dynamic whole-body model, allowing ground-based manipulators to minimize human joint overload. However, few of these studies [14,65] specifically address the unique challenges associated with aerial robots interacting with humans.

In addition to addressing strictly planning aspects, research has also focused on enhancing perception systems for ARs to detect and respond to human collaborators. For instance, [16] proposed a Nonlinear Model Predictive Control (NMPC) approach that integrates human ergonomics while also enforcing perception and actuation constraints. This system models the human collaborator within the NMPC to predict future poses and optimize control actions accordingly. Another approach, detailed in [3], formulated a quadratic programming problem to control an AR interacting physically with a human. This framework used admittance control to prioritize ergonomics and safety while an interaction supervisor adjusted compliance based on predefined interaction zones around the human operator. While these studies enable ARs to interact safely and ergonomically with humans, they often fail to address situations requiring the explicit specification of time-based requirements, comfort preferences, and complex mission objectives. For instance, in certain scenarios, human safety may require that the robot approaches slowly from the front rather than from the sides [18,67], or adheres to specific behaviors like maintaining reduced speed [12] or staying within a defined area for a set time before reaching the human [53]. In such cases, expressing mission and comfort requirements through formal specification languages proves beneficial.

In the field of robot controller synthesis based on formal specifications, significant advancements have been made. For instance, [34] presented a controller generated from high-level STL specifications for human-robot handovers, ensuring the precise timing of each handover phase. This approach allows end-users to specify robot behaviors using high-level goals rather than low-level control parameters. However, this method does not address crucial aspects related to ARs, such as the dynamic feasibility of aerial platforms and the need to account for physical actuation constraints. Similarly, [66] employed probabilistic model checking to ensure safety and liveness compliance in human-robot handover tasks [6]. However, this approach lacks the ca-

capacity to manage the continuous, nonlinear dynamics and actuation limits of ARs, which are essential for safe and stable operations in unpredictable environments. In addition, [35] proposed a formalism for human-in-the-loop control synthesis, developing a semi-autonomous controller based on TL specifications. However, this approach does not explicitly address the unique ergonomic and comfort challenges involved in close-proximity HRIs. This limitation is especially critical in high-risk environments like power line maintenance, where aerial robots must work seamlessly with human operators while meeting strict safety and ergonomic standards.

To the best of authors' knowledge, this paper is the first to address the trajectory planning problem with a specific focus on MRAVs for ergonomic and comfortable human-robot collaboration. This approach leverages STL specifications while fully accounting for the nonlinearity of the MRV model. Furthermore, a risk-aware framework is introduced to evaluate the risks associated with meeting complex system specifications and trajectory requirements under human pose uncertainty. An event-driven replanning strategy is also incorporated to manage unforeseen events and external disturbances, ensuring mission continuity. Table 1 provides a comprehensive comparison between related works and our proposed approach, highlighting the inclusion of key features such as system dynamics, ergonomics, comfort, time requirements, safety, and physical actuation limits.

1.2 Contributions

This paper introduces a novel motion planning framework for HRI involving MRAVs, with a primary focus on enhancing ergonomics and operator comfort. The proposed approach leverages STL to formally encode mission objectives, encompassing safety, temporal requirements, and human preferences. By utilizing the expressive capabilities of STL, the proposed method ensures that key aspects of HRI are systematically addressed. An optimization problem is formulated to generate dynamically feasible trajectories that satisfy these specifications while accounting for physical actuation limits and dynamics constraints of the aerial platform. Solving this problem requires addressing a complex nonlinear, non-smooth, and non-convex optimization challenge. To tackle these complexities, we employ smooth approximations, enabling the use of gradient-based optimization techniques. The effectiveness of this approach is demonstrated through an illustrative task: object handovers by an MRV in a power line maintenance scenario.

This framework builds upon previous work [13, 56, 57] on MRV task assignment and trajectory generation. [56] introduced an STL-based planner for fleet-based power line inspection, incorporating obstacle avoidance, inter-robot safety distances, and energy minimization. [13] extended it to bird diverter installation, addressing payload constraints and recharging limitations. [57] further refined the framework with heterogeneous time-bound constraints and a generalized robustness scoring method. Unlike these works, which do not consider human-robot collaboration, this framework integrates ergonomic and comfort constraints while embedding the full nonlinear MRV dynamics. This ensures stable flight and human safety, which is crucial as aerial robots cannot be powered down mid-flight. Additionally, it intro-

duces a systematic risk assessment to quantify deviations due to human pose uncertainty and an event-driven replanner for human-in-the-loop interactions. Designed for both under-actuated and fully actuated MRAs [30], the approach is broadly applicable, with an integrated energy-saving component further optimizing operational efficiency.

The main contributions of this paper can be summarized as follows:

- The problem of object handovers in a power line maintenance scenario, serving as a motivating example for human-robot collaboration using an MRAs, is formulated in Section 2. Mission specifications for this problem are established in Section 4.1, and the corresponding STL formula is derived. An optimization problem is then formulated to determine dynamically feasible trajectories that satisfy safety constraints, temporal requirements, and human preferences while adhering to the STL formula (see Section 4.2).
- The proposed STL optimization problem (Section 4.2) seeks globally optimal solutions but is inherently nonlinear, non-smooth, and non-convex, presenting computational challenges [28,37]. To address these complexities, smooth approximations (Section 3.4) are introduced, enabling the use of gradient-based optimization techniques. While gradient-based methods can be sensitive to local optima [11], various practical strategies exist to mitigate this issue, as explored in previous work [13,57]. Additionally, an energy minimization term is incorporated to implicitly extend the MRAs’s endurance during the mission, contributing to overall efficiency (see Section 5.2).
- The proposed method includes a risk-aware analysis (see Section 4.3) that considers uncertainties in human pose. This analysis provides a systematic approach to assess and quantify the risks associated with potential deviations from STL specifications in the obtained trajectories. It enables the determination of whether a specified success rate for meeting the given STL specifications (e.g., 80%) can be reliably achieved or not under uncertain conditions.
- An event-triggered replanning strategy (Section 4.4) is introduced to handle disturbances and unforeseen events during the mission. This approach reshapes the optimization problem to compute a feasible trajectory that reconnects the drone to the previously computed optimal offline trajectory.
- Numerical simulations in MATLAB (Section 5) evaluate the method’s overall performance in terms of mission specification fulfillment. Additionally, Gazebo simulations (Section 5.4), conducted in a mock-up setting, demonstrate the method’s validity and feasibility under conditions resembling real-world scenarios. Conclusions are presented in Section 6.

2 Motivating Example

The primary aim of this paper is to improve the ergonomic and comfortable collaboration between humans and aerial robots, specifically in high-altitude work environments. This research was conducted as part of the AERIAL-CORE European

project¹, which was initiated in response to findings that safety regulation violations are a leading cause of fatal injuries during maintenance operations on electric power transmission infrastructures. Enhancing human safety and operational efficiency in these high-risk environments is a critical motivation for this work.

The scenario under investigation involves an MRV equipped with a rigidly attached stick carrying a tool, tasked with performing repetitive object handovers to a human operator in a power line maintenance setting (see Figure 1). During maintenance, the MRV frequently approaches the operator to deliver the tool, making the task particularly suited for offline planning. Precomputing trajectories ensures optimization of ergonomic and comfort requirements while accounting for the MRV's physical dynamics and constraints. Given the variability in human pose between successive deliveries, it is crucial to assess the robustness and risks associated with planned trajectories. Evaluating the impact of human pose variations on trajectory feasibility and safety provides insights into the system's reliability and its ability to meet mission objectives. This risk-aware analysis ensures the effectiveness of pre-planned trajectories under pose deviations, emphasizing its importance for repetitive and collaborative tasks. The rigid stick attachment eliminates pendulum effects, ensuring system stability during MRV motion. While power line maintenance serves as a motivating example, the principles of ergonomic and safe collaboration are applicable to a wide range of human-robot interaction scenarios where aerial robots and human workers collaborate efficiently [27].

In such collaborative tasks, the handover configuration must meet several key criteria: *safety*, *visibility*, *ergonomics*, and *comfort*, as outlined in prior studies [60, 61]. To achieve this, the objective is to design a trajectory for the MRV that takes into account human ergonomic needs. The MRV must be able to approach the operator from various directions – front, left, right, above, or below – based on the operator's preferences [67]. Furthermore, to ensure comfort, the speed of approach is carefully regulated to enhance the operator's perception of safety, as described in [12, 53].

For simplicity, the handover operation is modeled within a 3D workspace, where the MRV begins its mission already equipped with the tool. Only one object can be delivered at a time. To meet visibility requirements, the MRV must first reach a designated location directly in front of the operator and remain there for a specified duration before proceeding with the handover. Once the MRV reaches the operator, a low-level onboard controller manages the precise handover process, as established in the authors' previous work [3, 16]. The MRV's operations are subject to physical actuation limits, particularly in terms of maximum propeller velocity. These constraints are critical for ensuring safe interaction with the human operator. The MRV must also account for the forces and torques generated by its rotors, which can impact both the stability of the robot and the comfort of the human operator. Reducing noise and airflow disturbances generated by the propellers is an important consideration for minimizing discomfort during the handover process [15].

The primary goal of this study is to plan a trajectory for the MRV that successfully completes the mission within a predefined time frame while adhering to dynamic constraints and ensuring safety. Safety requirements include remaining within

¹ <https://aerial-core.eu>

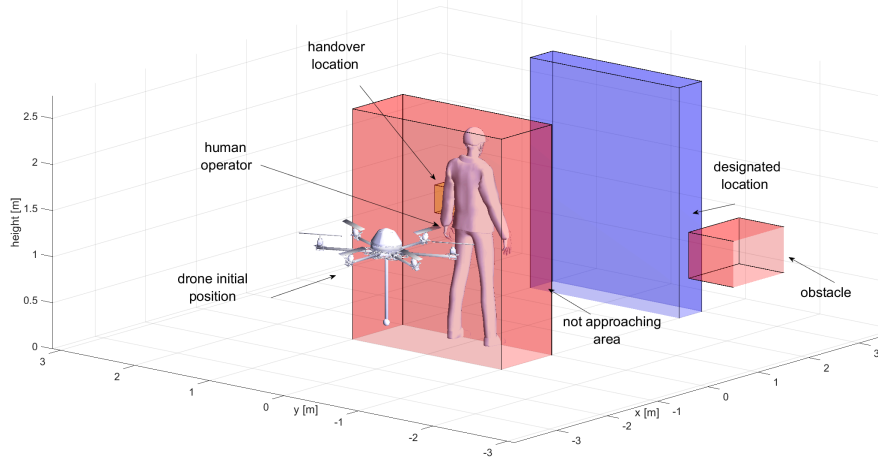


Fig. 2: Schematic depiction of the object handover scenario, highlighting the operator's preferred handover region (yellow), the designated location (blue), and obstacles along with the restricted area behind the operator (red).

the designated workspace, avoiding obstacles, and never approaching the operator from behind. It is assumed that a detailed environment map, including polyhedral representations of obstacles, is known in advance. Figure 2 provides a schematic representation of the scenario.

Although the schematic representation does not explicitly depict a power line environment, it serves as a generalized model to clearly illustrate the core components of the motion planning framework. The principles demonstrated in Figure 2, such as safety, visibility, and ergonomic handover operations, are directly applicable to real-world power line maintenance scenarios. This abstraction allows us to emphasize the versatility and generalizability of the proposed approach, which can be readily adapted to various high-risk environments where similar human-robot collaboration requirements exist.

3 Technical Preliminaries

This section introduces the technical preliminaries necessary to understand the contributions of this paper. It provides an overview of the fundamental concepts related to system dynamics, signal temporal logic, stochastic processes, and risk measures, which form the basis of the proposed approach. Additionally, to facilitate understanding and improve readability, the notation used throughout the paper is summarized in Table 2.

3.1 System modeling

Let us consider the discrete-time dynamical model of the MRAV, described as a Generically-Tilted Multi-Rotor (GTMR) system [3, 16], expressed in the general form

Table 2: Notation - System variables, general symbols and reference frames.

$\mathcal{F}_W, \mathcal{F}_B$	world and body reference frames
\mathbf{p}, \mathbf{v}	MRAV position and velocity in \mathcal{F}_W
$\boldsymbol{\eta}, \boldsymbol{\omega}$	MRAV orientation, and MRAV angular velocity in \mathcal{F}_B
$\mathbf{p}_0, \boldsymbol{\eta}_0$	MRAV initial position and orientation
N_p	number of motor-propeller actuators
$m, \mathbf{J}, \mathbf{g}$	MRAV mass and inertia tensor, and gravity vector
$c\xi_i, c\tau_i, \mathbf{p}m_i$	force and torque constant parameters related to the i -th propeller's design, and i -th motor position expressed in \mathcal{F}_B
$\xi, \tau, \dot{\xi}, \dot{\tau}$	MRAV motor forces and torques, time derivatives of forces, and the squared speed of the i -th motor
\mathbf{z}_{P_i}	unit vector aligned with the i -th motor's axis of rotation
$\mathbf{T}, \mathbf{R}, \mathbf{G}$	Jacobian matrix mapping $\boldsymbol{\omega}$ to $\dot{\boldsymbol{\eta}}$, rotation matrix mapping from \mathcal{F}_B to \mathcal{F}_W , and force/torque allocation matrix
$\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}}$	human operator position and orientation
$\mathbf{t}, N, T_s, \bullet_k$	time vector, number of samples, sampling period, generic k -th element of a sequence
$\pi, I, p_i, \mu_i, \lambda$	STL formula, generic time interval, i -th predicate and its real-valued function, tunable parameter for $\tilde{\rho}_\pi(\mathbf{x})$
$\neg, \wedge, \vee, \implies$	negation, conjunction, disjunction, and implication Boolean operators
$\Diamond, \Box, \mathcal{U}, \bigcirc$	eventually, always, until, and next temporal operators
\mathbf{M}, \mathbf{AP}	set of real-valued functions and the corresponding predicates
$\rho_\pi(\mathbf{x}), \tilde{\rho}_\pi(\mathbf{x})$	robustness and smooth robustness values of the STL formula π
$\rho_\pi(\mathbf{x}, Y(\cdot, \varepsilon)), \tilde{\rho}_\pi(\mathbf{x}, Y(\cdot, \varepsilon))$	robustness and smooth robustness values of the STL formula π considering the realization of the stochastic process Y
Σ, \mathcal{G}, P	sample space, σ -algebra of Σ , and probability measure
Z, ε, F_Z	random variable, element of the sample space Σ , and CDF for Z
$Y, Y(\cdot, \varepsilon)$	stochastic process and realization of the stochastic process Y
$\text{VaR}_\beta(Z), \text{CVaR}_\beta(Z)$	β -Value-at-Risk and Conditional β -Value-at-Risk
$\pi_{\text{ws}}, \pi_{\text{obs}}, \pi_{\text{beh}}$	STL safety requirements
$\pi_{\text{vr}}, \pi_{\text{vis}}, \pi_{\text{pr}}$	STL visibility and ergonomic requirements
$\pi_{\text{vel}}, \pi_{\text{pro}}, \pi_{\text{ho}}$	STL comfort and mission requirements
T_N, T_{vr}	mission duration and visibility time intervals
$N_{\text{obs}}, N_{\text{pr}}$	number of obstacles and number of preference regions
$\underline{\Gamma}_{\text{vel}}, \bar{\Gamma}_{\text{vel}}, \underline{\Gamma}_{\text{pro}}, \bar{\Gamma}_{\text{pro}}, \gamma$	desired minimum and maximum linear velocity, desired minimum and maximum propeller speed, and heading maneuverability margin
$\mathbf{p}_\bullet^{(j)}, \bar{\mathbf{p}}_\bullet^{(j)}$	generic vertices of the rectangular regions defining safety, ergonomic, comfort, and mission requirements
$\mathcal{L}(\mathbf{x}), w, \kappa$	energy term and relative weight, and robustness threshold
$\bar{\xi}, \xi$	maximum and minimum values for the MRAV motor forces
$\mu_z, \sigma_z, \mathbb{I}, K, \delta$	mean and covariance of the normal distribution, indicator function, number of realizations, and level of confidence
$\underline{\text{VaR}}_\beta, \overline{\text{VaR}}_\beta, \underline{\text{CVaR}}_\beta, \overline{\text{CVaR}}_\beta$	lower and upper bounds of VaR_β and CVaR_β
$T_e, T_g, \bar{\mathbf{t}}, \hat{\mathbf{t}}, t_c$	event-driven period, "waypoint" period, corresponding time vectors, and maximum expected computation time for replanning
$\zeta, \tilde{\mathbf{p}}, \mathbf{p}^*$	replanning threshold, actual and optimal MRAV positions

$x_{k+1} = f(x_k, u_k)$. Here, $x_{k+1}, x_k \in \mathcal{X} \subset \mathbb{R}^n$ represent the next and current states of the system at time step k , respectively, and $u_k \in \mathcal{U} \subset \mathbb{R}^m$ is the control input. The system is actuated by N_p motor-propeller units, and its dynamics are derived using the Newton-Euler formalism. These actuators are placed arbitrarily and oriented with respect to (w.r.t.) the vehicle's main body. The number of actuators and their orienta-

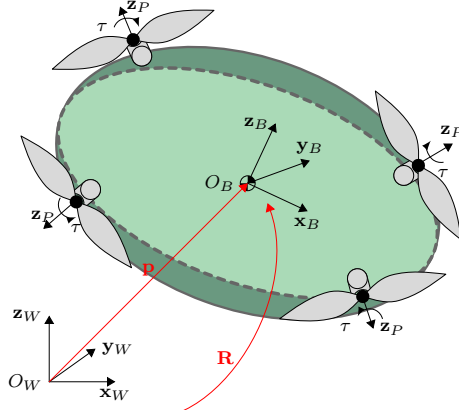


Fig. 3: Schematic representation of an GTMR system with its world $\mathcal{F}_W = \{O_W, \mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$ and body $\mathcal{F}_B = \{O_B, \mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$ reference frames.

tion w.r.t. the vehicle body determines whether the system is an under-actuated or a fully-actuated platform [30].

