Safe Human-UAS Collaboration From High-Level Planning to Low-Level Tracking

Hossein Rastgoftar

***Abstract*— This paper studies the problem of safe human-uncrewed aerial system (UAS) collaboration in a shared work environment. By considering human and UAS as co-workers, we use Petri Nets to abstractly model evolution of shared tasks assigned to human and UAS co-workers. Particularly, the Petri Nets' "places" represent work stations; and therefore, the Petri Nets' transitions can formally specify displacements between the work stations. The paper's first objective is to incorporate uncertainty regarding the intentions of human co-workers into motion planning for UAS, when UAS closely interacts with human co-workers. To this end, the proposed Petri Nets model uses "conflict" constructs to represent situations at which UAS deals with incomplete knowledge about human co-worker intention. The paper's second objective is then to plan the motion of the UAS in a resilient and safe manner, in the presence of non-cooperative human co-workers. In order to achieve this objective, UAS equipped with onboard perception and decision-making capabilities are able to, through real-time processing of in-situ observation, predict human intention, quantify human distraction, and apply a non-stationary Markov Decision Process (MDP) model to safely plan UAS motion in the presence of uncertainty. Given the current and next UAS waypoints, assigned by the MDP planner, the paper applies Potryagin's minimal principle to plan the desired trajectory of the UAS and uses a feedback linearaztion trajectory control to enable UAS with stable tracking of the desired trajectory.**

***Note to Practitioners*—Despite advances in aerial robotics, there are still significant barriers for their integration and application into human-centered jobs. Safety-related concerns, potential hazards to labor, and limited mission duration are some major barriers for using such a helpful technology. The main goal of this paper is to come up with a reliable solution for long-term collaboration between humans and unmanned aerial system (UAS) for safe and efficient accomplishment of a human-centered work. To this end, we propose to use Petri Nets to abstractly model and specify human-robot collaboration and effectively plan the tasks assigned to UAS and human co-workers. In this context, human co-workers can do some tasks that are difficult and somehow unsafe to be carried out by UAS. On the other hand, UAS can be deployed to monitor the work environment, recognize human wellness, or carry some small payloads. UAS co-workers enabled with on-board perception capabilities can also predict human coworker intention and quantify human co-worker distraction in real time through online processing of in-situ observations. Furthermore, UAS can safely plan its motion without human intervention or supervision, by combining Markov Decision Process, optimal control, and trajectory tracking control models.**

***Index Terms*— Decision-making, Markov decision process (MDP), human intention prediction, unscrewed aerial system (UAS), Petri Nets.**

I. Introduction

WITH advances in artificial intelligence, learning, and optimization approaches, copter-powered uncrewed aerial systems (UAS) have found numerous applications in agriculture [1], [2], warehouse industries [3], [4], mining [5], [6], surveillance [7], [8], construction [9], [10], and many other fields. Despite the widespread UAS applications, to date, UAS operation is closely supervised by human professionals. This paper, on the other hand, studies semi-autonomous missions that are jointly operated by UAS and human in the same environment. The primary focus of the paper is on ensuring safety of UAS operation in shared workplaces where humans and UAS coexist. Towards this goal, we combine Petri Nets with non-stationary Markov Decision Process (MDP) and Pontryagin's minimum principle to provide a framework for safe human-UAS collaboration.

A. Related Work

In the literature, Petri Nets have been used to model evolution of discrete-event systems and applied for a variety of applications. Manufacturing systems have been the primary applications of Petri Nets [11], [12]. In [13], timed Petri Nets are used to model the integration of virtual sensors proposed for electric mobility services. In [14], variable Petri Nets are proposed to describe "system connectivity," "interaction soundness and data validity", and modeling and analysis of interactive systems. Petri Nets are used in [15] to model game flow and student learning optimization as the main application. Researchers have used Petri Nets to optimally schedule tasks for humans and robots cooperating in a shared workplace [16], [17], [18], [19], [20]. Authors in [21] used colored Petri Nets (CPN) for validation and verification of safety-critical systems [21]. CPN was also used in [22] to model human actions, in a human-robot collaboration, under partial observability assumption.

This paper considers a multicopter drone as the robot interacting with humans in a shared workplace. Application

Received 13 July 2024; accepted 8 September 2024. Date of publication 4 October 2024; date of current version 14 March 2025. This work was supported by the National Science Foundation under Award 2133690 and Award 1914581. This article was recommended for publication by Associate Editor Q. Chang and Editor J. Yi upon evaluation of the reviewers' comments.

The author is with the Department of Aerospace and Mechanical Engineering, The University of Arizona, Tucson, AZ 85721 USA (e-mail: hrastgoftar@arizona.edu).

Digital Object Identifier 10.1109/TASE.2024.3461726

of drones as safety inspectors in construction site was studied in [23]. In [24], the authors rely on user experiences to incorporate human emotions into drone's trajectory planning. Refs. [25] and [26] study potential human-drone communication interfaces in a collaborative environment. Signal temporal logic [27], [28] is used in [29] to plan ergonomic and safe human-UAS collaboration in a construction environment.

B. Contributions

In the literature, researchers have extensively investigated autonomous UAS operation applications or UAS operation supervised by human professionals. This paper, on the other hand, aims to develop a model for semi-autonomous operations that are jointly conducted by UAS and humans in a shared workplace. By considering humans and UAS as co-workers, sharing a work environment, safety assurance becomes highly important since human and UAS co-workers can closely interact. Because UAS needs to deal with the uncertainty associated with incomplete knowledge about the intentions of each human co-worker, safety assurance is indeed very challenging. To address this problem, this paper proposes to apply Petri Nets to model task evolution in a semi-autonomous operation, conducted by human and UAS co-workers. This novel application of Petri Nets assumes that multiple Work Stations (WSs) exist in a shared workplace and uses the Petri Nets' places to abstractly represent WSs. Therefore, the Petri Nets's transitions either specify evolution of tasks in the same WS, or displacement between the WSs. Particularly, we use "cyclic" construct to model evolution of incomplete task, or progression of multiple tasks, in the same place, when change of a WS is not needed. On the other hand, we use "sequential," "dependency," and "conflict" to model displacement of UAS, or human, between WSs.

While existing work, related to Petri Nets' applications, mainly focuses on high-level specification of task flow, this paper applies Petri Nets, MDP method, Pontryagin minimum principle, and control theory to specify high-level tasks, safely plan high-level actions, determine desired UAS trajectory, and ensure stable tracking of the desired trajectory by a multicopter UAS, where a high-fidelity nonlinear dynamics is used to model multicopter UAS motion. By applying the proposed model for abstraction of human-UAS collaboration, the paper offers the following novel contributions:

1) We enable UAS with on-board perception capability to learn human intention and quantify human co-worker distraction by real-time processing of in-situ observations. By predicting human co-worker intention, UAS can resolve "conflict" situations resulted from incomplete knowledge about human intentions. Also, UAS can quantify distraction of every human co-worker by specifying the probability distribution over a moving neighboring set (MNS). MNS is defined as a finite set of cells that form a rigid rectangular zone around each human coworker's desired position, where the human's desired position is along a desired trajectory that is estimated using the available search methods, such as A* search [30], [31]. Note that the A* search is indeed applied to obtain the shortest path between two WSs. We consider this path as the reference to quantify the human distraction when traveling between two workstations.
2) We develop a non-stationary MDP model for UAS motion planning in the presence of multiple non-cooperative human co-workers. While the state space, transition function, discount factor, and action components of the proposed MDP are time-invariant, the MDP cost is time-varying, and it is consistently updated so that learning of human intention and co-worker distraction are properly incorporated in the UAS motion planning. We use value iteration method to obtain optimal UAS actions by solving the Bellman equation.
3) The paper applies the Pontryagin's minimum principle to establish an interface between the high-level MDP-based motion planning and low-level trajectory tracking by obtaining the UAS desired trajectory through solving a fixed-final-state and fixed-final-time optimal control problem.

C. Outline

This paper is organized as follows: Preliminary notions of Petri Nets are reviewed in Section II. The problem of UAS-Human collaboration is formulated in Section III and followed by the paper's approach for UAS onboard perception and motion planning in Section IV. Simulation results are presented in Section V. Concluding remarks are stated in Section VI.