The MRAV is described in two reference frames: the world frame \mathcal{F}_W and the body frame \mathcal{F}_B . The body frame is attached to the vehicle's Center of Mass (CoM), denoted by O_B , as shown in Figure 3. The system's key parameters include mass $m \in \mathbb{R}_{>0}$, inertia tensor $\mathbf{J} \in \mathbb{R}^{3 \times 3}$, position $\mathbf{p} \in \mathbb{R}^3$, and linear velocity $\mathbf{v} \in \mathbb{R}^3$ expressed in \mathcal{F}_W . The orientation of the vehicle is described using Euler angles $\boldsymbol{\eta} = (\varphi, \vartheta, \psi)^\top$ – roll, pitch, and yaw, while its angular velocity is given by $\boldsymbol{\omega} \in \mathbb{R}^3$ in \mathcal{F}_B . The forces generated by the motors are represented as $\boldsymbol{\xi} \in \mathbb{R}^{N_p}$, acting on the vehicle's CoM.

The force ξ_i and torque τ_i exerted by the i -th motor, with $i = \{1, \dots, N_p\}$, on the vehicle are given by the relations $\xi_i = c_{\xi_i} \Omega_i \mathbf{z}_{P_i}$ and $\tau_i = (c_{f_i} \mathbf{p}_{m_i} \times \mathbf{z}_{P_i} + c_{\tau_i} \mathbf{z}_{P_i}) \Omega_i$, where c_{ξ_i} and c_{τ_i} are constants related to the propeller design, $\mathbf{p}_{m_i} \in \mathbb{R}^3$ is the motor position in \mathcal{F}_B , $\mathbf{z}_{P_i} \in \mathbb{S}^2$ is the motor's axis of rotation, and $\Omega_i \in \mathbb{R}_{\geq 0}$ is the squared motor speed [3, 16].

To describe the time evolution of the system, a time vector $\mathbf{t} = (t_0, \dots, t_N)^\top \in \mathbb{R}^{N+1}$ is defined, where $N \in \mathbb{N}_{>0}$ represents the number of time steps, and the system evolves with a sampling period $T_s \in \mathbb{R}_{>0}$. The state \mathbf{x} and control input \mathbf{u} sequences are denoted as $\mathbf{x} = (\mathbf{p}, \boldsymbol{\eta}, \mathbf{v}, \boldsymbol{\omega}, \boldsymbol{\xi})^\top \in \mathbb{R}^{(12+N_p) \times N}$ and $\mathbf{u} = \dot{\boldsymbol{\xi}} \in \mathbb{R}^{N_p \times N}$, with each k -th element of the sequences (i.e., \mathbf{p} , \mathbf{v} , etc.) denoted as \bullet_k (i.e., p_k , v_k , etc.). Using the Newton-Euler approach, the MRAV dynamics can be described by the set of equations:

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ \dot{\boldsymbol{\eta}} = \mathbf{T}(\boldsymbol{\eta})\boldsymbol{\omega} \\ m\dot{\mathbf{v}} = m\mathbf{g} + \mathbf{R}(\boldsymbol{\eta})\mathbf{G}\boldsymbol{\xi} \\ \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{G}\boldsymbol{\xi} \end{cases}, \quad (1)$$

where $\mathbf{T}(\boldsymbol{\eta}) \in \mathbb{R}^{3 \times 3}$ is the Jacobian matrix mapping $\boldsymbol{\omega}$ to $\dot{\boldsymbol{\eta}}$, $\mathbf{R}(\boldsymbol{\eta}) \in \mathbb{R}^{3 \times 3}$ is the rotation matrix from \mathcal{F}_B to \mathcal{F}_W , $\mathbf{g} = (0, 0, -g)^\top$ is the gravitational acceleration vector, and $\mathbf{G} \in \mathbb{R}^{6 \times N_p}$ is the force/torque allocation matrix [3, 16] mapping the

forces produced by each actuator to the total force and torque acting on the vehicle's CoM.

3.2 Signal temporal logic

STL was first introduced in [43] to monitor and specify the behavior of real-valued signals over time. It provides a powerful, compact, and unambiguous way to represent complex system behaviors by encoding mission specifications into a single formula π [43]. For example, it can capture requirements such as: “At least two vehicles must survey regions A and B, one must visit region C within the time interval $[t_1, t_2]$, and all vehicles must adhere to safety constraints”. The formal syntax and semantics of STL are described in detail in [20, 43], but they are not included here for brevity.

In essence, an STL formula π is built from a set of predicates p_i , where $i \in \mathbb{N}_0$. These predicates serve as atomic propositions that can represent simple system conditions, such as whether a state variable belongs to a certain region or satisfies a threshold condition. Formally, let $M = \{\mu_1, \dots, \mu_L\}$ be a set of real-valued functions of the system state, where $\mu_i: \mathcal{X} \rightarrow \mathbb{R}$. Each predicate p_i corresponds to a subset of the state space \mathcal{X} , specifically defined as $p_i = \{x \in \mathcal{X} | \mu_i(x) \geq 0\}$. Together, these predicates form the atomic propositions $AP := \{p_1, \dots, p_L\}$ used to define more complex STL specifications.

These predicates can be combined using standard Boolean operators such as *negation* (\neg), *conjunction* (\wedge), *disjunction* (\vee), and *implication* (\implies), as well as temporal operators. Temporal operators like *eventually* (\Diamond), *always* (\Box), *until* (\mathcal{U}), and *next* (\bigcirc), enable STL to specify constraints over non-singleton intervals $I \subset \mathbb{R}$. Thus, an STL formula π is recursively constructed from predicates p_i using the following grammar:

$$\pi := \top | \neg \pi | \pi_1 \wedge \pi_2 | \pi_1 \vee \pi_2 | \Box_I \pi | \Diamond_I \pi | \bigcirc_I \pi | \pi_1 \mathcal{U}_I \pi_2,$$

where π_1 and π_2 are STL formulae. The resulting STL formula π is considered valid if it evaluates to true (\top), and invalid if false (\perp). For example, informally, the expression $\pi_1 \mathcal{U}_I \pi_2$ implies that formula π_2 must hold at some point within the interval I , and until then, formula π_1 must remain continuously satisfied.

3.3 Robust signal temporal logic

System uncertainties, dynamic environmental changes, and unforeseen events can all impact the satisfaction of an STL formula π . To introduce flexibility in how well π is satisfied and to quantify how effectively a given specification is met, the concept of *robust semantics* for STL formulae was developed [20, 24, 43]. This *robustness*, denoted as ρ , provides a quantitative measure to guide optimization towards the most feasible solution for satisfying the mission specifications. Robustness is defined re-

cursively using the following formulae:

$$\begin{aligned}
\rho_{p_i}(\mathbf{x}, t_k) &= \mu_i(\mathbf{x}, t_k), \\
\rho_{\neg\pi}(\mathbf{x}, t_k) &= -\rho_{\pi}(\mathbf{x}, t_k), \\
\rho_{\pi_1 \wedge \pi_2}(\mathbf{x}, t_k) &= \min(\rho_{\pi_1}(\mathbf{x}, t_k), \rho_{\pi_2}(\mathbf{x}, t_k)), \\
\rho_{\pi_1 \vee \pi_2}(\mathbf{x}, t_k) &= \max(\rho_{\pi_1}(\mathbf{x}, t_k), \rho_{\pi_2}(\mathbf{x}, t_k)), \\
\rho_{\Box_I \pi}(\mathbf{x}, t_k) &= \min_{t'_k \in [t_k, t_k+I]} \rho_{\pi}(\mathbf{x}, t'_k), \\
\rho_{\Diamond_I \pi}(\mathbf{x}, t_k) &= \max_{t'_k \in [t_k, t_k+I]} \rho_{\pi}(\mathbf{x}, t'_k), \\
\rho_{\bigcirc_I \pi}(\mathbf{x}, t_k) &= \rho_{\pi}(\mathbf{x}, t'_k), \text{ with } t'_k \in [t_k, t_k+I], \\
\rho_{\pi_1 \mathcal{U}_I \pi_2}(\mathbf{x}, t_k) &= \max_{t'_k \in [t_k, t_k+I]} \left(\min(\rho_{\pi_2}(\mathbf{x}, t'_k)), \min_{t''_k \in [t_k, t'_k]} (\rho_{\pi_1}(\mathbf{x}, t''_k)) \right).
\end{aligned}$$

In this context, $t_k + I$ denotes the Minkowski sum of the scalar t_k and the interval I . These formulae, as said, are recursively defined from *predicates* p_i and their corresponding real-valued functions $\mu_i(\mathbf{x}, t_k)$. Predicates are considered true if their robustness value is greater than zero ($\mu_i(\mathbf{x}, t_k) > 0$) and false otherwise ($\mu_i(\mathbf{x}, t_k) \leq 0$).

The entire formula behaves as a logical expression, evaluating to false if at least one predicate is false. In simple terms, the STL formula $\pi_1 \wedge \pi_2$ is satisfied if either π_1 or π_2 is true. Evaluation follows the application of logical and temporal operators (e.g., *always*, *eventually*, *conjunction*) from the innermost part to the outermost part of the formula. For instance, the robustness of the formula could determine whether and “how well” a system remains inside a target region or avoids an obstacle at a particular interval I . Further details can be found in [20,21,43]. In this context, we say that \mathbf{x} satisfies the STL formula π at time t_k if $\rho_{\pi}(\mathbf{x}, t_k) > 0$ (denoted as $\mathbf{x}(t_k) \models \pi$) and \mathbf{x} violates π if $\rho_{\pi}(\mathbf{x}, t_k) \leq 0$. To simplify notation, we use $\rho_{\pi}(\mathbf{x})$ instead of $\rho_{\pi}(\mathbf{x}, 0)$ when $t_k = 0$. Moreover, the value of $\rho_{\pi}(\mathbf{x}, t_k)$ represents “how well” the formula π is satisfied (if $\rho_{\pi}(\mathbf{x}, t_k) > 0$) or “how much” is violated (if $\rho_{\pi}(\mathbf{x}, t_k) \leq 0$), implicitly introducing a robustness criterion.

With this understanding, control inputs \mathbf{u} can be optimized to maximize the robustness $\rho_{\pi}(\mathbf{x})$ over a set of finite state and input sequences \mathbf{x} and \mathbf{u} . An optimal control sequence \mathbf{u}^* is considered valid if the resulting robustness $\rho_{\pi}(\mathbf{x}^*)$ is positive, where \mathbf{x}^* and \mathbf{u}^* adhere to the dynamical system. A higher robustness value $\rho_{\pi}(\mathbf{x}^*)$ indicates that the system can tolerate greater disturbances without violating the STL specification.

3.4 Smooth approximation

The computation of $\rho_{\pi}(\mathbf{x})$ involves non-differentiable functions like \min and \max (see Section 3.3), which can significantly increase the computational complexity, especially when used in optimization routines. To address this challenge, it is advantageous to utilize a *smooth approximation*, denoted as $\tilde{\rho}_{\pi}(\mathbf{x})$, of the robustness function $\rho_{\pi}(\mathbf{x})$. This approach facilitates more efficient computations by replacing the non-differentiable \min and \max operations with smooth, differentiable alternatives. Hence, considering $\lambda \in \mathbb{R}_{>0}$ as a tunable parameter, we can express the smooth

approximation of the min and max operators with β -arguments as follows:

$$\begin{aligned}\max(\rho_{\pi_1}, \dots, \rho_{\pi_\beta}) &\approx \frac{\sum_{i=1}^{\beta} \rho_{\pi_i} e^{\lambda \rho_{\pi_i}}}{\sum_{i=1}^{\beta} e^{\lambda \rho_{\pi_i}}}, \\ \min(\rho_{\pi_1}, \dots, \rho_{\pi_\beta}) &\approx -\frac{1}{\lambda} \log \left(\sum_{i=1}^{\beta} e^{-\lambda \rho_{\pi_i}} \right).\end{aligned}$$

In contrast to our earlier work [55, 56], this paper employs an improved smooth robustness measure [28], which offers significant advantages over the widely-used Log-Sum-Exponential (LSE) method [20]. This enhanced approach maintains several key properties, such as *asymptotic completeness* and *smoothness everywhere*, similar to LSE. Additionally, it guarantees *soundness*, meaning that an optimal control sequence \mathbf{u}^* with strictly positive smooth robustness ($\tilde{\rho}_\pi(\mathbf{x}) > 0$) satisfies the specification π , while a sequence \mathbf{u}^* with strictly negative smooth robustness ($\tilde{\rho}_\pi(\mathbf{x}) < 0$) violates it. The property of *asymptotic completeness* ensures that as the parameter λ increases ($\lambda \rightarrow \infty$), the smooth approximation $\tilde{\rho}_\pi(\mathbf{x})$ converges to the true robustness $\rho_\pi(\mathbf{x})$. Moreover, *smoothness everywhere* guarantees that the approximation is infinitely differentiable, making it compatible with gradient-based optimization methods, which are more computationally efficient for solving complex problems [28]. Increasing λ improves the accuracy of the approximation, allowing it to better reflect the true robustness of the system (see Section 3.3).

3.5 Stochastic processes and risk measure

In addition to interpreting STL formulae over deterministic signals, it is equally important to extend their interpretation to stochastic processes. To do so, we consider a probability space (Σ, \mathcal{G}, P) , where Σ represents the sample space, \mathcal{G} is a σ -algebra over Σ , and P is a probability measure mapping from \mathcal{G} to the interval $[0, 1]$. In this framework, we define a real-valued *random vector* as Z , which is a measurable function $Z: \Sigma \rightarrow \mathbb{R}^n$. For $n = 1$, Z is referred to as a random variable. Each realization $Z(\varepsilon)$ corresponds to a specific outcome Z , where $\varepsilon \in \Sigma$ [23].