II. Preliminaries

We use Petri Nets to model UAS-human collaboration in a constrained workplace. The proposed Petri Nets is defined by tuple $\mathcal{PN} = (\mathcal{P}, \mathcal{T}, \mathcal{E}_H, \mathcal{E}_U, \mathcal{W}_H, \mathcal{W}_U, \mathcal{M}_H, \mathcal{M}_U)$, where $\mathcal{P}$ is a finite set of places; $\mathcal{T}$ is a finite set of transitions; $\mathcal{E}_H \subset (\mathcal{P} \times \mathcal{T}) \bigcup (\mathcal{T} \times \mathcal{P})$ and $\mathcal{E}_U \subset (\mathcal{P} \times \mathcal{T}) \bigcup (\mathcal{T} \times \mathcal{P})$ define unweighted arcs of human and UAS co-workers, respectively; $\mathcal{W}_H : \mathcal{E}_H \to \mathbb{N}$ defines weights of human co-worker transitions; $\mathcal{W}_U : \mathcal{E}_U \to \mathbb{N}$ defines weights of UAS co-worker transitions; marking $\mathcal{M}_H : \mathcal{P} \to \mathbb{N}$ specifies distribution of human co-workers; and marking $\mathcal{M}_U : \mathcal{P} \to \mathbb{N}$ specifies distribution of UAS co-workers. In this work, set $\mathcal{P}$ defines actual Work Stations (WSs), therefore, $\mathcal{T}$ defines possible displacements between WSs.

Assumption 1: This paper assumes that no bias exists for activating every possible transition $\zeta \in \mathcal{T}$ by the human at every place $p \in \mathcal{P}$, where $(p, \zeta) \in \mathcal{E}_H$. Therefore,

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{\zeta \in \mathcal{T}} \bigwedge_{(p,\zeta) \in \mathcal{E}_H} (\mathcal{W}_H(p, \zeta) = 1). \tag{1}$$

Assumption 2: This paper assumes that no bias exists for activating every possible transition $\zeta \in \mathcal{T}$ by the UAS at every place $p \in \mathcal{P}$, where $(p, \zeta) \in \mathcal{E}_U$. Therefore,

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{\zeta \in \mathcal{T}} \bigwedge_{(p,\zeta) \in \mathcal{E}_U} (\mathcal{W}_U(p, \zeta) = 1). \tag{2}$$

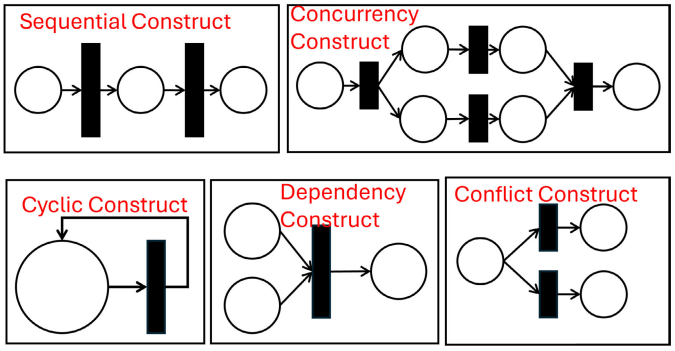


Fig. 1. The constructs are used for modeling human-UAS collaboration.

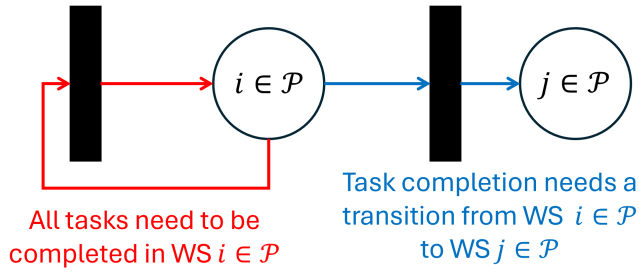


Fig. 2. Tasks are classified into two groups. Group one represents the tasks that are specified by $\mathcal{E}'$ and do not require a human or UAS to change the WS. On the other hand, group two represents tasks defined by $\mathcal{E}_H \setminus (\mathcal{E}' \cap \mathcal{E}_H)$ or $\mathcal{E}_U \setminus (\mathcal{E}' \cap \mathcal{E}_U)$ and require human or UAS co-workers two transition between to workstations.

Definition 1: Given human arc set $\mathcal{E}_H$, we define out-neighbor transition set

$$\mathcal{N}_H(p) = \{\zeta \in \mathcal{T} : (p, \zeta) \in \mathcal{E}_H\}, \qquad p \in \mathcal{P}, \tag{3}$$

to specify possible transitions that can be fired by human co-workers at $p \in \mathcal{P}$.

Definition 2: Given UAS arc set $\mathcal{E}_U$, we define out-neighbor transition set

$$\mathcal{N}_U(p) = \{\zeta \in \mathcal{T} : (p, \zeta) \in \mathcal{E}_U\}, \qquad p \in \mathcal{P}, \tag{4}$$

to specify possible transitions that can be fired by UAS at $p \in \mathcal{P}$.

Definition 3: Set

$$\mathcal{E}' = \{(p, \zeta) \cup (\zeta, p) : \forall p \in \mathcal{P}, \ \forall \zeta \in \mathcal{T}\} \tag{5}$$

defines cyclic arcs (See the cyclic arc schematic in the bottom-left picture of Fig. 1).

Note that $\mathcal{E}'$ is used to specify tasks that can be performed in the same workstation without requiring the UAS or a human coworker to switch workstations. The remaining tasks requiring change of WS are specified by $\mathcal{E}_H \setminus (\mathcal{E}' \cap \mathcal{E}_H)$, for human co-workers, and $\mathcal{E}_U \setminus (\mathcal{E}' \cap \mathcal{E}_U)$ for the UAS (See Fig. 2).

Definition 4: Set

$$\mathcal{R}_H(p) = \left\{p' \in \mathcal{P} : (\zeta, p') \in \mathcal{E}_H \setminus (\mathcal{E}' \cap \mathcal{E}_H), \zeta \in \mathcal{N}_H(p)\right\}, \ p \in \mathcal{P}, \tag{6}$$

defines all possible next WSs for *human* co-workers at $p \in \mathcal{P}$.

Definition 5: Set

$$\mathcal{R}_U(p) = \left\{p' \in \mathcal{P} : (\zeta, p') \in \mathcal{E}_U \setminus (\mathcal{E}' \cap \mathcal{E}_U), \ \zeta \in \mathcal{N}_U(p)\right\}, \ p \in \mathcal{P} \tag{7}$$

defines all possible next WSs for *UAS* co-workers at $p \in \mathcal{P}$.

We note that $\mathcal{R}_H(p) = \emptyset$, if the task proceeding in $p \in \mathcal{P}$ does not require human co-workers to leave $p \in \mathcal{P}$. Similarly, $\mathcal{R}_U(p) = \emptyset$, if a UAS co-worker does not require to change its current WS for executing the assigned task at $p \in \mathcal{P}$.

The Petri Nets, used for the abstraction of human UAS collaboration, consist of five possible constructs that are shown in Fig. 1. Functionality of these constructs are described below:

Cyclic Construct: The task progress in $p \in \mathcal{P}$ is represented by the "cyclic construct". Set $\mathcal{E}'$, given by Eq. (5), defines the non-weighted arcs that represent progression of tasks in the same WS.

Sequential Construct: The picture on the top-left of Fig. 1 illustrates the "sequential construct". A sequential construct represents situations at which the transition of the human co-worker or UAS co-worker is known because the worker has only one possible target WS to follow next.

Conflict Construct: The conflict construct is used to incorporate UAS uncertainty associated with the human decision/intention to choose the next place $p' \in \mathcal{R}_H(p)$. For place $p \in \mathcal{P}$ dealing with a conflict construct, $|\mathcal{R}_H(p)| > 1$.

Dependency Construct: The "dependency construct" represents a situation at which inputs from several incoming locations are needed to fire a transition $\zeta \in \mathcal{T}$.

Concurrency Construct: The "concurrency construct" shown in Fig. 1 represents a situation at which a human co-worker, or UAS, is conducting multiple tasks simultaneously.

We define the following rules to activate Petri Nets' transitions:

Rule 1: The marking functions are non-negative, i.e. $\mathcal{M}_H(p) \geq 0$ and $\mathcal{M}_U(p) \geq 0$ at every place $p \in \mathcal{P}$. If $M_H(p) = 0$, there is no human co-worker in place $p \in \mathcal{P}$. Otherwise, $M_H(p) > 0$ assigns the number of human co-workers in place $p \in \mathcal{P}$. Similarly, $\mathcal{M}_U(p) \geq 0$ specifies the number of UAS co-worker in place $p \in \mathcal{P}$.

Rule 2: Transition ζ is fired for a human co-worker at $p \in \mathcal{P}$, if the following

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{\zeta \in \mathcal{T}} \bigwedge_{(p,\zeta) \in \mathcal{E}_H} (\mathcal{M}_H(p) - \mathcal{W}_H(p, \zeta) \geq 0), \tag{8}$$

where $\bigwedge$ means "include all".

Rule 3: Transition ζ is fired for a UAS co-worker at $p \in \mathcal{P}$, if the following

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{\zeta \in \mathcal{T}} \bigwedge_{(p,\zeta) \in \mathcal{E}_U} (\mathcal{M}_U(p) - \mathcal{W}_U(p, \zeta) \geq 0), \tag{9}$$

where $\bigwedge$ means "include all".

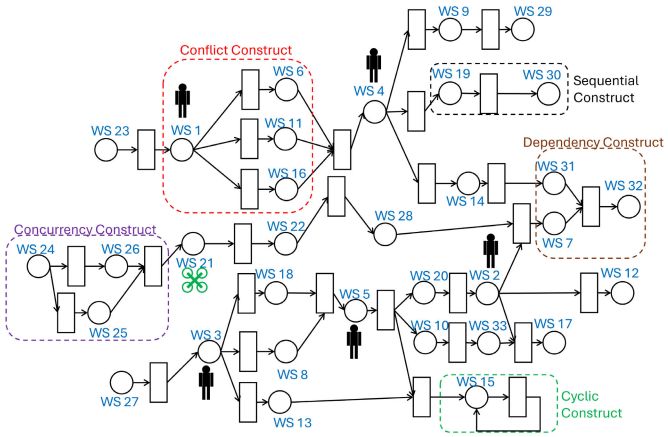


Fig. 3. Example Petri Nets with places representing actual WSs and transitions representing displacement between WSs.

The marking functions $\mathcal{M}_H(p)$ and $\mathcal{M}_U(p)$ are both defined over the place set $\mathcal{P}$. On the other hand, $\mathcal{W}_H(p,\zeta)$ and $\mathcal{W}_U(p,\zeta)$ can be considered as transition functions quantifying cost of transition from place $p \in \mathcal{P}$ to $\zeta \in \mathcal{T}$, thus needing to be defined over $\mathcal{P} \times \mathcal{T}$.

Rule 4: The "dependency" construct's transition uses logical "or" reasoning to approve a potential displacement. This implies that, in order for a "dependency" transition $\zeta \in \mathcal{T}$ to occur, there must be a human or UAS coworker present at $p \in \mathcal{P}$, where $(p,\zeta) \in \mathcal{E}_U \cup \mathcal{E}_H$.

Rule 5: To ensure safety of the human co-workers, UAS co-workers can be present at WS $p \in \mathcal{P}$, if the following condition is met:

$$\mathcal{M}_U(p) + \mathcal{M}_H(p) = \begin{cases} \leq 2 & \mathcal{M}_U(p) > 0 \\ \geq 0 & \mathcal{M}_U(p) = 0, \end{cases} \qquad \forall p \in \mathcal{P}. \tag{10}$$

By imposing condition (10), the following safety regulations are imposed for the presence of UAS and human co-workers in WS $p \in \mathcal{P}$:

Regulation 1: No constraints are imposed on the number of human co-workers in WS $p \in \mathcal{P}$, if no UAS exists in $p \in \mathcal{P}$.

Regulation 2: The number of UAS and human co-workers cannot exceed 2 in WS $p \in \mathcal{P}$, if one UAS exists in WS $p \in \mathcal{P}$.

Regulation 3: At most 2 UAS co-workers can work in WS $p \in \mathcal{P}$, if no human coworker is present in WS $p \in \mathcal{P}$.

To better clarify the above notations, Fig. 3 shows an example of Petri Nets with place set $\mathcal{P} = \{1, \cdots, 33\}$ defining 33 WSs in a shared motion space. The Petri Nets consist of the five constructs shown in Fig. 1. In the illustrated Petri Nets, places representing WSs are shown by circles, and transitions are shown by rectangles. The Petri Nets contain sequential, dependency, concurrency, conflict, and cyclic constructs.

III. Problem Statement

We consider operation of UAS in the presence of N_H human co-workers identified by set $\mathcal{H} = \{1, \cdots, N_H\}$ in a shared workplace. The entire workplace is spatially discretized and abstractly represented by finite set $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ where $\times$ is the Cartesian product symbol; $\mathcal{X} = \{1, \cdots, n_x\}$ and $\mathcal{Y} = \{1, \cdots, n_y\}$ identify the x and y coordinates of the cells, respectively. We also define $\mathcal{O} \subset \mathcal{D}$ as stationary obstacles in the workplace that cannot be reached neither by UAS or a human co-worker. The WS positions are defined by finite set

$$\bar{\mathcal{D}} = \{\bar{\mathbf{r}}_i = (\bar{x}_i, \bar{y}_i) \in \mathcal{D} \setminus \mathcal{O} : i \in \mathcal{P}\}. \tag{11}$$

We consider scenarios at which the UAS co-worker aims to reach $j \in \mathcal{R}_U(i)$ from $i \in \mathcal{P}$, where the UAS motion can be abstracted by a sequential construct through firing transition $\zeta_u \in \mathcal{T}$. Therefore, $(i,\zeta_u) \bigcup (\zeta_u, j) \in \mathcal{E}_U$. The paper focuses in particular on scenarios that the UAS models human coworker behavior using a conflict construct when the humans' decisions are uncertain to UAS.

Given the above problem setting, the paper studies the following four problems:

Problem 1 (Human Distraction Quantification): The first problem aims to quantify the *distraction* of human co-worker $h \in \mathcal{H}$ moving from $p_h \in \mathcal{P}$ to $p'_h \in \mathcal{R}_H(p_h)$, where $\mathcal{R}_H(p_h) \neq \emptyset$ which in turn implies that $p'_h \neq p_h$. To this end, we first specify an MNS around the desired trajectory of a human co-worker $h \in \mathcal{H}$, which is obtained by using an available search method. We then quantify human distraction through real-time processing of in-situ observations of human motion.

Problem 2 (Human Intention Prediction): The second problem is to learn intention of human co-worker $h \in \mathcal{H}$ by quantifying the intention probability $\Pr(p'_h|p_h)$ for $p'_h \in \mathcal{R}_H(p_h)$, where

$$\sum_{p'_h \in \mathcal{R}_H(p_h)} \Pr(p'_h|p_h) = 1, \qquad p_h \in \mathcal{P},\ h \in \mathcal{H}. \tag{12}$$

Problem 3 (UAS Motion Planning): The third problem is to obtain a safe trajectory for a UAS in a workplace shared with the human co-workers. The key assumption of the paper is that UAS flies at a low-altitude close to the ground. This assumption is necessary for indoor UAS operation in some workplaces such as underground mines, warehouse sites, or under-canopy spaces in farms. Therefore, the objective of trajectory planning is that UAS avoids areas where the likelihood of existence of human co-workers are high. We develop a non-stationary MDP to optimize the UAS trajectory.

Problem 4 (UAS Trajectory Planning and Tracking): A finite-time optimal control problem is defined to plan the UAS desired trajectory given every two consecutive desired way points that are assigned by the MDP planner. We use the nonlinear dynamics developed in [32] to model the motion of each UAS and apply a feedback linearization-based trajectory tacking control to ensure that the UAS desired trajectory is stably tracked at any time.