Since Z is measurable, we can construct a probability space specifically for it, which allows us to define a Cumulative Distribution Function (CDF), denoted as $F_Z(z)$, for the random vector Z . If a measurable function $g: \mathbb{R}^n \rightarrow \mathbb{R}$ is applied to Z , the result $g(Z(\varepsilon))$ becomes a *derived random variable*, because function composition preserves measurability, as detailed in [23].

Hence, a *stochastic process* is defined as a function $Y: T \times \Sigma \rightarrow \mathbb{R}^n$, where T represents the time domain, and for each fixed time $t_k \in T$, $Y(t_k, \cdot)$ is a random vector. Essentially, a stochastic process is a collection of random vectors $\{Y(t_k, \cdot) | t_k \in T\}$, with each vector defined on the probability space (Σ, \mathcal{G}, P) and indexed by time. When $\varepsilon \in \Sigma$ is fixed, the function $Y(\cdot, \varepsilon)$ becomes a *realization* of the process. Alternatively, a stochastic process can be viewed as a collection of deterministic time functions $\{Y(\cdot, \varepsilon) | \varepsilon \in \Sigma\}$, each indexed by elements from Σ [23]. This duality is particularly useful for applying STL formulae to systems governed by stochastic processes [38].

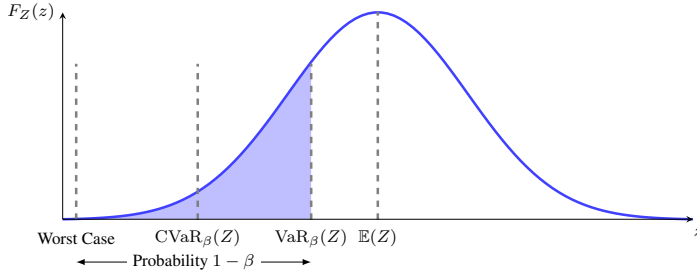


Fig. 4: Illustration of the expected value $\mathbb{E}(Z)$, β -Value-at-Risk $\text{VaR}_\beta(Z)$, and Conditional β -Value-at-Risk $\text{CVaR}_\beta(Z)$ for a specified risk level $\beta \in (0, 1)$. The axes represent the stochastic variable z and its CDF $F_Z(z)$. The shaded area corresponds to $\% \beta$ of the total area under $F_Z(z)$. $\text{VaR}_\beta(Z)$ represents the value of z at the β -tail of the distribution, while $\text{CVaR}_\beta(Z)$ averages the worst-case values of z in the β -tail. A negative $\text{CVaR}_\beta(Z)$ indicates unsafe behavior.

A *risk measure* is a function $R : \mathcal{H}(\Sigma, \mathbb{R}) \rightarrow \mathbb{R}$, which maps real-valued random variables (often referred to as *cost random variables*) to real numbers. These measures are critical for assessing risks associated with uncertain outcomes, particularly in safety-critical applications.

In this paper, we focus on common risk measures, such as the expected value, β -Value-at-Risk (VaR_β), and Conditional β -Value-at-Risk (CVaR_β) at a specified risk level $\beta \in (0, 1)$, as shown in Figure 4. Hence, the VaR_β of a random variable $Z : \Sigma \rightarrow \mathbb{R}$ is defined as:

$$\text{VaR}_\beta(Z) := \inf\{\alpha \in \mathbb{R} \mid F_Z(\alpha) \geq \beta\},$$

in other words, $\text{VaR}_\beta(Z)$ corresponds to the $(1 - \beta)$ quantile of the random variable Z . The CVaR_β of the random variable Z is defined as:

$$\text{CVaR}_\beta(Z) := \inf_{\alpha \in \mathbb{R}} \left(\alpha + (1 - \beta)^{-1} \mathbb{E}([Z - \alpha]^+) \right),$$

where $[Z - \alpha]^+ := \max(Z - \alpha, 0)$ and $\mathbb{E}(\cdot)$ represents the expected value. In cases where the CDF F_Z of Z is continuous, $\text{CVaR}_\beta(Z)$ can be expressed as $\mathbb{E}(Z \mid Z \geq \text{VaR}_\beta(Z))$. In simpler terms, $\text{CVaR}_\beta(Z)$ represents the expected value of Z under the condition that Z is greater than or equal to $\text{VaR}_\beta(Z)$ [42].

4 Methodology

In this section, we present the motion planning framework designed to address the mission specifications outlined in Section 2 and expressed as an STL formula π (Section 4.1). The motivating example focuses on the challenge of object handovers in a power line maintenance scenario, which highlights the application of human-robot collaboration with MRAsVs. This task requires solving a complex optimization problem that is nonlinear, non-smooth, and non-convex, in order to generate dynamically feasible trajectories that meet safety constraints, temporal requirements, and human ergonomic and comfort preferences, while ensuring adherence to the STL formula π .

To address the computational challenges posed by the robustness function $\rho_\pi(\mathbf{x})$, we employ smooth approximations (Section 3.4), which enable the use of gradient-based optimization techniques. These approximations make the problem more tractable by facilitating the optimization of the mission’s robustness w.r.t. the desired system behaviors. Additionally, we enhance the planner by incorporating an energy minimization term, which implicitly extends the endurance of the MRV during the mission (Section 4.2). This addition not only improves the vehicle’s energy efficiency but also ensures that the planned trajectory remains feasible under the vehicle’s physical and actuation constraints.

Additionally, we augment the framework with a risk-aware analysis (Section 4.3) that accounts for uncertainties in human pose. This systematic approach evaluates and quantifies the risks associated with potential deviations from STL specifications, ensuring that the planned trajectories maintain a high probability of mission success even in uncertain environments. Lastly, an event-triggered replanner is embedded to cope with disturbances or unforeseen events (Section 4.4).

4.1 Mission specifications encoding

This section outlines the mission specifications for the problem introduced in Section 2 and defines the corresponding STL formula π . The scenario involves an MRV equipped with a rigidly attached stick carrying a small object, tasked with performing object handovers in a power line maintenance setting, as depicted in Figure 2. The mission specifications are categorized into four key areas: safety, visibility, ergonomic, and comfort requirements.

The *safety requirements* are crucial for ensuring safe operation throughout the mission duration, T_N . These include keeping the MRV within the designated workspace (π_{ws}), avoiding collisions with surrounding objects (π_{obs}), and preventing the drone from approaching the operator from behind (π_{beh}). The *visibility requirements* ensure that during the mission, the MRV reaches a designated location in front of the human and remains there for a specific duration T_{vr} (π_{vr}) before proceeding further. While approaching the operator, the drone must align its heading with the direction of movement (π_{vis}), allowing the operator to maintain continuous visual contact with the small object on the stick. Once the MRV reaches the operator, a low-level onboard controller manages the handover process (π_{ho}). The *ergonomic requirements* ensure that the MRV approaches the operator from their preferred direction – whether from the front, left, right, above, or below – based on the operator’s communicated preferences before the mission (π_{pr}), reducing strain on the operator [67]. The *comfort requirements* aim to enhance operator comfort during the mission by limiting the drone’s approach speed (π_{vel}), as a slower speed increases perceived safety, as discussed in [12, 53]. Additionally, the maximum propeller velocity is restricted (π_{pro}) to reduce noise and wind from the propellers, mitigating any discomfort for the operator [14, 15]. This is especially important in environments where the operator may be working in close proximity to the drone for extended periods, as in power line maintenance scenarios.

Importantly, the role of the proposed planner is to compute the optimal trajectory for the MRV from the start to the handover location, leaving the execution of the handover itself to the onboard system. Example such as those from previous work by the authors [3, 16] offer viable options, but any appropriate handover algorithm can be applied based on the specific mission requirements.

Hence, all above mission specifications are combined into the following STL formula:

$$\begin{aligned} \pi = & \Box_{[0, T_N]} (\pi_{ws} \wedge \pi_{obs} \wedge \pi_{beh}) \wedge \\ & (\Diamond_{[0, T_N - T_{vr}]} \Box_{[0, T_{vr}]} \pi_{vr}) \mathcal{U}_{[0, T_N - T_{vr}]} \left((\pi_{pr} \wedge \pi_{vel} \wedge \pi_{pro} \wedge \pi_{vis}) \wedge \right. \\ & \left. \Box_{[1, T_N - T_{vr} - 1]} (\pi_{ho} \implies \bigcirc_{[0, t_{k+1}]} \pi_{ho}) \right) \end{aligned} \quad (2)$$

The STL formula π consists of nine specifications (π_{ws} , π_{obs} , π_{beh} , π_{vr} , π_{pr} , π_{vel} , π_{pro} , π_{vis} , and π_{ho}) and two time intervals (T_N and T_{vr}). The following equations describe each of these specifications:

$$\pi_{ws} = \bigwedge_{j=1}^3 \mathbf{p}^{(j)} \in (\underline{p}_{ws}^{(j)}, \bar{p}_{ws}^{(j)}), \quad (3a)$$

$$\pi_{obs} = \bigwedge_{j=1}^3 \bigwedge_{q=1}^{N_{obs}} \mathbf{p}^{(j)} \notin (\underline{p}_{obs,q}^{(j)}, \bar{p}_{obs,q}^{(j)}), \quad (3b)$$

$$\pi_{beh} = \bigwedge_{j=1}^3 \mathbf{p}^{(j)} \notin (\underline{p}_{beh}^{(j)}, \bar{p}_{beh}^{(j)}), \quad (3c)$$

$$\pi_{vr} = \bigwedge_{j=1}^3 \mathbf{p}^{(j)} \in (\underline{p}_{vr}^{(j)}, \bar{p}_{vr}^{(j)}), \quad (3d)$$

$$\pi_{pr} = \bigwedge_{j=1}^3 \bigwedge_{q=1}^{N_{pr}} \left(\mathbf{p}^{(j)} \in (\underline{p}_{pr,q}^{(j)}, \bar{p}_{pr,q}^{(j)}) \right), \quad (3e)$$

$$\pi_{vel} = \|\mathbf{v}(t_k)\| \in (\underline{\Gamma}_{vel}, \bar{\Gamma}_{vel}), \quad (3f)$$

$$\pi_{pro} = \bigwedge_{q=1}^{N_p} \Omega_q(t_k) \in (\underline{\Gamma}_{pro}, \bar{\Gamma}_{pro}), \quad (3g)$$

$$\begin{aligned} \pi_{vis} = & \psi(t_k) \in (\psi_{vis}(t_k) - \gamma, \psi_{vis}(t_k) + \gamma), \text{ with} \\ & \psi_{vis} = \text{atan2}(p_k^{(2)} - p_{k-1}^{(2)}, p_k^{(1)} - p_{k-1}^{(1)}), \end{aligned} \quad (3h)$$

$$\pi_{ho} = \bigwedge_{j=1}^3 \mathbf{p}^{(j)} \in (\underline{p}_{ho}^{(j)}, \bar{p}_{ho}^{(j)}). \quad (3i)$$

In (3a), the MRV's position $\mathbf{p}^{(j)}$, where $j = \{1, 2, 3\}$, along the j -axis of the world frame \mathcal{F}_W , is constrained to remain within the workspace boundaries, defined by $\underline{p}_{ws}^{(j)}$ and $\bar{p}_{ws}^{(j)}$. Obstacle avoidance and the restriction preventing the MRV from approaching the operator from behind are captured by (3b) and (3c), respectively. Here, N_{obs} represents the number of obstacles in the environment, with rectangular regions having vertices $(\underline{p}_{obs,q}^{(j)}, \bar{p}_{obs,q}^{(j)})$ and $(\underline{p}_{beh}^{(j)}, \bar{p}_{beh}^{(j)})$ defining obstacle areas and the region behind the operator, respectively.

The visibility requirements, described in (3d) and (3h), ensure that the MRV reaches a designated location in front of the operator and aligns its heading with the direction of movement. The margin for maneuverability, denoted by $\gamma \in \mathbb{R}_{>0}$, aids the optimization process by allowing the MRV to adjust its heading within a defined range, while still meeting visibility and approach requirements. The operator's preferences for the drone's approach direction (front, left, right, above, or below), as

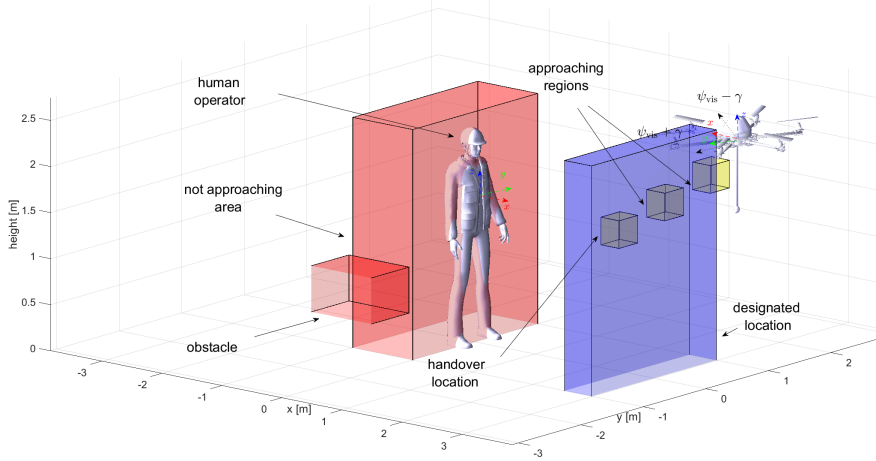


Fig. 5: Object handover scenario with highlighted approaching regions (yellow) representing the ergonomic preferences from top-to-bottom and left-to-right. Reference axes aid in visualizing the drone’s maneuverability margin ($\psi_{\text{vis}} \pm \gamma$) in the displacement direction.

detailed in [67], are encoded in (3e). Specifically, the vertices $(p_{\text{pr},q}^{(j)}, \bar{p}_{\text{pr},q}^{(j)})$ define in order the regions within \mathcal{F}_W where the drone is allowed to approach, based on both the operator’s pose and preferences. These regions are geometrically derived by extending from the handover location to the stopping area in front of the operator (π_{vr}), based on the operator’s approach preferences (left, right, above, or below). This creates well-defined approach corridors for the drone to follow. A schematic representation of this setup is shown in Figure 5.

Comfort requirements, which include limiting the drone’s approach speed and setting a maximum propeller velocity, are described in (3f) and (3g), respectively. The parameters $(\underline{\Gamma}_{\text{vel}}, \bar{\Gamma}_{\text{vel}}) \in \mathbb{R}_{>0}$ and $(\underline{\Gamma}_{\text{pro}}, \bar{\Gamma}_{\text{pro}}) \in \mathbb{R}_{>0}$ set thresholds for the linear velocity of the drone and the rotational speed of the actuators. The relationship between the propeller speed Ω_q , with $q = \{1, \dots, N_p\}$, and the system state \mathbf{x} is detailed in Section 3.1, where the dynamics governing this interaction are thoroughly explained.

Lastly, (3i) provides guidelines for completing the mission, defining the object handover position through the boundaries $p_{\text{ho}}^{(j)}$ and $\bar{p}_{\text{ho}}^{(j)}$. These mission specifications (3) ensure that the MRV performs the task safely while considering for operator comfort and preferences.

The *always* operators (\square) in (2) guarantees compliance with time requirements T_N and T_{vr} , corresponding to the mission duration and visibility requirement, respectively. The *eventually* operator (\diamond) ensures that the π_{vr} specification is satisfied within the time frame $T_N - T_{\text{vr}}$. Additionally, the *until* operator (\mathcal{U}) ensures that the MRV does not approach the operator for handover before reaching the designated location (π_{vr}).