IV. Approach

Problems 1, 2, 3, and 4 are formulated and explained in Sections IV-A, IV-B, IV-C, and IV-D below. We let $p_h(k) \in \mathcal{P}$ and $p'_h(k) \in \mathcal{R}_H(p_h(k))$ be the "origin" WS and "destination"

TABLE I
NOMENCLATURE

$\mathcal{H}$	Finite set identifying human co-workers
$k \in \mathbb{N}$	Discrete time
$t \in \mathbb{R}$	Continuous time
$\mathcal{P}$	The Petri Nets' "place" set defining all workstations
$\mathcal{T}$	The Petri Nets' "transition" set
$\mathcal{E}_H$	Set of unweighted arcs specifying possible displacements of human co-workers between places (workstations)
$\mathcal{E}_U$	Set of unweighted arcs specifying possible displacements of UAS between places (workstations)
$\mathcal{E}'$	Set unweighted arcs specifying displacements that can be performed at the same workstation
$\mathcal{N}_H(p) \subset \mathcal{T}$	The set defining possible transitions that can be fired by human co-workers at $p \in \mathcal{P}$
$\mathcal{N}_U(p) \subset \mathcal{T}$	The set defining possible transitions that can be fired by UAS co-workers at $p \in \mathcal{P}$
$\mathcal{R}_p^H \subset \mathcal{P}$	The set defining all possible next WSs for human co-workers
$\mathcal{R}_p^U \subset \mathcal{P}$	The set defining all possible next WSs for human co-workers
$\mathcal{M}_H$	The Petri Nets' marking function used to specify human co-worker distribution is the work environment
$\mathcal{M}_U$	The Petri Nets' marking function used to specify UAS co-worker distribution is the work environment
$\mathcal{D}$	A set specifying all discrete positions of all nodes used for discretization of motion space
$\bar{\mathcal{D}} \subset \mathcal{D}$	A set specifying all discrete positions of workstations, defined by set $\mathcal{P}$
$\mathcal{O} \subset \mathcal{D}$	A set specifying all obstacle nodes distributed over the motion space
$\bar{\mathbf{r}}_{h,p_h,p'_h} \in \bar{\mathcal{D}}$	"Current" desired position when $h \in \mathcal{H}$ travels from $p_h \in \mathcal{P}$ to $p'_h \in \mathcal{R}_H(p_h)$
$\bar{\mathbf{r}}^+_{h,p_h,p'_h} \in \bar{\mathcal{D}}$	"Next" desired position when $h \in \mathcal{H}$ travels from $p_h \in \mathcal{P}$ to $p'_h \in \mathcal{R}_H(p_h)$
$\bar{\mathbf{v}}_h$	Desired velocity of human co-worker $h \in \mathcal{H}$
$\bar{\mathbf{r}}_U \in \bar{\mathcal{D}}$	"Discrete Actual" position of UAS
$\bar{\mathbf{r}}_h \in \bar{\mathcal{D}}$	"Discrete Actual" position when $h \in \mathcal{H}$ travels from $p_h \in \mathcal{P}$ to $p'_h \in \mathcal{R}_H(p_h)$
$\mathbf{r}(t) \in \mathbb{R}^3$	"Actual" position of UAS at continuous time t
$\mathbf{r}_d(t) \in \mathbb{R}^3$	"Desired" position of UAS at continuous time t
$\Pr\left(p'_h \vert p_h\right)$	Probability that $p'_h \in \mathcal{R}_H(p_h)$ is the "destination" for human $h \in \mathcal{H}$ when s/he departed WS $p_h \in \mathcal{P}$
$\mathbf{r}_d(t) \in \mathbb{R}^3$	"Desired" position of UAS at continuous time t
$\mathcal{S}_{i,g}$	MDP state set specified based on "origin" WS $i \in \mathcal{P}$ and "goal" WS $g \in \mathcal{R}_U(i)$
$\mathcal{A}$	MDP action set
$C_{g,k} : \mathcal{S}_{i,g} \times \mathcal{A} \to \mathbb{R}$	MDP cost function at discrete time k when $i \in \mathcal{P}$ and $g \in \mathcal{R}_U(i)$ are the "origin" and "goal" WSs
γ	MDP discount factor
$\mathcal{F}_{i,g} : \mathcal{S}_{i,g} \times \mathcal{A} \to \mathcal{S}_{i,g}$	Deterministic MDP transition function
$\mathbf{x}$, $\mathbf{u}$, and $\mathbf{y}$	Quadcopter UAS state, input, and output vector, respectively
m	Mass of the quadcopter UAS
$\mathbf{J}_q$	Mass moment of inertia of the quadcopter UAS

WS place of a human co-worker $h \in \mathcal{H}$ at discrete time k. We use the following position notations:

1) $\bar{\mathbf{r}}_{h,p_h,p'_h} \in \bar{\mathcal{D}}$: "Current" desired position of human co-worker $h \in \mathcal{H}$ given $p_h(k) \in \mathcal{P}$ and $p'_h(k) \in \mathcal{R}_H(p_h)$ as the "origin" WS and "destination" WS, respectively, at discrete time k.
2) $\bar{\mathbf{r}}^+_{h,p_h,p'_h} \in \bar{\mathcal{D}}$: "Next" desired position of human co-worker $h \in \mathcal{H}$ given $p_h(k) \in \mathcal{P}$ and $p'_h(k) \in \mathcal{R}_H(p_h(k))$ as the "origin" WS and "destination" WS, respectively, at discrete time k.
3) $\bar{\mathbf{r}}_h \in \bar{\mathcal{D}}$: "Discrete Actual" position of human worker $h \in \mathcal{H}$ at discrete time k.
4) $\bar{\mathbf{r}}_U \in \bar{\mathcal{D}}$: "Discrete Actual" position of UAS at discrete time k.
5) $\mathbf{r}(t) \in \mathbb{R}^3$: "Actual" position of UAS at continuous time t.
6) $\mathbf{r}_d(t) \in \mathbb{R}^3$: "Desired" position of UAS at continuous time t.

Note that the desired trajectory of human worker $h \in \mathcal{H}$ can be assigned by using an A* search over set $\mathcal{D}$. Also, the actual position of every human co-worker $h \in \mathcal{H}$ is observed by the UAS at every discrete time k, where it is either captured by on-site cameras and communicated to the UAS, or it is accurately estimated by the UAS through processing of onboard visual sensory information.

The parameters and variables used for modeling of the proposed human-UAS interaction problem are described in Table I.

A. Problem 1: Distraction Probability Quantification

In this section, we present an approach for updating distraction probability of human co-workers in real-time based on empirical data collected over the past n_p time steps from discrete time k, at every discrete time k.

Definition 6: We define

$$\bar{\mathbf{v}}_h = \bar{\mathbf{r}}^+_{h,p_h,p'_h} - \bar{\mathbf{r}}_{h,p_h,p'_h}. \tag{13}$$

as desired velocity of human worker $h \in \mathcal{H}$ obtained based on "current" desired position $\bar{\mathbf{r}}_{h,p_h,p'_h}$ and "next" desired position $\bar{\mathbf{r}}^+_{h,p_h,p'_h}$ of human worker $h \in \mathcal{H}$.

Given $\bar{\mathbf{v}}_h$, the motion of human worker $h \in \mathcal{H}$ is categorized as follows:

- We say human co-worker $h \in \mathcal{H}$ moves "diagonally," if $\|\bar{\mathbf{v}}_h\| = \sqrt{2}$.
- We say human co-worker $h \in \mathcal{H}$ moves "straight," if $\|\bar{\mathbf{v}}_h\| = 1$.
- A human co-worker $h \in \mathcal{H}$ desires to "Stay", if $\bar{\mathbf{v}}_h = (0, 0)$.

Definition 7: For every human worker $h \in \mathcal{H}$,

$$\mathcal{I}_d\big(\bar{\mathbf{r}}_{h,p_h,p'_h}\big) = \Big\{\bar{\mathbf{r}}_{h,p_h,p'_h} + (i_s, j_s) \in \mathcal{D} : i_s, j_s \in \{-d, \cdots, d\}\Big\}. \tag{14}$$

is the moving neighboring set (MNS) of degree d, where d is called *degree of neighborhood*.

Because d is time-invariant, MNS remains rigid at every discrete time k. Note that MNS $\mathcal{I}_d\big(\bar{\mathbf{r}}_{h,p_h,p'_h}\big)$ is a rectangular box that is centered at $\bar{\mathbf{r}}_{h,p_h,p'_h}$ and contains $(2d+1)\times(2d+1)$ cells.

Definition 8: We use $b_h\big(\bar{\mathbf{r}}_h, \bar{\mathbf{r}}_{h,p_h,p'_h}, \|\bar{\mathbf{v}}_h\|\big)$ to denote the number of visits of $\bar{\mathbf{r}}_h \in \mathcal{I}_d\big(\bar{\mathbf{r}}_{h,p_h,p'_h}\big)$, when human co-worker $h \in \mathcal{H}$ moves from origin (WS) $p_h(k) \in \mathcal{P}$ towards destination $p'_h(k) \in \mathcal{R}^H_{p_h}$ with desired velocity $\bar{\mathbf{v}}_h$.

The distraction probability of human worker $h \in \mathcal{H}$ is denoted by α_h and defined by

$$\alpha_h\big(\bar{\mathbf{r}}_h(k)|\bar{\mathbf{r}}_{h,p_h,p'_h}, \|\bar{\mathbf{v}}_h\|\big) = \frac{b_h\big(\bar{\mathbf{r}}_h(k), \bar{\mathbf{r}}_{h,p_h,p'_h}, \|\bar{\mathbf{v}}_h\|\big)}{\sum_{\bar{\mathbf{r}}_h(k)\in\mathcal{I}_d\left(\bar{\mathbf{r}}_{h,p_h,p'_h}\right)} b_h\big(\bar{\mathbf{r}}_h, \bar{\mathbf{r}}_{h,p_h,p'_h}, \|\bar{\mathbf{v}}_h\|\big)}, \tag{15}$$

for every human worker $h \in \mathcal{H}$. Note that the desired trajectory of every human co-worker $h \in \mathcal{H}$ can be obtained offline between every "origin" WS $p_h \in \mathcal{P}$ and "goal" WS $p'_h \in \mathcal{R}_H(p_h)$ by using A* search without imposing computation cost for real time perception. Therefore, the distraction α_h can be computed and updated with minimal computation cost at every discrete time k.

B. Problem 2: Human Intention Prediction

We define

$$\delta\big(\bar{\mathbf{r}}_h(k), \bar{\mathbf{r}}_{h,p_h,p'_h}(k)\big) = \exp\big(-\big\|\bar{\mathbf{r}}_h(k) - \bar{\mathbf{r}}_{h,p_h,p'_h}(k)\big\|\big) \tag{16}$$

to quantify deviation of human co-worker $h \in \mathcal{H}$ from the desired trajectory $\bar{\mathbf{r}}_{h,p_h,p'_h}$ when the worker moves from origin $p_h \in \mathcal{P}$ towards destination $p'_h \in \mathcal{R}_H(p_h)$. Note that $\delta\big(\bar{\mathbf{r}}_h, \bar{\mathbf{r}}_{h,p_h,p'_h}\big)$ can be considered as a reward function decreases from 1 to 0 as $\big\|\mathbf{r} - \bar{\mathbf{r}}_{h,p_h,p'_h}(p, p')\big\|$ increases from 0 to $+\infty$.

By knowing $p_h \in \mathcal{P}$, $p'_h \in \mathcal{R}_H(p_h)$, and trajectory of human worker $h \in \mathcal{H}$, over the past N_p time steps,

$$\Pr\big(p'_h|p_h\big) = \frac{\sum_{\tau=k-N_p}^{k-1} \delta\big(\bar{\mathbf{r}}_h(\tau), \bar{\mathbf{r}}_{h,p_h,p'_h}(\tau)\big)}{\sum_{p'_h\in\mathcal{R}^H_p}\sum_{\tau=k-N_p}^{k-1} \delta\big(\bar{\mathbf{r}}_h(\tau), \bar{\mathbf{r}}_{h,p_h,p'_h}(\tau)\big)} \tag{17}$$

assigns the probability that $p'_h \in \mathcal{R}_H(p_h)$ is the goal WS for human worker $h \in \mathcal{H}$ at discrete time k.

C. Problem 3: UAS Motion Planning

We consider a scenario at which UAS is in transitioning mode from WS $i \in \mathcal{P}$ to goal WS $g \in \mathcal{R}_U(i)$ while interacting with multiple human co-workers. The UAS motion planning is considered a non-stationary MDP with time-invariant states, actions, transitions, and discount factor but time-varying cost. To be more precise, the MDP cost is updated at every discrete time k such that human intention and distraction are consistently learned and incorporated into UAS motion planning.

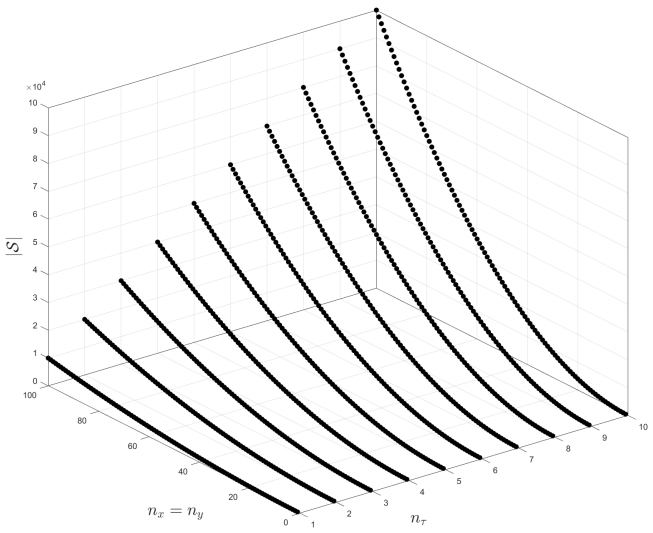


Fig. 4. $|\mathcal{S}|$ versus n_x ($n_x = |\mathcal{X}|$) and n_τ ($n_\tau = |\mathcal{K}|$), where $n_x = n_y$ ($n_y = |\mathcal{Y}|$).

1) MDP Components: Assuming the UAS aims to displace from $i \in \mathcal{P}$, at $\bar{\mathbf{r}}_i \in \bar{\mathcal{D}}$, to $g \in \mathcal{R}_U(i)$, at $\bar{\mathbf{r}}_g \in \bar{\mathcal{D}}$, the proposed (non-stationary) MDP is given by tuple $\big(\mathcal{S}_{i,g}, \mathcal{A}, \mathcal{C}_{g,k}, \mathcal{F}_{i,g}, \gamma\big)$ at discrete time k with state set $\mathcal{S}_{i,g}$, action set $\mathcal{A}$, transition function $\mathcal{F}_{i,g} : \mathcal{S}_{i,g} \times \mathcal{A} \to \mathcal{S}_{i,g}$, cost $\mathcal{C}_{g,k} : \mathcal{S}_{i,g} \times \mathbb{N} \to \mathbb{R}$, and discount factor $\gamma \in [0, 1]$. These components are defined as follows:

State Set $\mathcal{S}_{i,g}$: We use set $\mathcal{K} = \{1, \cdots, n_\tau\}$ to define a future horizon interval of length n_τ. Then, MDP state set $\mathcal{S}_{i,g}$ is a subset of

$$\mathcal{S} = \big\{\mathbf{s} = \big(\bar{\mathbf{r}}, \tau\big) : \bar{\mathbf{r}} \in \mathcal{D} \setminus \mathcal{O},\ \tau \in \mathcal{K}\big\}. \tag{18}$$

for every $i \in \mathcal{P}$ and $g \in \mathcal{R}_U(i)$. If set $\mathcal{S}$ defines the state set, regardless of the locations of the "origin" and "goal" WS positions, the motion planning becomes computationally expensive, and real-time motion planning may not be feasible. For better clarification, Fig. 4 shows $|\mathcal{S}|$ for different n_x ($n_x = |\mathcal{X}|$) and n_τ ($n_\tau = |\mathcal{K}|$), where $n_x = n_y$ ($n_y = |\mathcal{Y}|$).

To address this issue, we define a variable-size state set $\mathcal{S}_{i,g}$, for every $i \in \mathcal{P}$ and $g \in \mathcal{R}_U(i)$, such that the the following conditions hold:

1) $\big(\bar{\mathbf{r}}_i, \tau\big) \in \mathcal{S}_{i,g}$; and $\big(\bar{\mathbf{r}}_g, \tau\big) \in \mathcal{S}_{i,g}$ for every $\tau \in \mathcal{K}$, $i \in \mathcal{P}$, and $g \in \mathcal{R}_U(i)$.
2) $\bar{\mathbf{r}}_g \in \mathcal{D}$ can be reached from $\bar{\mathbf{r}}_i \in \mathcal{D}$ for every $i \in \mathcal{P}$ and $g \in \mathcal{R}_U(i)$.

We note that $\mathcal{S}_{i,g} \subset \mathcal{S}$ can be obtained, for every $i \in \mathcal{P}$ and $g \in \mathcal{R}_U(i)$ offline, without imposing computation cost on real-time UAS motion planning, while conditions 1 and 2 are both satisfied.