To solve the optimization problem for satisfying these mission specifications, it is necessary to compute the robustness score $\rho_\pi(\mathbf{x})$ associated with the STL formula π . The robustness values, calculated as Euclidean distances in \mathbb{R}^n , indicate how well

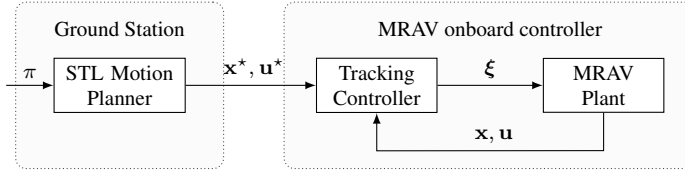


Fig. 6: System Architecture: The *STL Motion Planner* at the ground station generates optimal trajectories $(\mathbf{x}^*, \mathbf{u}^*)$ for the MRV. These trajectories are then fed into the *Tracking Controller*, which operates in closed loop to compute the motor forces ξ , ensuring precise flight maneuvers.

the MRV adheres to the mission requirements. For instance, a positive robustness score indicates that the MRV is within the specified region, while a negative score indicates that it has violated constraints, such as entering obstacle areas or approaching the operator from behind. Further details are provided in the authors' previous work [13, 57], here omitted for brevity.

4.2 Motion planner

In this section, we explain how to generate the trajectory for the MRV based on the mission specifications π . The motion planner is designed as an optimization problem that produces a feasible trajectory while accounting for the vehicle's physical constraints. These trajectories are then used by the tracking controller to execute the handover task. Figure 6 illustrates the overall system architecture.

Leveraging the robust semantics of STL (see Section 3.3), the motion planner is synthesized by framing the task of determining a dynamically feasible control sequence for the MRV as an optimization problem. This problem seeks to satisfy a given STL formula π (Section 4.1) by optimizing over the control sequence $\mathbf{u} = (u_0, \dots, u_{N-1})^\top \in \mathbb{R}^N$ and the corresponding state sequence $\mathbf{x} = (x_0, \dots, x_{N-1})^\top \in \mathbb{R}^{N+1}$ (see Section 3.1), as follows:

$$\underset{\mathbf{x}, \mathbf{u}}{\text{minimize}} \quad w(\mathbf{x}) - \rho_\pi(\mathbf{x}) \quad (4a)$$

$$\text{s.t.} \quad \mathbf{x}_0 = \mathbf{x}(t_0), \quad (4b)$$

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad (4c)$$

$$\rho_\pi(\mathbf{x}) \geq \kappa, \quad (4d)$$

$$\underline{\xi} \leq \xi_k \leq \bar{\xi}, \forall k = \{0, 1, \dots, N-1\}. \quad (4e)$$

In the above equations, (4a) defines the objective function, which consists of two terms: the energy minimization term $\mathcal{L}(\mathbf{x})$ and the robustness degree $\rho_\pi(\mathbf{x})$. (4b) specifies the system's initial state $\mathbf{x}(t_0)$, while (4c) models the system dynamics as $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$, following the GTMR system framework (see Section 3.1). This ensures that the generated trajectories account for the nonlinear dynamics of the MRV, whether it operates in an under-actuated configuration or a fully-actuated one [30]. (4d) sets a minimum robustness threshold, acting as a safety margin to ensure satisfaction of the STL formula π even if the energy prioritization leads to a

trade-off. Lastly, (4e) imposes the motor force constraints, with $\underline{\xi}$ and $\bar{\xi}$, representing the vehicle's minimum and maximum physical actuation limits.

The energy minimization term $\mathcal{L}(\mathbf{x})$, part of the objective function (4a), is designed to extend the MRV's endurance by reducing energy consumption. This is achieved through a weight coefficient $w \in \mathbb{R}_{>0}$, which allows for tuning the balance between energy efficiency and the satisfaction of mission objectives, represented by the robustness degrees $\rho_\pi(\mathbf{x})$ of the STL formula π . A higher w increases the emphasis on energy minimization, while a lower w focuses more on meeting mission objectives. The energy minimization term $\mathcal{L}(\mathbf{x})$ includes the power consumption and kinetic energy, as expressed by the following equation:

$$w(\mathbf{x}) = \sum_{i=1}^{N_p} c_{\xi_i} \Omega_i^3 + \frac{1}{2} [\mathbf{v}^\top \ \boldsymbol{\omega}^\top] \begin{bmatrix} m\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}, \quad (5)$$

where $\mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$ and $\mathbf{0}_3 \in \mathbb{R}^{3 \times 3}$ are the identity and zero matrices, respectively.

It is crucial to distinguish that, unlike the specification π_{pro} in (3g), which restricts propeller velocity (and thus motor forces, see Section 3.1) only during the approach phase to ensure comfort, the motor force constraints in this optimization guarantee apply throughout the entire mission. This ensures that the vehicle operates within its physical limits at all times. Typically, the allowable propeller velocity range (Γ_{pro} and $\bar{\Gamma}_{\text{pro}}$) is significantly narrower than the range of physically actuatable propeller speeds ($\underline{\xi}$ and $\bar{\xi}$). This limitation minimizes noise and wind, enhancing operator comfort during close-range interactions.

As described in Section 3.4, the robustness function $\rho_\pi(\mathbf{x})$ involves non-differentiable functions like max and min, making it non-differentiable itself w.r.t. the state \mathbf{x} and control inputs \mathbf{u} . While various approaches like mixed-integer programming solvers [51], non-smooth optimizers [1], or stochastic heuristics [2] can be employed to find a solution for this problem, it is crucial to acknowledge that the problem is inherently NP-hard, and these methods may encounter difficulties, particularly as the number of variables increases [11]. However, as demonstrated in [48], one effective strategy for managing the computational complexity is to adopt a smooth approximation $\tilde{\rho}_\pi(\mathbf{x})$ of the robust function $\rho_\pi(\mathbf{x})$. In this scenario, the resulting optimization problem remains nonlinear and non-convex, but it becomes amenable to smooth optimization techniques such as sequential quadratic programming, which can help identify a local minimum [11].

Thus, the problem (4) can be reformulated by replacing $\rho_\pi(\mathbf{x})$ with its smooth counterpart $\tilde{\rho}_\pi(\mathbf{x})$ as follows:

$$\begin{aligned} & \underset{\mathbf{x}, \mathbf{u}}{\text{minimize}} && w(\mathbf{x}) - \tilde{\rho}_\pi(\mathbf{x}) \\ & \text{s.t.} && \mathbf{x}_0 = \mathbf{x}(t_0), \\ & && \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \\ & && \tilde{\rho}_\pi(\mathbf{x}) \geq \kappa, \\ & && \underline{\xi} \leq \xi_k \leq \bar{\xi}, \forall k = \{0, 1, \dots, N-1\} \end{aligned} \quad (6)$$

It is important to note that the choice of the parameter λ in Section 3.4 significantly impacts the problem's computational complexity. Higher values of λ lead to

a larger number of Degree of Freedoms (DoFs), which in turn increases the computational burden of solving the optimization problem. Conversely, smaller values of λ impose constraints on the motion planner's DoFs, potentially resulting in trajectories that do not satisfy the STL specification. Furthermore, as demonstrated in [48], an appropriate value of κ can be chosen to ensure that the approximation error remains within a desired bound, such that $|\rho_\pi(\mathbf{x}) - \tilde{\rho}_\pi(\mathbf{x})| \leq \kappa$.

After the successful completion of the optimization process (6), the obtained solution can be transmitted to the onboard tracking controller (such as those described in [3, 16]) of the MRV for precise tracking and execution (see Figure 6).

4.3 Uncertainty-aware risk analysis

The motion planner introduced in this paper generates feasible trajectories for an MRV, enhancing ergonomic, safety, and comfort considerations in HRI, especially for tasks performed at heights, such as power line maintenance. Using an object handover task as a motivating example, we address the challenge of uncertainties in human pose, which can affect the satisfaction of the STL formula π and potentially lead to mission failure. To mitigate this, we propose a risk-aware analysis framework that assesses and quantifies the risks associated with deviations from STL specifications (2) in the trajectory derived from the optimization problem (6). This framework evaluates whether a given success rate (e.g., 80%) for satisfying STL specifications can still be achieved under uncertainty, drawing on methodologies presented in [38–40].

While the STL formula π , as defined in Section 4.1, traditionally applies to deterministic signals represented by the state sequence \mathbf{x} , we extend its interpretation to handle uncertainties using a stochastic process, denoted as Y (see Section 3.5). This extension accounts for variability in human pose, which directly influences critical mission elements such as the object handover (π_{ho}), the MRV's approach direction (π_{pr}), heading alignment (π_{vis}), and safety requirements like not approaching the operator from behind (π_{beh}). Human pose is modeled as uncertain, with a random distribution characterized by specific mean (μ_z) and covariance (σ_z) parameters – for example, using a normal distribution $\mathcal{N}(\mu_z, \sigma_z)$.

For any given realization $Y(\cdot, \varepsilon)$ of the stochastic process, we can evaluate whether the realization satisfies the STL formula π . However, in this stochastic context, satisfaction of π becomes probabilistic – some realizations of Y may satisfy π , while others may not. To manage this uncertainty, we apply risk measures, as described in Section 3.5, to quantify the likelihood that the stochastic process Y fails to satisfy the STL specification π . This approach provides a systematic way to handle uncertainty in HRI and ensures that mission objectives are met with a quantifiable level of confidence.

To compute the level of risk of not satisfying the STL formula π , the following steps are taken:

- (i) *Identify the stochastic elements Y in the mission specifications (see Section 4.1) that introduce uncertainty, such as human pose or environmental conditions.*
- (ii) *Compute the robustness score $\rho_\pi(\mathbf{x})$, which evaluates the degree to which the MRV satisfies the STL formula π under a specific realization $Y(\cdot, \varepsilon)$ of the*

stochastic process. The robustness score indicates how far the system is from violating or satisfying the mission objectives.

- (iii) *Apply an appropriate risk measure*, such as β -Value-at-Risk (VaR_β), to quantify the risk of not meeting the STL specification π . This provides a statistical threshold for the level of risk posed by uncertainties in the system.

Once the stochastic elements – such as human pose – are identified, we proceed to compute the robustness score $\rho_\pi(\mathbf{x})$ for each realization of the stochastic process Y . The STL formula π in (2) consists of two types of predicates: those evaluating whether the MRV's position belongs to a specific region (π_{ws} , π_{vr} , π_{pr} , π_{vel} , π_{pro} , π_{vis} , and π_{ho}) and those ensuring it avoids restricted regions (π_{obs} and π_{beh}). Given the structural similarities among these predicates, the analysis can focus on two key ones: π_{ho} , which defines the handover location (in (3i)), and π_{beh} , which ensures the MRV does not approach the human from behind (in (3c)). These predicates are critical because they establish the spatial boundaries that the MRV must either remain within or avoid for safe and efficient collaboration.

For the handover location, the robustness score $\rho_{\pi_{\text{ho}}}$ is calculated by measuring how close the MRV's current position is to the defined boundaries of the handover region. We compute the robustness score as follows:

$$\rho_{\pi_{\text{ho}}}(\mathbf{x}, Y(\cdot, \varepsilon)) = \min_{\mathbf{p}^{(j)}} \left(\min \left(\rho_{\pi^{(1)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \rho_{\pi^{(2)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \rho_{\pi^{(3)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \right. \right. \\ \left. \left. \rho_{\bar{\pi}^{(1)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \rho_{\bar{\pi}^{(2)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \rho_{\bar{\pi}^{(3)}}(\mathbf{x}, Y(\cdot, \varepsilon)) \right) \right), \quad (7)$$

with

$$\rho_{\pi^{(j)}}(\mathbf{x}, Y(\cdot, \varepsilon)) = \mathbf{p}_k^{(j)} - \underline{p}_{\text{ho}}^{(j)}(Y(\cdot, \varepsilon)), \quad \rho_{\bar{\pi}^{(j)}}(\mathbf{x}, Y(\cdot, \varepsilon)) = \bar{p}_{\text{ho}}^{(j)}(Y(\cdot, \varepsilon)) - \mathbf{p}_k^{(j)}.$$

Here, the terms $\underline{p}_{\text{ho}}^{(j)}(Y(\cdot, \varepsilon))$ and $\bar{p}_{\text{ho}}^{(j)}(Y(\cdot, \varepsilon))$ represent the lower and upper bounds of the handover location, modeled as stochastic variables. The robustness score measures the Euclidean distance between the MRV's current position $\mathbf{p}_k^{(j)}$ and these bounds, reflecting how uncertainty in the handover location impacts mission success.

Similarly, for the no-approach-from-behind predicate π_{beh} , the robustness score is calculated by taking the inverse of the minimum distance to the boundary, as shown below:

$$\rho_{\pi_{\text{beh}}}(\mathbf{x}, Y(\cdot, \varepsilon)) = \min_{\mathbf{p}^{(j)}} \left(- \min \left(\rho_{\pi^{(1)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \rho_{\pi^{(2)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \rho_{\pi^{(3)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \right. \right. \\ \left. \left. \rho_{\bar{\pi}^{(1)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \rho_{\bar{\pi}^{(2)}}(\mathbf{x}, Y(\cdot, \varepsilon)), \rho_{\bar{\pi}^{(3)}}(\mathbf{x}, Y(\cdot, \varepsilon)) \right) \right). \quad (8)$$

The terms $\rho_{\pi^{(j)}}(\mathbf{x}, Y(\cdot, \varepsilon))$ and $\rho_{\bar{\pi}^{(j)}}(\mathbf{x}, Y(\cdot, \varepsilon))$ in (8) are computed by replacing the handover location bounds in (7) with the corresponding bounds for the region behind the operator $\underline{p}_{\text{beh}}^{(j)}(\mathbf{x}, Y(\cdot, \varepsilon))$ and $\bar{p}_{\text{beh}}^{(j)}(\mathbf{x}, Y(\cdot, \varepsilon))$, respectively.

The robustness scores (7) and (8) reflect how uncertainty in human pose affects satisfaction of the STL specifications. If the robustness score is positive for a given realization $Y(\cdot, \varepsilon)$, the MRV satisfies the STL formula π . A zero or negative score indicates a failure to meet the specification. For simplifying the analysis, we assume

that the realization $Y(\cdot, \varepsilon)$ of the stochastic process Y is time-independent, meaning the human pose remains constant throughout the mission. In cases where the position changes, a new trajectory plan is computed for the MRV, as outlined in Section 4.4.

To evaluate the risk of mission failure under uncertainty, we apply a risk measure, such as VaR_β or its counterpart, β -Conditional-Value-at-Risk (CVaR_β), as outlined in Section 3.5. The VaR_β provides a threshold value, below which the robustness score $\rho_\pi(\mathbf{x})$ is unlikely to fall with a probability of at least $1 - \beta$. This measure quantifies the system's tolerance to uncertainty while still ensuring mission success. As described in Section 3.5, the formal definition of VaR_β is given by:

$$\text{VaR}_\beta(\rho_\pi(\mathbf{x})) := \inf\{\alpha \in \mathbb{R} | F_{\rho_\pi}(\alpha) \geq \beta\}, \quad (9)$$

where $F_{\rho_\pi}(\alpha)$ represents the CDF of the robustness scores ρ_π . However, since the exact CDF is often unknown and may require complex computations, such as high-dimensional integrals [38–40], we adopt a data-driven approach by estimating it using empirical data from K observed realizations of the stochastic process Y (e.g., human pose). The empirical CDF is computed as:

$$\hat{F}_{\rho_\pi}(\alpha, \rho_\pi(\mathbf{x})) := \frac{1}{K} \sum_{i=1}^K \mathbb{I}(\rho_\pi(\mathbf{x}) \leq \alpha), \quad (10)$$

where \mathbb{I} is the indicator function, which returns 1 if the robustness score for the i -th realization ($\rho_\pi(\mathbf{x})$) of Y is less than or equal to α , and 0 otherwise. Based on this empirical data (10), we can calculate upper and lower bounds for VaR_β at a confidence level $\delta \in (0, 1)$, as detailed in [38–40], as follows:

$$\overline{\text{VaR}}_\beta(\rho_\pi(\mathbf{x}), \delta) := \inf \left\{ \alpha \in \mathbb{R} | \hat{F}_{\rho_\pi}(\alpha, \rho_\pi(\mathbf{x})) - \sqrt{\frac{\ln(2/\delta)}{2K}} \geq \beta \right\}, \quad (11a)$$

$$\underline{\text{VaR}}_\beta(\rho_\pi(\mathbf{x}), \delta) := \inf \left\{ \alpha \in \mathbb{R} | \hat{F}_{\rho_\pi}(\alpha, \rho_\pi(\mathbf{x})) + \sqrt{\frac{\ln(2/\delta)}{2K}} \geq \beta \right\}. \quad (11b)$$

Thus, by observing K realizations of the stochastic process Y , and selecting an appropriate values for $\delta \in (0, 1)$, we can calculate upper and lower bounds for VaR_β . This method provides an estimate of the β -Value-at-Risk, and as the number of realizations K increases, the estimate becomes more accurate, converging to the true VaR_β value. This approach enables us to quantify the risk of the MRV failing to meet mission objectives under the inherent uncertainty in human pose.