Action Set $\mathcal{A}$: We use set

$$\mathcal{A} = \{\text{E, NE, N, NW, W, SW, S, SE, O}\} \tag{19}$$

to define nine possible actions including go towards "east" (E), go towards "northeast" (NE), go towards "north" (N), go toward "northwest" (NW), go towards "west" (W), go towards "southwest" (SW), go towards "south" (S), go towards "southeast" (SE), and "no motion" (O).

Cost $\mathcal{C}_{g,k}$: By expressing $\mathbf{s}$ as $\mathbf{s} = (\bar{\mathbf{r}}, \tau)$, for every $\bar{\mathbf{r}} \in \mathcal{D}$ and every $\tau \in \mathcal{K}$, cost is imposed on operation of the UAS intending to go to the "goal" WS $g \in \mathcal{R}_U(i)$ from any state state $\mathbf{s} \in \mathcal{S}_{i,g}$. To define the MDP cost, we define spatiotemporal heat map

$$\mathcal{J}_{g,k}(\mathbf{s}) = \bar{\mathcal{J}}_g(\bar{\mathbf{r}}) + \underline{\mathcal{J}}_k(\bar{\mathbf{r}}, \tau), \tag{20}$$

where

$$\bar{\mathcal{J}}_g(\mathbf{r}) = \|\bar{\mathbf{r}} - \bar{\mathbf{r}}_g\|, \qquad g \in \mathcal{P},\ \bar{\mathbf{r}} \in \mathcal{D},\ \bar{\mathbf{r}}_g \in \bar{\mathcal{D}}, \tag{21}$$

is the distance from $\bar{\mathbf{r}} \in \mathcal{D}$ and $\bar{\mathbf{r}}_g \in \bar{\mathcal{D}}$ and

$$\underline{\mathcal{J}}_k(\bar{\mathbf{r}}, \tau) = c_0 \sum_{h \in \mathcal{H}} \sum_{p_h \in \mathcal{P}} \sum_{p'_h \in \mathcal{R}_H(p_h)} \Pr\big(p'_h(k+\tau) | p_h(k+\tau)\big) \times \alpha_h\big(\bar{\mathbf{r}} | \bar{\mathbf{r}}_{h,p_h,p'_h}(k+\tau), \bar{\mathbf{v}}_h(k+\tau)\big) \tag{22}$$

where $c_0 > 0$, $\tau \in \mathcal{K}$, and $\bar{\mathbf{r}} \in \mathcal{D}$. The MDP cost is then defined by

$$\mathcal{C}_{g,k}(\bar{\mathbf{r}}, \tau) = \begin{cases} c_g \ll 0 & \bar{\mathbf{r}} = \bar{\mathbf{r}}_g \in \bar{\mathcal{D}},\ g \in \mathcal{P},\ \forall k,\ \forall \tau \\ c_{obs} \gg 0 & \bar{\mathbf{r}} \in \mathcal{O},\ \forall k,\ \forall \tau \\ \mathcal{J}_{g,k}(\mathbf{s}) & \bar{\mathbf{r}} \neq \bar{\mathbf{r}}_g,\ \bar{\mathbf{r}} \notin \mathcal{O},\ \forall k,\ \forall \tau, \end{cases} \tag{23}$$

where $|c_g|$ and $|c_{obs}|$ are sufficiently large and constant.

Transition Probability: By assuming that UAS actuators and sensors are all healthy, we can suppose deterministic transitions over the state space. Under this assumption, $\mathcal{F}(\mathbf{s}^+|\mathbf{s}, a) \in \{0, 1\}$ is a binary variable, for every current state $\mathbf{s} \in \mathcal{S}$, next state $\mathbf{s}^+ \in \mathcal{S}$, and action $a \in \mathcal{A}$.

We use Algorithm 1 to specify deterministic transitions of the UAS over the MDP state space. To be more precise, Algorithm 1 uses $\mathbf{s} = (j, h, \tau)$ and $\mathbf{s}^+ = (j', h', \tau')$ to denote the "current" and "next" states, respectively. By default, Algorithm 1 assumes that $i' = i$, $j' = j$, and $\tau' = \tau + 1$. To update i', j', and τ', it follows the following rules, if needed:

1) If $\tau+1 \notin \mathcal{K}$, the "next" state is the same as the "current" state ($\mathbf{s}^+ = \mathbf{s}$).
2) Under action $a \in \{\text{SE, E, NE}\}$, $j' = j + 1$ is replaced, if $(j', h', \tau') \in \mathcal{S}_{i,g}$.
3) Under action $a \in \{\text{NE, N, NW}\}$, $h' = h + 1$ is replaced, if $(j', h', \tau') \in \mathcal{S}_{i,g}$.
4) Under action $a \in \{\text{SW, W, NW}\}$, $j' = j - 1$ is replaced, if $(j', h', \tau') \in \mathcal{S}_{i,g}$
5) Under action $a \in \{\text{SE, S, SW}\}$, $h' = h - 1$ is replaced, if $(j', h', \tau') \in \mathcal{S}_{i,g}$.

Algorithm 1 Obtaining Function $\mathcal{F}_{i,g}(\mathbf{s}^+|\mathbf{s}, a)$ Under Deterministic Transition Assumption Over the $\mathcal{S}_{i,g}$ ($i \in \mathcal{P}$ and $g \in \mathcal{R}_U(i)$)

Get: State set $\mathcal{S}_{i,g}$ and action $a \in \mathcal{A}$ for $i \in \mathcal{P}$ and $g \in \mathcal{R}_U(i)$.
Return: Binary function $\mathcal{F}_{i,g}(\mathbf{s}^+|\mathbf{s}, a)$ for every $\mathbf{s} \in \mathcal{S}_{i,g}$.
if $\tau + 1 \in \mathcal{K}$ **then**
  $\tau' \leftarrow \tau + 1$
  for every $(j, h, \tau) \in \mathcal{S}_{i,g}$ **do**
    $\mathbf{s} \leftarrow (j, h, \tau)$
    $j' \leftarrow j$ and $h' \leftarrow h$.
    if $a \in \{\text{SE, E, NE}\}$ **then**
      if $(j + 1, h, \tau + 1) \in \mathcal{S}_{i,g}$ **then**
        $j' \leftarrow j + 1$.
      end if
    end if
    if $a \in \{\text{NE, N, NW}\}$ **then**
      if $(j, h + 1, \tau + 1) \in \mathcal{S}_{i,g}$ **then**
        $h' \leftarrow h + 1$.
      end if
    end if
    if $a \in \{\text{NW, W, SW}\}$ **then**
      if $(j - 1, h, \tau + 1) \in \mathcal{S}_{i,g}$ **then**
        $j' \leftarrow j - 1$.
      end if
    end if
    if $a \in \{\text{SE, S, SW}\}$ **then**
      if $(j, h - 1, \tau + 1) \in \mathcal{S}_{i,g}$ **then**
        $h' \leftarrow h - 1$.
      end if
    end if
  end for
else
  $j' \leftarrow j$, $h' \leftarrow h$, and $\tau' \leftarrow \tau$.
end if
$\mathbf{s}^+ \leftarrow (j', h', \tau')$.
$\mathcal{F}(\mathbf{s}^+|\mathbf{s}, a) \leftarrow 1$.

2) Solution: The optimal UAS trajectory is obtained by solving the Bellman equation which is given by

$$V_{i,g,k}(\mathbf{s}) = \min_{a \in \mathcal{A}} \sum_{\mathbf{s}^+ \in \mathcal{S}_{i,g}} \mathcal{F}_{i,g}(\mathbf{s}^+|\mathbf{s}, a)(\mathcal{C}_{g,k}(\mathbf{s}) + V_{i,g,k}(\mathbf{s}^+)), \qquad \forall \mathbf{s} \in \mathcal{S}_{i,g} \tag{24}$$

where $g \in \mathcal{P}$ is the goal WS for the UAS and $k \in \mathbb{N}$ denotes discrete time. The optimal policy is assigned by

$$\pi^*_{i,g,k}(\bar{\mathbf{r}}, \tau) = \min_{a \in \mathcal{A}} \sum_{\mathbf{s}^+ \in \mathcal{S}_{i,g}} \mathcal{F}_{i,g}(\mathbf{s}^+|\mathbf{s}, a)(\mathcal{C}_{g,k}(\mathbf{s}) + V_{i,g,k}(\mathbf{s}^+)). \tag{25}$$

over for state $(\bar{\mathbf{r}}, \tau) \in \mathcal{S}_{i,g}$, $i \in \mathcal{P}$, $g \in \mathcal{R}_U(i)$.