4.4 Event-driven replanner

As described earlier, the motion planner presented in this paper ensures feasible trajectories for the MRV, prioritizing ergonomic, safety, and comfort considerations in HRI. However, during real-time operations, unexpected disturbances – such as battery issues, technical faults, or wind gusts – can cause significant deviations between the planned trajectory and the MRV's actual state. To address these deviations, we introduce an online, event-driven replanner.

When a failure event is detected, a new trajectory is computed online using the same motion planner described in Sections 4.1 and 4.2. The goal of this replanning process is to generate a new trajectory that reconnects the MRV to the previously optimized trajectory, ensuring minimal disruption to the mission.

Data is collected at specific discrete time instances, denoted by $\bar{\mathbf{t}}$. The event-driving period $T_e \in \mathbb{R}_{\geq 0}$ is a multiple of the sampling period T_s , and the “topic” waypoint period $T_g \in \mathbb{R}_{\geq 0}$ defines the frequency of low-rate state updates for the MRV ($T_g \gg T_s$). These discrete time instances are represented as vectors: $\bar{\mathbf{t}} = (\bar{t}_0, \dots, \bar{t}_L)^\top \in \mathbb{R}^{L+1}$ and $\hat{\mathbf{t}} = (\hat{t}_0, \dots, \hat{t}_G)^\top \in \mathbb{R}^{G+1}$, where $t_L \subseteq t_N$ and $t_G \subset t_N$. The term \bar{t}_l refers to the l -th element of the vector $\bar{\mathbf{t}}$, while \hat{t}_g refers to the g -th element of the vector $\hat{\mathbf{t}}$.

We define $\tilde{\mathbf{p}}$ as the actual position of the MRV during runtime, which may deviate from the optimal planned position \mathbf{p}^* due to disturbances. At each time instance $\bar{t}_l \in \bar{\mathbf{t}}$, the condition $|\tilde{\mathbf{p}}_l - \mathbf{p}_l^*| > \zeta$ is evaluated, where $\zeta \in \mathbb{R}_{>0}$ is a threshold designed to trigger an event. If this condition is met, an event is triggered, and the current MRV position is communicated to the ground station.

Upon receiving this information, the ground station (see Figure 6) performs an optimal replanning process for the time interval $[\bar{t}_l + t_c, \hat{t}_{g+1}]$, where t_c represents the maximum expected computation time for replanning, and \hat{t}_{g+1} refers to the next topic waypoint \mathbf{p}_{g+1} . The computation time t_c is estimated by running multiple instances of the STL optimization problem under varying conditions, such as different MRV initial positions, obstacle placements (π_{obs}), visibility requirements (π_{vis}), and operator preferences (π_{vis}). This ensures that the MRV computes a feasible trajectory from the triggered position \mathbf{p}_l to the next topic waypoint \mathbf{p}_{g+1} , mitigating discrepancies between the planned and actual trajectories.

It is important to note that the computational effort required for replanning is significantly lower than for initial planning. This is because the replanning occurs over a reduced time interval ($[\bar{t}_l + t_c, \hat{t}_{g+1}] \ll [0, T_N]$) and considers a smaller set of mission specifications. During replanning, only critical safety constraints must be met, such as remaining within the workspace (π_{ws}), avoiding obstacles (π_{obs}), and ensuring the MRV does not approach the operator from behind (π_{beh}) throughout the mission’s duration T_N . Other constraints, such as energy efficiency or precise trajectory smoothness, are deprioritized in favor of computational efficiency and rapid response to disturbances, ensuring the MRV can quickly return to a feasible operational state.

5 Simulation Results

To validate and assess the effectiveness of the proposed planning approach, a series of simulations were conducted. First, numerical simulations were performed using MATLAB to evaluate the performance of the planning algorithm and gain insights into its behavior. This initial phase allowed for the assessment of the planner without the need to explicitly model the MRV’s dynamics or the trajectory tracking controller. Following this, the simulations were extended to the Gazebo robotics simulator to validate the feasibility of the generated trajectories in more realistic settings. These simulations leveraged the advantages of Software-In-The-Loop (SITL)

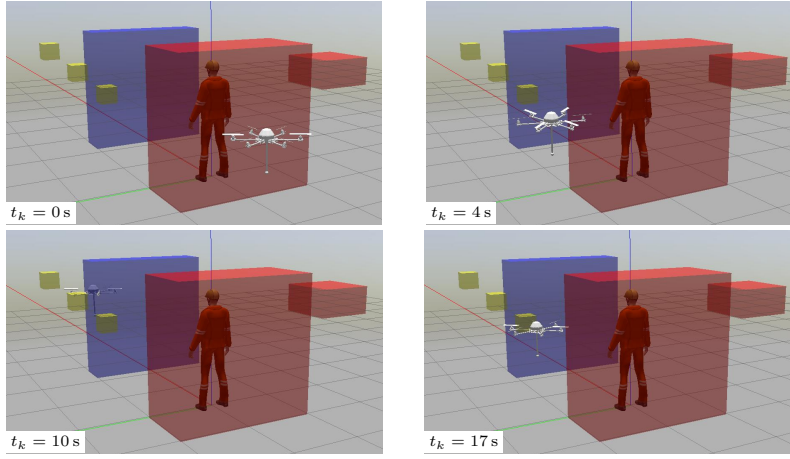


Fig. 7: Snapshots from Gazebo simulations illustrating a left-handed, top-to-bottom preferred approach direction. The blue, red, and yellow regions represent the designated location, no-fly zones, and approach areas, respectively.

simulations [19, 58], enabling the planner to be tested under conditions that closely approximate real-world scenarios, including detailed simulation of the MRV's dynamics and onboard low-level controllers. This transition from MATLAB to Gazebo offered a more comprehensive validation of the planning approach, considering both high-level trajectory generation and the dynamic response of the MRV in realistic environments.

The simulations were designed to validate several key aspects: (i) the adherence of planned trajectories to mission requirements; (ii) the capability of our approach to extend MRV endurance by comparing energy-aware solutions with those that do not incorporate energy considerations; (iii) the ability of the system to replan in response to unexpected disturbances; and (iv) the effectiveness of the proposed risk-aware analysis framework in assessing and quantifying risks associated with deviations from STL mission requirements under uncertain conditions.

The optimization algorithm was implemented using MATLAB R2019b, with the STL motion planner developed using the CasADi library² and solved with IPOPT³. Gazebo simulations were conducted with the GenoM robotics middleware [44] along with the TeleKyb3 software, available on the OpenRobots platform⁴. All numerical and Gazebo simulations were executed on a computer equipped with an i7-8565U processor (1.80 GHz) and 32GB of RAM, running on Ubuntu 20.04. For additional details and visual demonstrations of the simulation results, readers are encouraged to visit <https://mrs.felk.cvut.cz/stl-ergonomy-risk-analysis>. Figure 7 provides snapshots from the Gazebo simulations, illustrating key moments in the experiments.

² <https://web.casadi.org>

³ <https://coin-or.github.io/Ipopt/>

⁴ <https://git.openrobots.org/projects/telekyb3>

Table 3: Parameter values for the optimization problem.

Description	Symbol	Value	Description	Symbol	Value
Min motor forces	$\underline{\xi}$	0.29 N	Max motor forces	$\bar{\xi}$	11.5 N
Weight coefficient	w	0.50 [-]	Number of motors	N_p	6 [-]
MRV mass	m	2.25 kg	Inertia	\mathbf{J}	$\text{diag}\{2.07, 2.10, 3.10\} \times 10^{-2}$
Force cst. parameter	c_{ξ_i}	$11.5 \times 10^{-4} [-]$	Torque cst. parameter	c_{τ_i}	$2.38 \times 10^{-5} [-]$
Operator init. position	\mathbf{p}_{hum}	$\{0.0, 0.0, 0.0\} \text{m}$	Operator init. orientation	η_{hum}	$\{0.0, 0.0, 0.0\}^\circ$
MRV init. position	\mathbf{p}_0	$\{-1.80, 0.0, 1.0\} \text{m}$	MRV init. orientation	η_0	$\{0.0, 0.0, 0.0\}^\circ$
Number of samples	N	170 [-]	Sampling period	T_s	0.1 s
Smooth appr. parameter	λ	10 [-]	Mission duration	T_N	17 s
Max linear velocity	$\bar{\Gamma}_{\text{vel}}$	0.4 m s^{-1}	Min linear velocity	$\underline{\Gamma}_{\text{vel}}$	0 m s^{-1}
Max prop. speed	$\sqrt{\bar{\Gamma}_{\text{pro}}}$	80 Hz	Min prop. speed	$\sqrt{\underline{\Gamma}_{\text{pro}}}$	40 Hz
Level of confidence	δ	0.01 [-]	Heading manu. margin	γ	30°
Number obstacles	N_{obs}	1 [-]	Handover location	$\mathbf{p}_{\text{pr},3}$	$\{1.0, 1.0, 1.0\} \text{m}$
Number pref. regions	N_{pr}	3 [-]	Number of realizations	K	15 000 [-]
Replanning threshold	ζ	1 m	Event-driven period	T_e	0.5 s
Waypoint period	T_g	1 s	Computation time	t_c	0.4 s
Visibility time	T_{vr}	5 s	Robustness threshold	κ	0.2 [-]

5.1 Object handover scenario

The proposed planning strategy was evaluated using the object handover scenario outlined in Section 2. The simulation environment, depicted in Figure 8, measured $7.2 \text{ m} \times 5.7 \text{ m} \times 2.5 \text{ m}$ and included a human operator, an obstacle, and an MRV equipped with a rigidly attached stick to perform the handover task. As detailed in Section 2, the rigid attachment of the stick ensured no pendulum effect during flight, simplifying the trajectory planning process. This design choice allowed us to focus on optimizing human-robot collaboration and improving ergonomics through trajectory planning that respects vehicle dynamics, physical constraints, and ensure mission completion within predefined temporal requirements. Table 3 lists the key parameters used to configure the optimization problem. Notably, the mission duration (T_N) and visibility time (T_{vr}) were intentionally kept short to focus on the critical phases of the trajectory and avoid unnecessary analysis of extended waiting periods.

Figure 8 illustrates the planned trajectory, incorporating the operator's preference for a left-handed, top-to-bottom approach. The figure demonstrates that the trajectory is free of obstacles (π_{obs}) and avoids crossing into the non-approaching area behind the operator (π_{beh}), in compliance with the safety constraints. Additionally, the MRV follows the designated approach regions within the workspace \mathcal{F}_W , based on the operator's pose and preferences (π_{pr}), ensuring that these regions are crossed in the correct sequence. This ensures safe and efficient navigation while adhering to both ergonomic and mission-specific requirements. Solving the optimization problem, which integrates the STL requirements detailed in Section 4.1, required approximately 11 min.

Figure 9 presents data obtained from Gazebo simulations, demonstrating the planned trajectory's compliance with mission requirements. The graph demonstrates that comfort constraints, such as the drone's speed of approach (π_{vel}) and propeller velocity (π_{pro}), remain within their respective threshold values, Γ_{vel} , $\underline{\Gamma}_{\text{pro}}$ and $\bar{\Gamma}_{\text{pro}}$ ⁵.

⁵ To improve readability, the graph displays the propeller rotation speeds, incorporating bounds derived from motor force constraints ($\underline{\xi}$ and $\bar{\xi}$). Although these constraints are originally expressed in terms of forces, the corresponding propeller speeds were calculated by inverting the relationship between motor rotation and forces (see Section 3.1), and the resulting bounds were plotted accordingly. Similarly, for the propeller speed constraints ($\underline{\Gamma}_{\text{pro}}$ and $\bar{\Gamma}_{\text{pro}}$), originally defined as limits on squared motor speeds, the

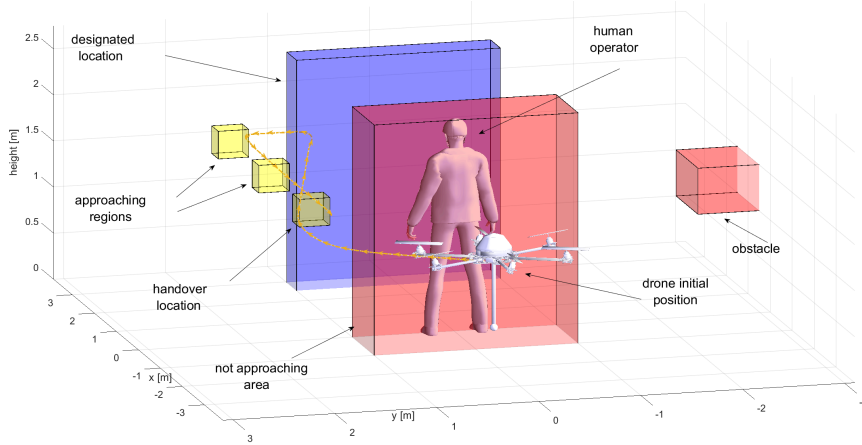


Fig. 8: Planned trajectory for the MRV, showing a left-handed, top-to-bottom preferred approach direction. Arrows indicate the paths followed by the MRV throughout the mission.

The visibility requirements are met within the appropriate time frames for reaching the designated location (π_{pr}) and the handover location (π_{ho}), represented by the blue and yellow time windows, respectively. Unlike the visibility requirement, reaching the handover location has no strict time constraint, which is why the duration of the yellow windows varies with different approach preferences. This flexibility allows the MRV to adapt its trajectory for smoother collaboration and improved ergonomic efficiency. The graph also confirms that the MRV's heading angle consistently aligns with its displacement direction (π_{vis}), meeting the directional alignment requirements. Furthermore, the MRV operates within its actuation limits, maintaining its lower and upper actuation bounds ($\underline{\xi}$ and $\bar{\xi}$) throughout the mission, thereby fully utilizing the platform's capabilities.

These results highlight the successful integration of ergonomic considerations, dynamic constraints, and physical actuation limits into the trajectory planning process. The proposed method ensures mission objectives are met while maintaining operator comfort, system safety, and efficient collaboration between humans and robots.

Lastly, as discussed in Sections 3.4 and 4.2, the STL optimization problem (4) was only solvable when using the smooth approximation $\tilde{\rho}_\pi(\mathbf{x})$ of the robustness function $\rho_\pi(\mathbf{x})$, due to the inherently NP-hard nature of the problem. This numerical evaluation highlights the importance of the smooth formulation (6), which effectively addresses the non-differentiable aspects of the max and min functions used to encode Boolean and temporal operators in the STL formula π (see Section 3.3), facilitating the identification of a local minimum.

bounds were converted to equivalent propeller rotation speeds. With a slight abuse of notation, both sets of constraints are represented on the same plot to simplify interpretation and enhance clarity.

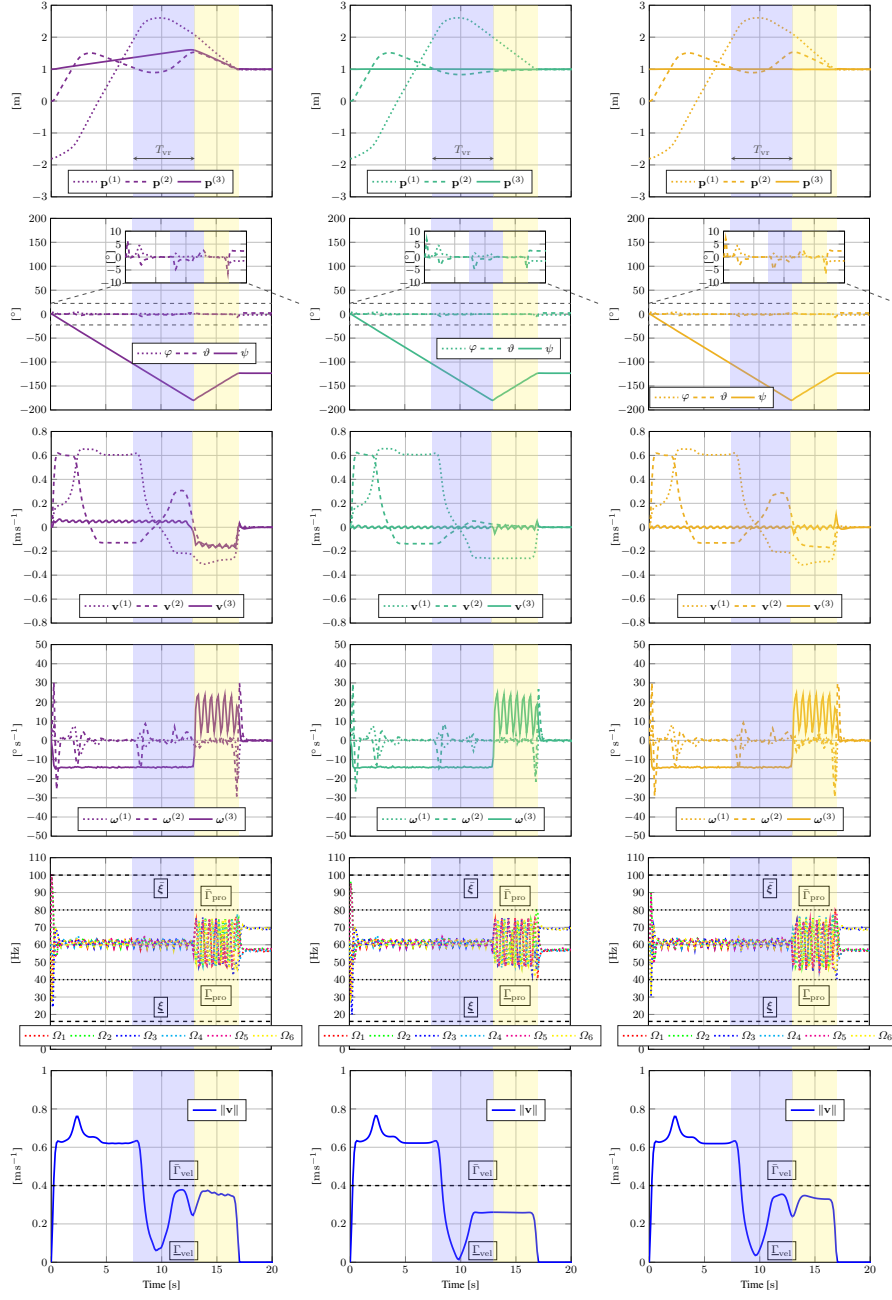


Fig. 9: Position, orientation, linear and angular velocities, propeller speed, and velocity magnitude for the planned trajectories based on different operator approach preferences: left-handed top-to-bottom approach (left), front trajectory (center), and left-sided approach (right). Blue and yellow segments indicate the time frames for meeting the visibility (T_{VT}) and handover requirements, respectively.

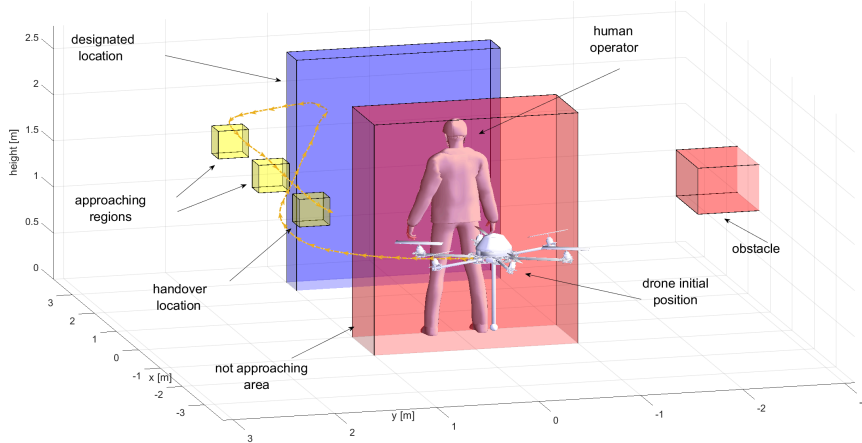


Fig. 10: Planned trajectory for the MRV, showing a left-handed, top-to-bottom preferred approach direction, with the energy term excluded from the optimization problem (i.e., setting $w = 0$ in (6)). Arrows indicate the paths followed by the MRV throughout the mission.

5.2 Energy-aware analysis and replanning strategy

This section highlights our approach’s effectiveness in extending MRV endurance by comparing energy-aware trajectories with those generated without energy optimization (i.e., setting $w = 0$ in (6)), as outlined in Section 4.2. Additionally, we evaluate the event-driven replanning mechanism’s performance in ensuring mission continuity amid unexpected disturbances, as detailed in Section 4.4. Results from MATLAB numerical simulations are presented in Figures 10 and 12.

Figure 10 illustrates that trajectories planned without the energy term $\mathcal{L}(\mathbf{x})$ in (6) tend to be longer and exhibit sharper direction changes compared to those incorporating energy (as shown in Figure 8). This difference arises because, with the inclusion of the energy term, the optimization problem (6) enforces stricter constraints on allowable actuator rotational speeds, as well as on linear and angular velocities, than in cases where energy considerations are omitted ($w = 0$). This approach enhances MRV endurance by reducing overall energy consumption while still meeting mission objectives.

Figure 11 compares energy consumption values between scenarios with and without the energy term $\mathcal{L}(\mathbf{x})$ in the STL optimization problem (6), based on data obtained from Gazebo simulations. The normalized energy term, along with MRV linear and angular velocities and propeller speeds, are plotted for both cases. As shown, trajectories without energy optimization result in higher linear and angular velocities and increased propeller speeds (left plots in Figure 11), making them less energy-efficient than energy-aware trajectories (right plots in Figure 11). Importantly, in both cases, the mission requirements encoded in the STL formula π are fully met. The constraint (4d) ensures a minimum robustness level κ to satisfy mission objectives, even when energy optimization might slightly reduce robustness values.

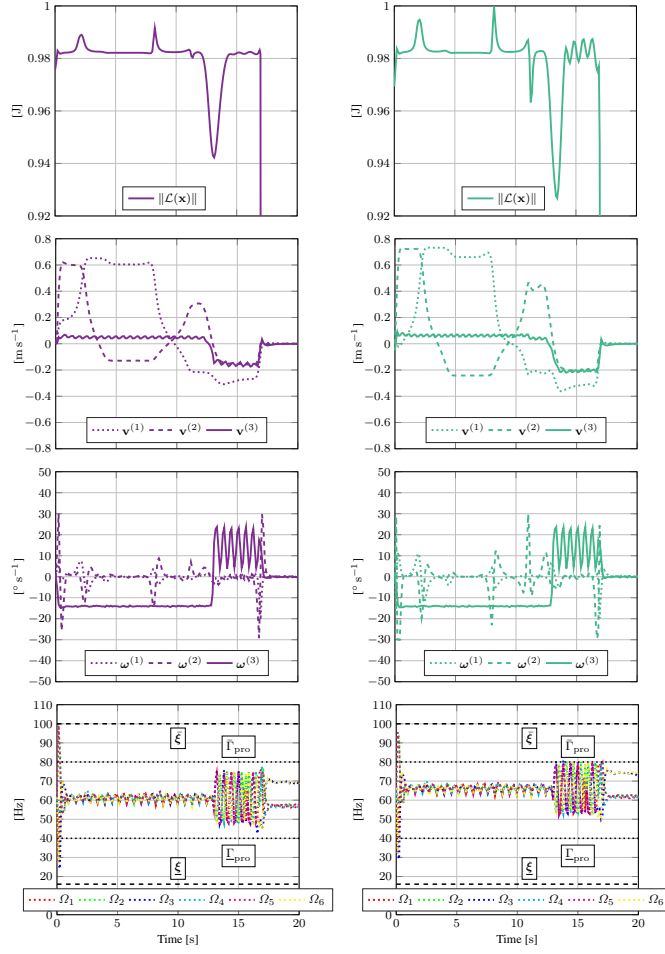


Fig. 11: Comparison of the normalized energy term, MRV linear and angular velocities, and propeller speed for scenarios where the energy term is either included (left) or excluded (right) in the optimization problem (6).

To assess the performance of the event-driven replanner, MATLAB simulations were conducted with unexpected disturbances that caused the MRV to deviate from its planned trajectory. These disturbances emulate real-world scenarios such as wind gusts, hardware malfunctions, or operator-induced variability, as outlined in Section 4.4. The replanner effectively detected significant deviations ($\|\tilde{\mathbf{p}}_t - \mathbf{p}_t^*\| > \zeta$), triggering a partial online replanning process to guide the MRV back to the next “topic” waypoint (see Figure 12). The newly computed trajectory served as a reference for the tracking controller (see Figure 6). The optimization process was completed in less than 1 s for all disturbance events, ensuring rapid trajectory updates.

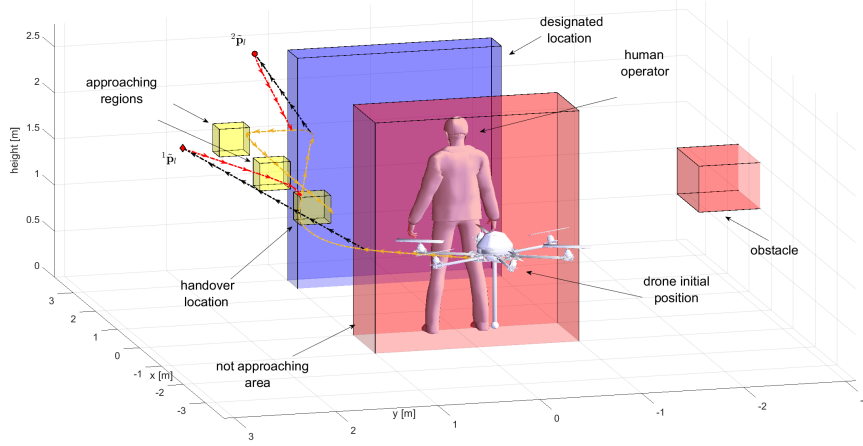


Fig. 12: Event-driven replanner trajectories for two disturbance scenarios, showing the MRV's left-handed approach. Black paths indicate deviations from the original trajectory, red paths show updated trajectories from the replanner, and red markers denote the MRV's position (${}^1\tilde{\mathbf{p}}_l$ and ${}^2\tilde{\mathbf{p}}_l$) when replanning is triggered.

5.3 Uncertainty-aware risk assessment

As outlined in Section 4.3, this section presents the results of the numerical assessment conducted to validate the proposed uncertainty-aware risk analysis framework. The evaluation quantifies risks associated with deviations from mission requirements, encoded in the STL formula π , for the trajectories derived from the planning problem (6) under uncertainty in human pose. Specifically, it investigates the extent to which the human operator can deviate from the planned handover position without violating mission requirements, thereby necessitating a new execution of the planning problem.

The simulations were conducted in MATLAB, where uncertainty in human pose was modeled as a stochastic process, Y , following a Gaussian distribution $\mathcal{N}(\mu_z, \sigma_z)$. This stochastic model captures variability in human position and orientation, which directly influences mission-critical predicates, including the handover location (π_{ho}), approach direction (π_{pr}), and the no-approach-from-behind constraint (π_{beh}). A total of $K = 15\,000$ realizations of the stochastic process Y were generated to simulate diverse human poses during the mission. This number balances the computational cost of generating $Y(\cdot, \varepsilon)$ and ensures reliable, non-conservative estimates.

Four random vectors, ${}^i\mathbf{p}_{hum}$, with $i = \{1, 2, 3, 4\}$, were employed to model variations in the human position, while the human attitude, $\boldsymbol{\eta}_{hum}$, was fixed to maintain consistent metrics for trajectory performance evaluation. Although a similar analysis could involve fixing the position and varying the attitude, human positional uncertainty was prioritized due to its greater impact on trajectory validity, as well as on ergonomic and comfort requirements. The random vectors for positional uncertainty

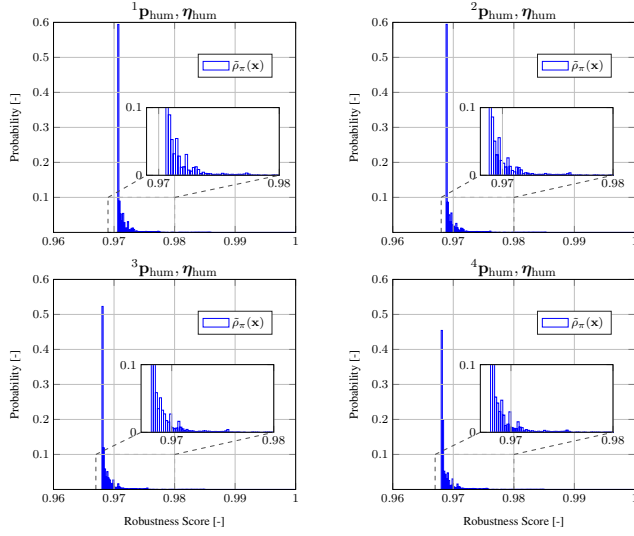


Fig. 13: Histogram of the smooth robustness score $\tilde{\rho}_\pi(\mathbf{x})$ of the specification π in (2), accounting for uncertainty in human pose (${}^i\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}}$) for the K realizations.

were defined as follows:

$$\begin{aligned} {}^1\mathbf{p}_{\text{hum}} &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.125 & 0 & 0 \\ 0 & 0.125 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right), & {}^2\mathbf{p}_{\text{hum}} &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right), \\ {}^3\mathbf{p}_{\text{hum}} &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right), & {}^4\mathbf{p}_{\text{hum}} &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.75 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right). \end{aligned} \quad (12)$$

The human attitude, $\boldsymbol{\eta}_{\text{hum}}$, was modeled using a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z)$, with mean vector $\boldsymbol{\mu}_z = (0.07, 0.07, 0.09)^\top$ and covariance $\boldsymbol{\sigma}_z = (0.09, 0.09, 0.52)^\top$.

Following the three-step process detailed in Section 4.3, the smooth robustness score $\tilde{\rho}_\pi(\mathbf{x})$ was computed using the trajectory presented in Section 5.1 and illustrated in Figure 8. This score quantifies the degree to which the MRV satisfies the STL formula π under a specific realization $Y(\cdot, \varepsilon)$ of the stochastic process, indicating how close the system is to either violating or satisfying the mission objectives. The resulting histograms of $\tilde{\rho}_\pi(\mathbf{x})$ for the K realizations are presented in Figure 13. The data show that for lower covariance values in the human position, the probability of achieving higher robustness scores increases. Conversely, as covariance increases, the likelihood of obtaining lower robustness scores rises.