While Eq. (25) species the optimal policy over the entire space set $\mathcal{S}_{i,g}$, UAS only applies optimal actions over a subset of $\mathcal{S}_{i,g}$ that is associated with $\tau = 1 \in \mathcal{T}$. Therefore, the optimal action of the UAS is denoted by $a^*(k)$, at discrete

time k, and obtained by

$$a^*(k) = \pi^*_{i,g,k}(\bar{\mathbf{r}}_U(k), 1), \qquad i \in \mathcal{P},\ g \in \mathcal{R}_U(i), \tag{26}$$

where $\bar{\mathbf{r}}_U(k)$ is the UAS actual discrete position of the UAS at discrete time k. UAS implements Algorithm 2 to assign optimal action $a^*(k)$ at every discrete time k.

Algorithm 2 UAS Motion Planning

Input: Set $\mathcal{D}$, set $\mathcal{K}$, actions set $\mathcal{A}$, transition function $\mathcal{F}$, discount factor γ, initial position $\bar{\mathbf{r}}_i$ ($i \in \mathcal{P}$ and $\bar{\mathbf{r}}_i \in \bar{\mathcal{D}}$), and UAS target position $\bar{\mathbf{r}}_g$ ($g \in \mathcal{R}_U(i) \subset \mathcal{P}$ and $\bar{\mathbf{r}}_g \in \bar{\mathcal{D}}$).
Set: $k \leftarrow 0$; $\bar{\mathbf{r}}_U \leftarrow \bar{\mathbf{r}}_i$.
for $<$ $h \in \mathcal{H}$ **do**
  Get: Get $p_h(0) \in \mathcal{P}$.
  for $<$ $p'_h(0) \in \mathcal{R}_H(p_h(0))$ **do**
    Determine desired trajectory $\bar{\mathbf{r}}_{h,p_h,p'_h}$.
  end for
end for
while $\bar{\mathbf{r}}_U(k) \neq \bar{\mathbf{r}}_g$ **do**
  for $<$ $h \in \mathcal{H}$ $>$ **do**
    for $<$ $p'_h(k) \in \mathcal{R}_H(p_h(k))$ $>$ **do**
      if $p_h(k+1) \neq p_h(k)$ or $p'_h(k+1) \neq p'_h(k)$ **then**
        Update desired trajectory $\bar{\mathbf{r}}_{h,p_h,p'_h}$.
      end if
      Update distraction probability by Eq. (15).
      Update $\Pr(p'_h|p_h)$ by Eq. (17).
      Update MDP cost function by Eq. (23).
    end for
  end for
  Obtain optimal policy by solving Eq. (25).
  Obtain optimal action $a^*(k) = \pi^*_{i,g,k}(\bar{\mathbf{r}}_U(k), 1)$.
  Obtain the next UAS position $\bar{\mathbf{r}}'$.
  $k \leftarrow k+1$.
  $\bar{\mathbf{r}}_U(k) \leftarrow \bar{\mathbf{r}}'$.
end while

D. Problem 4: UAS Trajectory Planning and Tracking

To model multicopter UAS motion, we use $x(t)$, $y(t)$, and $z(t)$ to define UAS position components at (continuous) time t. We also use $\phi(t)$, $\theta(t)$, and $\psi(t)$ to define roll, pitch, and yaw angles of the UAS at (continuous) time t, where we use the 3-2-1 Euler angle standard to characterize orientation of the UAS. The thrust force magnitude generated by the quacopter UAS is denoted by $f(t)$ at (continuous) time.

We model motion of a multicopter UAS with mass m by nonlinear dynamics

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \\ \mathbf{y} = \begin{bmatrix} x & y & z & \psi \end{bmatrix}^T \end{cases} \tag{27}$$

where $\mathbf{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} & f & \dot{f} \end{bmatrix}^T \in \mathbb{R}^{14\times 1}$ is the state vector, $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T$ is the control input, and

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \frac{f}{m}(\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta) \\ \frac{f}{m}(\cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi) \\ \frac{f}{m}\cos\phi\cos\theta - g \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ 0 \\ 0 \\ 0 \\ \dot{f} \\ 0 \end{bmatrix}, \tag{28}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \mathbf{g}_3 & \mathbf{g}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{9\times 1} & \mathbf{0}_{9\times 3} \\ \mathbf{0}_{3\times 1} & \mathbf{I}_3 \\ 0 & \mathbf{0}_{1\times 3} \\ 1 & \mathbf{0}_{1\times 3} \end{bmatrix} \tag{29}$$

are smooth functions. In Eq. (28), $g = 9.81 m/s^2$ is gravity acceleration. Note that $u_1 = \ddot{f}$, $u_2 = \ddot{\phi}$, $u_3 = \ddot{\theta}$, $u_4 = \ddot{\psi}$ are considered as the components of the control input vector $\mathbf{u}$.

Because we use the 3−2−1 Euler angle standard to quantify the multicopter orientation, $\Lambda = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ is related to the multicopter angular velocity vector Ω by

$$\Omega = \Gamma\Lambda, \tag{30}$$

where Ω is expressed with respect to the multicopter body frame and

$$\Gamma = \begin{bmatrix} 1 & 0 & \sin\theta \\ 0 & \cos\phi & \sin\theta\sin\phi \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}. \tag{31}$$

Control input vector $\mathbf{u} = \begin{bmatrix} \ddot{\phi} & \ddot{\theta} & \ddot{\psi} \end{bmatrix}^T$ can be related to the control torque $\mathbf{M}_q$ by applying the rotational dynamics of the multicopter, which can be expressed by

$$\mathbf{J}_q\Gamma\mathbf{u} + \mathbf{J}_q\dot{\Gamma}\Lambda + \Omega \times (\mathbf{J}_q\Omega) = \mathbf{M}_q \tag{32}$$

where $\mathbf{J}_q$ is the inertia matrix of the multicopter [32].

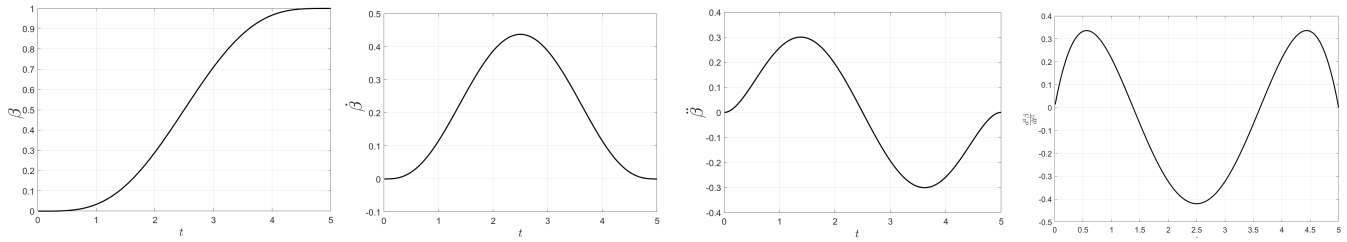
1) UAS Trajectory Planning: Given every two consecutive optimal waypoints $\bar{\mathbf{r}}_U(k)$ and $\bar{\mathbf{r}}_U(k+1)$, assigned by the MDP planner, the UAS desired trajectory is denoted by $\mathbf{r}_d(t) = \begin{bmatrix} x_d(t) & y_d(t) & z_d(t) \end{bmatrix}^T$ and defined by

$$\begin{bmatrix} x_d(t) \\ y_d(t) \end{bmatrix} = (1 - \beta(t - t_k))\bar{\mathbf{r}}_U(k) + \beta(t - t_k)\bar{\mathbf{r}}_U(k+1), \tag{33a}$$

$$z_d(t) = z_0, \tag{33b}$$

for $t \in [t_k, t_{k+1})$ and $k \in \mathbb{N}$, where $\Delta t = t_{k+1} - t_k$ is constant for every $k \in \mathbb{N}$. Without loss of generality, we desire that the multicopter flies at constant altitude, therefore, z_0 is time-invariant. By taking time derivative from $\mathbf{r}_d(t)$, we obtain

$$\frac{d^l\mathbf{r}_d(t)}{dt^l} = \frac{d^l\beta(t)}{dt^l}\left(\begin{bmatrix} \bar{\mathbf{r}}_U(k+1) \\ 0 \end{bmatrix} - \begin{bmatrix} \bar{\mathbf{r}}_U(k) \\ 0 \end{bmatrix}\right), \tag{34}$$



(a) (b) (c) (d)

Fig. 5. Optimal β, $\dot{\beta}$, $\ddot{\beta}$, and $\dddot{\beta}$ versus $t \in [0, \Delta t]$, where we choose $\Delta t = 5s$.