Risk measures, including VaR_β , were applied to quantify the likelihood of not satisfying the STL formula π . Table 4 presents the upper and lower bounds of VaR_β for various risk levels β . The difference between these bounds, $|\overline{\text{VaR}}_\beta - \underline{\text{VaR}}_\beta|$, remains small across all β , indicating that the estimates are tight.

To interpret VaR_β , consider it as the threshold value below which the robustness score $\tilde{\rho}_\pi(\mathbf{x})$ is unlikely to fall with a probability of at least β . For example, a $\text{VaR}_{0.80}$

Table 4: Estimated VaR_β values for different risk levels β and K realizations, considering the human poses defined in (12).

	$\text{VaR}_{0.70}$	$\text{VaR}_{0.80}$	$\text{VaR}_{0.90}$	$\overline{\text{VaR}}_{0.70}$	$\overline{\text{VaR}}_{0.80}$	$\overline{\text{VaR}}_{0.90}$
$(^1\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$	0.95771	0.95811	0.95891	0.98429	0.98469	0.98549
$(^2\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$	0.95591	0.95631	0.95731	0.98249	0.98289	0.98389
$(^3\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$	0.95511	0.95571	0.95651	0.98169	0.98229	0.98309
$(^4\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$	0.95511	0.95571	0.95651	0.98169	0.98229	0.98309

Table 5: Estimated CVaR_β values for different risk levels β and K realizations, considering the human poses defined in (12).

	$\text{CVaR}_{0.70}$	$\text{CVaR}_{0.80}$	$\text{CVaR}_{0.90}$	$\overline{\text{CVaR}}_{0.70}$	$\overline{\text{CVaR}}_{0.80}$	$\overline{\text{CVaR}}_{0.90}$
$(^1\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$	0.015711	0.015311	0.014511	0.042289	0.041889	0.041089
$(^2\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$	0.017511	0.017111	0.016111	0.044089	0.043689	0.042689
$(^3\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$	0.018311	0.017711	0.016911	0.044889	0.044289	0.043489
$(^4\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$	0.018311	0.017711	0.016911	0.044889	0.044289	0.043489

of 0.95811 for $(^1\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$ indicates that there is an 80% chance the robustness score will not drop below this value. Values lower than this threshold represent a 20% risk of violating the STL specifications π , providing a quantifiable measure of the system's tolerance to uncertainty.

The histograms in Figure 13 and the values in Table 4 collectively suggest that poses with larger covariance, such as $(^3\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$ and $(^4\mathbf{p}_{\text{hum}}, \boldsymbol{\eta}_{\text{hum}})$, are less favorable. This is reflected in the lower robustness scores, indicating higher risks of not meeting mission objectives.

In Table 5, CVaR_β values are estimated for various risk levels β . Similar observations regarding human poses in (12) can be made. However, the risks are generally lower because CVaR considers only the tail of the $F_{\tilde{\rho}_\pi}(\tilde{\rho}_\pi(\mathbf{x}))$ (see Figure 4). Notably, the estimates CVaR_β of CVaR are less tight than those for VaR_β , with the difference $|\overline{\text{CVaR}}_\beta - \text{CVaR}_\beta|$ increasing significantly for larger β , primarily due to division by $1 - \beta$ (see Section 3.5).

This analysis highlights the extent to which planned trajectories from (6) remain compliant with mission requirements, even under human pose uncertainty. As deviations in human position increase, the robustness score $\tilde{\rho}_\pi(\mathbf{x})$ decreases, providing a clear measure of the system's sensitivity to variability. The proposed framework provides a systematic approach to evaluate trajectory resilience to uncertainty and quantify robustness levels, ensuring safer and more reliable HRI operations while maintaining adherence to mission requirements.

5.4 Gazebo simulations

Flight tests conducted in the Gazebo simulator (see Figure 7) successfully validated the object handover task encoded by the STL formula (2). The simulations demonstrated adherence to critical physical constraints, including motor force limits ($\underline{\xi}$ and $\bar{\xi}$), while ensuring compliance with MRV dynamics. Ergonomic and comfort fea-

tures, such as propeller velocity constraints (π_{vel}) and restrictions on the drone's approaching velocity (π_{pro}), were also thoroughly evaluated. Furthermore, the visibility requirement (π_{vis}) was assessed, ensuring the drone consistently aligned its heading with its direction of movement, allowing the operator to maintain continuous visual contact with the small object on the stick. Simulation results are summarized in Figure 9, with parameter values for the optimization problem detailed in Table 3.

The Gazebo simulation environment was designed to closely replicate the MATLAB scenario, ensuring consistency between both testing frameworks. In the simulation, the designated handover location was represented as a blue region, no-fly zones (including no-approach areas and obstacles) were highlighted in red, and approach regions defined by the operator were marked in yellow. For this test, the approach direction was specified as left-handed, with a top-to-bottom preference, aligned with mission parameters defined prior to execution.

The system architecture, depicted in Figure 6, integrates the STL-based motion planner, which solves the optimization problem (6) to generate feasible trajectories (\mathbf{x}^* , \mathbf{u}^*) for the MRV. This trajectory generation process is performed as a one-shot computation at the initial time t_0 . The computed trajectories then serve as reference inputs for the MRV trajectory tracking controller [3, 16], ensuring precise execution of the planned path.

Videos showcasing the Gazebo simulations are available at <https://mrs.felk.cvut.cz/stl-ergonomy-risk-analysis>, providing a comprehensive visualization of the scenarios and the system's capabilities.

6 Conclusions

This paper introduced a novel motion planning approach for human-robot collaboration using an MRV equipped with a rigidly attached long holder carrying a small object, with a focus on ergonomics, comfort, and efficiency for robotics co-workers. The planner leverages STL specifications to encode diverse mission objectives, including safety, temporal constraints, and human preferences, highlighting the flexibility and expressive power of this formal specification language. The approach formulates an optimization problem to generate dynamically feasible trajectories that meet mission requirements while accounting for vehicle dynamics and physical actuation limits.

Energy consumption is enhanced through an integrated energy-saving term, minimizing consumption and improving operational efficiency. An event-driven replanning strategy was incorporated, enabling real-time trajectory updates in response to unexpected disturbances, a critical requirement for real-world robotic missions. Additionally, a risk-aware analysis framework was developed to quantify and assess potential violations of STL specifications under uncertainties in human pose, providing criteria to determine when re-execution of the planning problem is necessary. The planner was validated through extensive simulations conducted in MATLAB and Gazebo, demonstrating its effectiveness in achieving safe and efficient collaboration.

Future research will focus on conducting field experiments in mock-up scenarios that closely replicate real-world conditions, further demonstrating the method's practicality and robustness beyond the Gazebo simulation environment. Additionally,

efforts will focus on integrating human operator fatigue into the problem formulation by introducing adaptive weights for Boolean and temporal operators, allowing dynamic modulation of robustness based on the operator's state. These advancements will further strengthen the proposed approach, ensuring safer, more efficient, and adaptable human-robot collaboration in practical deployments.

Declarations

Acknowledgments This paper builds upon the initial idea proposed in our previous work presented at the 2023 International Conference on Unmanned Aircraft Systems (ICUAS'23) [55]

Authors' contributions **Giuseppe Silano**: Conceptualization, Methodology, Formal analysis, Software, Simulations, Writing - original draft. **Amr Afifi**: Conceptualization, Software, Writing – review & editing. **Martin Saska**: Writing – review & editing, Supervision. **Antonio Franchi**: Writing – review & editing, Supervision.

Code or data availability Data sets generated during the current study are available from the corresponding author on reasonable request.

Consent for publication Informed consent was obtained from all the co-authors of this publication.

Competing interests The authors declare that they have no conflict of interest.

Ethics approval and consent to participate All applicable institutional and national guidelines were followed.

Funding This work was partially funded by the CTU grant no. SGS23/177/OHK3/3T/13, by the Czech Science Foundation (GAČR) under research project no. 23-07517S, by the European Union under the project Robotics and Advanced Industrial Production (reg. no. CZ.02.01.01/00/22 008/0004590), by the European Union's Horizon 2020 research and innovation programme AERIAL-CORE under grant agreement no. 871479, by the European Union's Horizon Europe research and innovation programme AUTOASSESS under grant agreement no. 101120732, and by the research fund for the Italian Electrical System (Ricerca di Sistema) through the decree n. 388 of November 6th, 2024.

References

1. Abbas, H., Fainekos, G.: Computing descent direction of MTL robustness for non-linear systems. In: 2013 American Control Conference, pp. 4405–4410 (2013). DOI 10.1109/ACC.2013.6580518. URL <https://ieeexplore.ieee.org/document/6580518>
2. Abbas, H., Fainekos, G., Sankaranarayanan, S., Ivančić, F., Gupta, A.: Probabilistic Temporal Logic Falsification of Cyber-Physical Systems. *ACM Transaction on Embedded Computing Systems* **12**(2s)

- (2013). DOI doi.org/10.1145/2465787.2465797. URL <https://dl.acm.org/doi/10.1145/2465787.2465797>
3. Afifi, A., van Holland, M., Franchi, A.: Toward Physical Human-Robot Interaction Control with Aerial Manipulators: Compliance, Redundancy Resolution, and Input Limits. In: 2022 International Conference on Robotics and Automation, pp. 4855–4861 (2022). DOI 10.1109/ICRA46639.2022.9812451. URL <https://ieeexplore.ieee.org/document/9812451>
 4. Ajoudani, A., Zanchettin, A.M., Ivaldi, S., Albu-Schffer, A., Kosuge, K., Khatib, O.: Progress and prospects of the human-robot collaboration. *Autonomous Robots* **42**(5), 957–975 (2018). DOI 10.1007/s10514-017-9677-2. URL <https://link.springer.com/article/10.1007/s10514-017-9677-2>
 5. Alcántara, A., Capitán, J., Torres-González, A., Cunha, R., Ollero, A.: Autonomous Execution of Cinematographic Shots With Multiple Drones. *IEEE Access* **8**, 201300–201316 (2020). DOI 10.1109/ACCESS.2020.3036239. URL <https://ieeexplore.ieee.org/document/9249238>
 6. Alpern, B., Schneider, F.B.: Defining liveness. *Information Processing Letters* **21**, 181–185 (1985). DOI 10.1016/0020-0190(85)90056-0. URL <https://www.sciencedirect.com/science/article/abs/pii/0020019085900560>
 7. Barbosa, F.S., Karlsson, J., Tajvar, P., Tumova, J.: Formal Methods for Robot Motion Planning with Time and Space Constraints. In: International Conference on Formal Modeling and Analysis of Timed Systems, pp. 1–10 (2021). DOI 10.1007/978-3-030-85037-1_1. URL https://link.springer.com/chapter/10.1007/978-3-030-85037-1_1
 8. Belta, C., Bicchi, A., Egerstedt, M., Frazzoli, E., Klavins, E., Pappas, G.J.: Symbolic planning and control of robot motion [Grand Challenges of Robotics]. *IEEE Robotics & Automation Magazine* **14**(1), 61–70 (2007). DOI 10.1109/MRA.2007.339624. URL <https://ieeexplore.ieee.org/document/4141034>
 9. Benzi, F., Brunner, M., Tognon, M., Secchi, C., Siegart, R.: Adaptive Tank-based Control for Aerial Physical Interaction with Uncertain Dynamic Environments Using Energy-Task Estimation. *IEEE Robotics and Automation Letters* **7**(4), 9129–9136 (2022). DOI 10.1109/LRA.2022.3190074. URL <https://ieeexplore.ieee.org/document/9826901>
 10. Berkenkamp, F., Schoellig, A.P., Krause, A.: Safe controller optimization for quadrotors with Gaussian processes. In: 2016 IEEE International Conference on Robotics and Automation, pp. 491–496 (2016). DOI 10.1109/ICRA.2016.7487170. URL <https://ieeexplore.ieee.org/document/7487170>
 11. Bertsekas, D.: Dynamic programming and optimal control. Athena Scientific (2012). URL <http://www.athenasc.com/dpbook.html>
 12. Butler, J., Agah, A.: Psychological effects of behavior patterns of a mobile personal robot. *Autonomous Robots* **10**, 185–202 (2001). DOI 10.1023/A:1008986004181. URL <https://link.springer.com/article/10.1023/A:1008986004181>
 13. Caballero, A., Silano, G.: A Signal Temporal Logic Motion Planner for Bird Diverter Installation Tasks with Multi-Robot Aerial Systems. *IEEE Access* **11**, 81361–81377 (2023). DOI 10.1109/ACCESS.2023.3300240. URL <https://ieeexplore.ieee.org/document/10197369>
 14. Cauchard, J.R., Dutau, C., Corsini, G., Cognetti, M., Sidobre, D., Lacroix, S., Brock, A.M.: Considerations for Handover and Co-working with Drones. In: Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, pp. 302–306 (2024). DOI 10.1145/3610978.3640590. URL <https://dl.acm.org/doi/10.1145/3610978.3640590>
 15. Cauchard, J.R., Kevin, J.L.E., Zhai, Y., Landay, J.A.: Drone & me: an exploration into natural human-drone interaction. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 361–365 (2015). DOI 10.1145/2750858.2805823. URL <https://dl.acm.org/doi/abs/10.1145/2750858.2805823>
 16. Corsini, G., Jacquet, M., Das, H., Afifi, A., Sidobre, D., Franchi, A.: Nonlinear Model Predictive Control for Human-Robot Handover with Application to the Aerial Case. In: 2022 IEEE International Conference on Intelligent Robots and Systems, pp. 7597–7604 (2022). DOI 10.1109/IROS47612.2022.9981045. URL <https://ieeexplore.ieee.org/document/9981045>
 17. Cortez, W.S., Verginis, C.K., Dimarogonas, D.V.: Safe, Passive Control for Mechanical Systems with Application to Physical Human-Robot Interactions. In: 2021 IEEE International Conference on Robotics and Automation, pp. 3836–3842 (2021). DOI 10.1109/ICRA48506.2021.9561981. URL <https://ieeexplore.ieee.org/document/9561981>
 18. Dautenhahn, K., Walters, M., Woods, S., Koay, K.L., Nehaniv, C.L., Sisbot, A., Alami, R., Simeon, T.: How may i serve you? a robot companion approaching a seated person in a helping context.

- In: Proceedings of the 1st ACM SIGCHI/SIGART conference on Humanrobot interaction, pp. 172–179 (2006). DOI 10.1145/1121241.1121272. URL <https://dl.acm.org/doi/10.1145/1121241.1121272>
19. Dimmig, C.A., Silano, G., McGuire, K., Gabellieri, C., Hönig, W., Moore, J., Kobilarov, M.: Survey of Simulators for Aerial Robots: An Overview and In-Depth Systematic Comparisons [Survey]. *IEEE Robotics & Automation Magazine* **32**(2), 153–166 (2025). DOI 10.1109/MRA.2024.3433171. URL <https://ieeexplore.ieee.org/document/10665978>
 20. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: International Conference on Formal Modeling and Analysis of Timed Systems, pp. 92–106. Springer (2010). DOI 10.1007/978-3-642-15297-9_9. URL https://link.springer.com/chapter/10.1007/978-3-642-15297-9_9
 21. D’Souza, D., Prabhakar, P.: On the expressiveness of MTL in the pointwise and continuous semantics. *International Journal on Software Tools for Technology Transfer* **9**, 1–4 (2007). DOI 10.1007/s10009-005-0214-9. URL <https://link.springer.com/article/10.1007/s10009-005-0214-9>
 22. Duan, H., Yang, Y., Li, D., Wang, P.: Human–robot object handover: Recent progress and future direction. *Biomimetic Intelligence and Robotics* **4**(1), 100145 (2024). DOI 10.1016/j.birob.2024.100145. URL <https://www.sciencedirect.com/science/article/pii/S2667379724000032>
 23. Durrett, R.: Probability: theory and examples, vol. 49. Cambridge university press (2019). URL <https://www.amazon.it/Probability-Theory-Examples-Rick-Durrett/dp/0521765390>
 24. Fainekos, G.E., Pappas, G.J.: Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science* **410**(42), 4262–4291 (2009). DOI 10.1016/j.tcs.2009.06.021. URL <https://www.sciencedirect.com/science/article/pii/S0304397509004149>
 25. Garrell, A., Coll, C., Alquézar, R., Sanfeliu, A.: Teaching a Drone to Accompany a Person from Demonstrations using Non-Linear ASFM. In: 2019 IEEE International Conference on Intelligent Robots and Systems, pp. 1985–1991 (2019). DOI 10.1109/IROS40897.2019.8967675. URL <https://ieeexplore.ieee.org/document/8967675>
 26. Garrell, A., Garza-Elizondo, L., Villamizar, M., Herrero, F., Sanfeliu, A.: Aerial social force model: A new framework to accompany people using autonomous flying robots. In: 2017 IEEE International Conference on Intelligent Robots and Systems, pp. 7011–7017 (2017). DOI 10.1109/IROS.2017.8206627. URL <https://ieeexplore.ieee.org/document/8206627>
 27. Gienger, M., Ruiken, D., Bates, T., Regaieg, M., Meibner, M., Kober, J., Seiwald, P., Hildebrandt, A.C.: Human-Robot Cooperative Object Manipulation with Contact Changes. In: 2018 IEEE International Conference on Intelligent Robots and Systems, pp. 1354–1360 (2018). DOI 10.1109/IROS.2018.8594140. URL <https://ieeexplore.ieee.org/document/8594140>
 28. Gilpin, Y., Kurtz, V., Lin, H.: A Smooth Robustness Measure of Signal Temporal Logic for Symbolic Control. *IEEE Control Systems Letters* **5**(1), 241–246 (2021). DOI 10.1109/LCSYS.2020.3001875. URL <https://ieeexplore.ieee.org/document/9114883>
 29. Haddadin, S., Suppa, M., Fuchs, S., Bodenmüller, T., Albu-Schäffer, A., Hirzinger, G.: Towards the robotic co-worker. In: C. Pradalier, R. Siegwart, G. Hirzinger (eds.) *Robotics Research*, pp. 261–282. Springer Berlin Heidelberg (2011). DOI 10.1007/978-3-642-19457-3_16. URL https://link.springer.com/chapter/10.1007/978-3-642-19457-3_16
 30. Hamandi, M., Usai, F., Sablé, Q., Staub, N., Tognon, M., Franchi, A.: Design of multirotor aerial vehicles: A taxonomy based on input allocation. *The International Journal of Robotics Research* **40**(8–9), 1015–1044 (2021). DOI 10.1177/02783649211025998. URL <https://journals.sagepub.com/doi/full/10.1177/02783649211025998>
 31. Herdel, V., Lee, J.Y., Cauchard, J.R.: Above and Beyond: A Scoping Review of Domains and Applications for Human-Drone Interaction. pp. 1–22 (2022). DOI 10.1145/3491102.3501881. URL <https://dl.acm.org/doi/10.1145/3491102.3501881>
 32. Kratky, V., Alcantara, A., Capitan, J., Stepan, P., Saska, M., Ollero, A.: Autonomous Aerial Filming With Distributed Lighting by a Team of Unmanned Aerial Vehicles. *IEEE Robotics and Automation Letters* **6**(4), 7580–7587 (2021). DOI 10.1109/LRA.2021.3098811. URL <https://ieeexplore.ieee.org/document/9495218>
 33. Kruse, T., Pandey, A.K., Alami, R., Kirsch, A.: Human-aware robot navigation: A survey. *Robotics and Autonomous Systems* **61**(12), 1726–1743 (2013). DOI 10.1016/j.robot.2013.05.007. URL <https://www.sciencedirect.com/science/article/pii/S0921889013001048>

34. Kshirsagar, A., Kress-Gazit, H., Hoffman, G.: Specifying and Synthesizing Human-Robot Handovers. In: 2019 IEEE International Conference on Intelligent Robots and Systems, pp. 5930–5936 (2019). DOI 10.1109/IROS40897.2019.8967709. URL <https://ieeexplore.ieee.org/document/8967709>
35. Li, W., Sadigh, D., Sastry, S.S., Seshia, S.A.: Synthesis for human-in-the-loop control systems. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pp. 470–484 (2014). DOI 10.1007/978-3-642-54862-8_40. URL https://link.springer.com/chapter/10.1007/978-3-642-54862-8_40
36. Licea, D.B., Silano, G., Hammouti, H.E., Ghogho, M., Saska, M.: Reshaping UAV-Enabled Communications with Omnidirectional Multi-Rotor Aerial Vehicles. *IEEE Communications Magazine* **63**(5), 94–100 (2025). DOI 10.1109/MCOM.001.2400421. URL <https://ieeexplore.ieee.org/document/10829762>
37. Lindemann, L., Dimarogonas, D.V.: Robust control for signal temporal logic specifications using discrete average space robustness. *Automatica* **101**, 377–387 (2019). DOI 10.1016/j.automatica.2018.12.022. URL <https://www.sciencedirect.com/science/article/pii/S0005109818306289>
38. Lindemann, L., Jiang, L., Matni, N., Pappas, G.J.: Risk of Stochastic Systems for Temporal Logic Specifications. *ACM Transactions on Embedded Computing Systems* **22**(3), 1–31 (2023). DOI 10.1145/3580490. URL <https://dl.acm.org/doi/full/10.1145/3580490>
39. Lindemann, L., Matni, N., Pappas, G.J.: STL Robustness Risk over Discrete-Time Stochastic Processes. In: 60th IEEE Conference on Decision and Control, pp. 1329–1335 (2021). DOI 10.1109/CDC45484.2021.9683305. URL <https://ieeexplore.ieee.org/document/9683305>
40. Lindemann, L., Rodionova, A., Pappas, G.J.: Temporal Robustness of Stochastic Signals. In: Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control, pp. 1–11 (2022). DOI 10.1145/3501710.3519504. URL <https://dl.acm.org/doi/abs/10.1145/3501710.3519504>
41. Loianno, G., Mulgaonkar, Y., Brunner, C., Ahuja, D., Ramanandan, A., Chari, M., Diaz, S., Kumar, V.: Autonomous flight and cooperative control for reconstruction using aerial robots powered by smartphones. *The International Journal of Robotics Research* **37**(11), 1341–1358 (2018). DOI 10.1177/0278364918774136. URL <https://journals.sagepub.com/doi/abs/10.1177/0278364918774136>
42. Majumdar, A., Pavone, M.: How should a robot assess risk? Towards an axiomatic theory of risk in robotics. *Robotics Research* pp. 75–84 (2020). DOI 10.1007/978-3-030-28619-4_10. URL https://link.springer.com/chapter/10.1007/978-3-030-28619-4_10
43. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, pp. 152–166. Springer, Berlin, Heidelberg (2004). DOI 10.1007/978-3-540-30206-3_12. URL https://link.springer.com/chapter/10.1007/978-3-540-30206-3_12
44. Mallet, A., Pasteur, C., Herrb, M., Lemaignan, S., Ingrand, F.: GenoM3: Building middleware-independent robotic components. In: IEEE International Conference on Robotics and Automation, pp. 4627–4632 (2010). DOI 10.1109/ROBOT.2010.5509539. URL <https://ieeexplore.ieee.org/document/5509539>
45. Nageli, T., Meier, L., Domahidi, A., Alonso-Mora, J., Hilliges, O.: Real-time planning for automated multi-view drone cinematography. *ACM Transactions on Graphics* **36**(4), 1–10 (2017). DOI 10.1145/3072959.3073712. URL <https://dl.acm.org/doi/10.1145/3072959.3073712>
46. Ollero, A., Tognon, M., Suarez, A., Lee, D., Franchi, A.: Past, Present, and Future of Aerial Robotic Manipulators. *IEEE Transactions on Robotics* **38**(1), 626–645 (2022). DOI 10.1109/TRO.2021.3084395. URL <https://ieeexplore.ieee.org/document/9462539>
47. Ortenzi, V., Cosgun, A., Pardi, T., Chan, W.P., Croft, E., Kulić, D.: Object Handovers: A Review for Robotics. *IEEE Transactions on Robotics* **37**(6), 1855–1873 (2021). DOI 10.1109/TRO.2021.3075365. URL <https://ieeexplore.ieee.org/document/9444288>
48. Pant, Y.V., Abbas, H., Mangharam, R.: Smooth operator: Control using the smooth robustness of temporal logic. In: 2017 IEEE Conference on Control Technology and Applications, pp. 1235–1240 (2017). DOI 10.1109/CCTA.2017.8062628. URL <https://ieeexplore.ieee.org/document/8062628>
49. Peternel, L., Kim, W., Babič, J., Ajoudani, A.: Towards ergonomic control of human-robot co-manipulation and handover. In: 2017 IEEE 17th International Conference on Humanoid Robotics, pp. 55–60 (2017). DOI 10.1109/HUMANOIDS.2017.8239537. URL <https://ieeexplore.ieee.org/document/8239537>

50. Petracek, P., Kratky, V., Petrlik, M., Baca, T., Kratochvil, R., Saska, M.: Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles. *IEEE Robotics and Automation Letters* **6**(4), 7596–7603 (2021). DOI 10.1109/LRA.2021.3098304. URL <https://ieeexplore.ieee.org/document/9492802>
51. Raman, V., Donzé, A., Maasoumy, M., Murray, R.M., Sangiovanni-Vincentelli, A., Seshia, S.A.: Model predictive control with signal temporal logic specifications. In: 53rd IEEE Conference on Decision and Control, pp. 81–87 (2014). DOI 10.1109/CDC.2014.7039363. URL <https://ieeexplore.ieee.org/document/7039363>
52. Rios-Martinez, J., Spalanzani, A., Laugier, C.: From proxemics theory to socially-aware navigation: A survey. *International Journal of Social Robotics* **7**, 137–153 (2015). DOI 10.1007/s12369-014-0251-1. URL <https://link.springer.com/article/10.1007/s12369-014-0251-1>
53. Rubagotti, M., Tusseyeva, I., Baltabayeva, S., Summers, D., Sandygulova, A.: Perceived safety in physical human–robot interaction — A survey. *Robotics and Autonomous Systems* **151**, 1–22 (2022). DOI 10.1016/j.robot.2022.104047. URL <https://www.sciencedirect.com/science/article/pii/S0921889022000173>
54. Shakhathreh, H., Sawalmeh, A.H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N.S., Khreishah, A., Guizani, M.: Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **7**, 48572–48634 (2019). DOI 10.1109/ACCESS.2019.2909530. URL <https://ieeexplore.ieee.org/document/8682048>
55. Silano, G., Afifi, A., Saska, M., Franchi, A.: A Signal Temporal Logic Planner for Ergonomic Human–Robot Collaboration. In: 2023 International Conference on Unmanned Aircraft Systems, pp. 328–335 (2023). DOI 10.1109/ICUAS57906.2023.10156559. URL <https://ieeexplore.ieee.org/document/10156559>
56. Silano, G., Baca, T., Penicka, R., Liuzza, D., Saska, M.: Power Line Inspection Tasks With Multi-Aerial Robot Systems Via Signal Temporal Logic Specifications. *IEEE Robotics and Automation Letters* **6**(2), 4169–4176 (2021). DOI 10.1109/LRA.2021.3068114. URL <https://ieeexplore.ieee.org/document/9384182>
57. Silano, G., Cabellero, A., Liuzza, D., Iannelli, L., Bogdan, S., Saska, M.: A Signal Temporal Logic Approach for Task-Based Coordination of Multi-Aerial Systems: a Wind Turbine Inspection Case Study. *Robotics and Autonomous Systems* **186**(4), 1–16 (2025). DOI 10.1016/j.robot.2024.104905. URL <https://www.sciencedirect.com/science/article/pii/S0921889024002896>
58. Silano, G., Oppido, P., Iannelli, L.: Software-in-the-loop simulation for improving flight control system design: a quadrotor case study. In: 2019 IEEE International Conference on Systems, Man and Cybernetics, pp. 466–471 (2019). DOI 10.1109/SMC.2019.8914154. URL <https://ieeexplore.ieee.org/document/8914154>
59. Sisbot, E.A., Alami, R.: A Human-Aware Manipulation Planner. *IEEE Transactions on Robotics* **28**(5), 1045–1057 (2012). DOI 10.1109/TRO.2012.2196303. URL <https://ieeexplore.ieee.org/document/6197743>
60. Sisbot, E.A., Marin, L.F., Alami, R.: Spatial reasoning for human robot interaction. In: 2007 IEEE International Conference on Intelligent Robots and Systems, pp. 2281–2287 (2007). DOI 10.1109/IROS.2007.4399486. URL <https://ieeexplore.ieee.org/document/4399486>
61. Sisbot, E.A., Marin-Urias, L.F., Alami, R., Simeon, T.: A Human Aware Mobile Robot Motion Planner. *IEEE Transactions on Robotics* **23**(5), 874–883 (2007). DOI 10.1109/TRO.2007.904911. URL <https://ieeexplore.ieee.org/document/4339546>
62. Staub, N., Mohammadi, M., Bicego, D., Delamare, Q., Yang, H., Prattichizzo, D., Robuffo Giordano, P., Lee, D., Franchi, A.: The Tele-MAGMaS: An Aerial-Ground Comanipulator System. *IEEE Robotics & Automation Magazine* **25**(4), 66–75 (2018). DOI 10.1109/MRA.2018.2871344. URL <https://ieeexplore.ieee.org/abstract/document/8511046>
63. Tognon, M., Alami, R., Siciliano, B.: Physical Human-Robot Interaction With a Tethered Aerial Vehicle: Application to a Force-Based Human Guiding Problem. *IEEE Transactions on Robotics* **37**(3), 723–734 (2021). DOI 10.1109/TRO.2020.3038700. URL <https://ieeexplore.ieee.org/document/9353246>
64. Tognon, M., Chávez, H.A.T., Gasparin, E., Sablé, Q., Bicego, D., Mallet, A., Lany, M., Santi, G., Revaz, B., Cortés, J., Franchi, A.: A Truly-Redundant Aerial Manipulator System With Application to Push-and-Slide Inspection in Industrial Plants. *IEEE Robotics and Automation Letters* **4**(2), 1846–1851 (2019). DOI 10.1109/LRA.2019.2895880. URL <https://ieeexplore.ieee.org/document/8629273>

65. Truc, J., Singamaneni, P.T., Sidobre, D., Ivaldi, S., Alami, R.: KHAOS: a Kinematic Human Aware Optimization-based System for Reactive Planning of Flying-Coworker. In: 2022 International Conference on Robotics and Automation, pp. 4764–4770 (2022). DOI 10.1109/ICRA46639.2022.9811803. URL <https://ieeexplore.ieee.org/document/9811803>
66. Webster, M., Western, D., Araiza-Illan, D., Dixon, C., Eder, K., Fisher, M., Pipe, A.G.: A corroborative approach to verification and validation of human–robot teams. *The International Journal of Robotics Research* **39**(1), 73–99 (2020). DOI 10.1177/0278364919883333. URL <https://journals.sagepub.com/doi/full/10.1177/0278364919883333>
67. Wojciechowska, A., Frey, J., Sass, S., Shafir, R., Cauchard, J.R.: Collocated Human-Drone Interaction: Methodology and Approach Strategy. In: 2019 14th IEEE International Conference on Human-Robot Interaction, pp. 172–181 (2019). DOI 10.1109/HRI.2019.8673127. URL <https://ieeexplore.ieee.org/document/8673127>