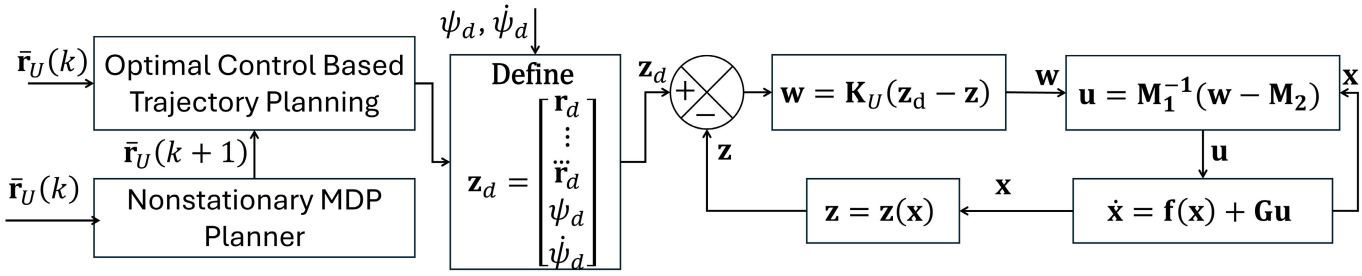


Fig. 6. Black diagram of UAS trajectory planning and tracking.

at any time $t \in [t_k, t_{k+1})$, for every $k \in \mathbb{N}$ and $l = 1, 2, 3, 4$. We let $\beta(t)$ be updated by the following fourth order dynamics:

$$\frac{d^4\beta(t)}{dt^4} = s_\beta(t), \qquad t \in [t_k, t_{k+1}), \ k \in \mathbb{N}, \tag{35}$$

where $s_\beta(t)$ is the control that is assigned by minimizing

$$\mathrm{J} = \frac{1}{2}\int_{t_k}^{t_k+\Delta t} s_\beta^2 dt, \qquad k \in \mathbb{N}, \tag{36}$$

subject to the following boundary conditions:

$$\beta(t_k) = 0, \qquad k \in \mathbb{N}, \tag{37a}$$
$$\beta(t_{k+1}) = 1, \qquad k \in \mathbb{N}, \tag{37b}$$
$$\frac{d^l\beta(t_k)}{dt^h} = \frac{d^l\beta(t_{k+1})}{dt^l} = 0, \ k \in \mathbb{N}, l = 1, 2, 3. \tag{37c}$$

The above boundary conditions ensure that the multicopter UAS fully stops at every waypoint $\bar{\mathbf{r}}_U(k)$ which is indeed required since the UAS may need to change its motion direction for 90 degrees to reach $\mathbf{r}_U(k+1)$ from $\mathbf{r}_U(k)$. We solve the above optimal control problem by using the approach detailed in the Appendix and obtain optimal β, $\dot{\beta}$, $\ddot{\beta}$, and $\dddot{\beta}$ plotted in Fig. 5.

2) Trajectory Tracking: We apply the feedback linearization method developed in Refs. [32] and [33] to design the multicopter trajectoy tracking control $\mathbf{u}$. For this purpose, we define state transformation

$$\mathbf{z}(\mathbf{x}) = \begin{bmatrix} \mathbf{r}^T & \dot{\mathbf{r}}^T & \ddot{\mathbf{r}}^T & \dddot{\mathbf{r}}^T & \psi & \dot{\psi} \end{bmatrix}^T \tag{38}$$

and update $\mathbf{z}$ by the following dynamics

$$\dot{\mathbf{z}} = \mathbf{A}_U\mathbf{z} + \mathbf{B}_U\mathbf{w} \tag{39}$$

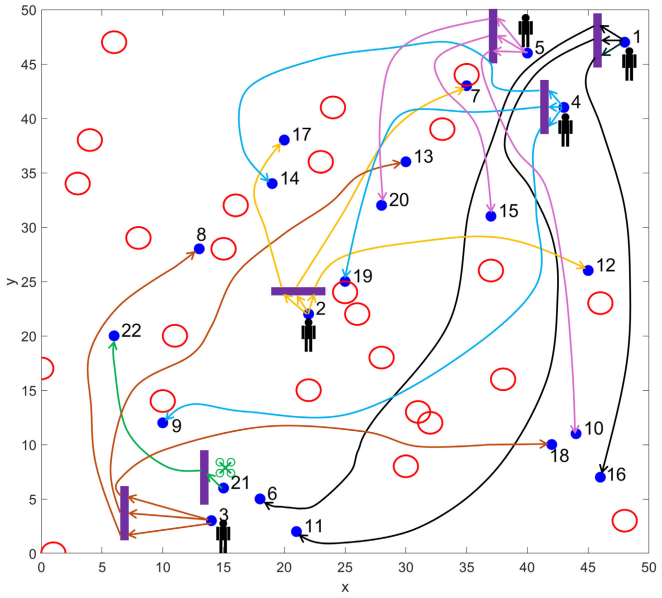
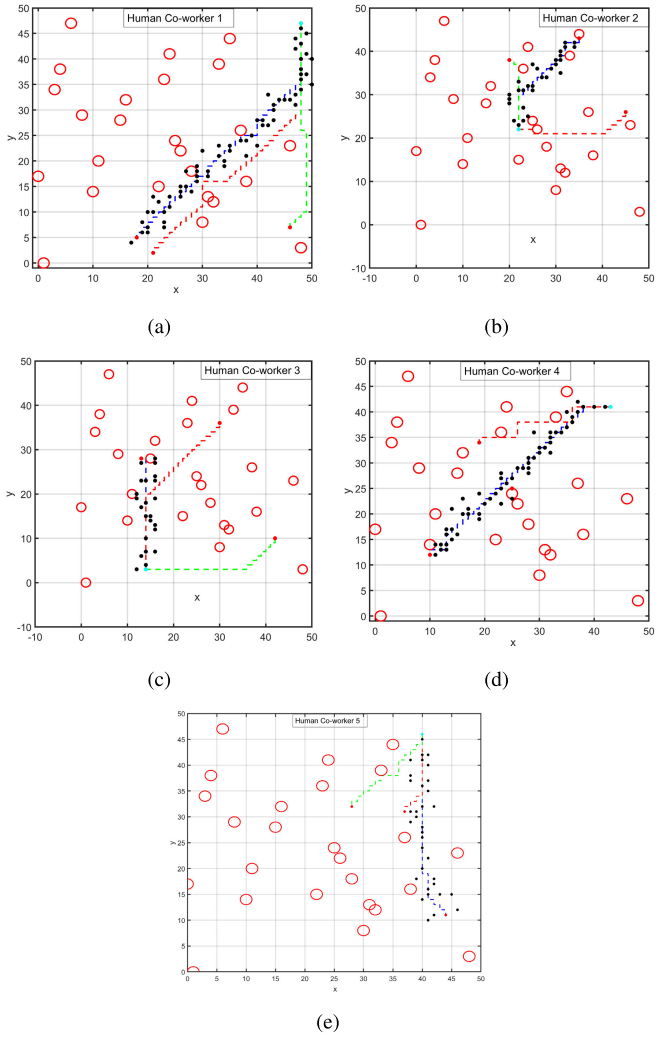


Fig. 7. Schematic of human-UAS interaction in the workplace.

where $\mathbf{w} = \begin{bmatrix} \dddot{x} & \dddot{y} & \dddot{z} & \ddot{\psi} \end{bmatrix}^T$ is the control input of dynamics (39), and

$$\mathbf{A}_U = \begin{bmatrix} \mathbf{0}_{9\times3} & \mathbf{I}_9 & \mathbf{0}_{9\times1} & \mathbf{0}_{9\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times9} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times9} & 0 & 1 \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times9} & 0 & 0 \end{bmatrix} \text{ and}$$

$$\mathbf{B}_U = \begin{bmatrix} \mathbf{0}_{9\times3} & \mathbf{0}_{9\times1} \\ \mathbf{I}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 0 \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}.$$



(a) (b) (c) (d) (e)

Fig. 8. Desired trajectories between the origin WS and three possible next WSs are shown by dashed plot for human co-workers 1, 2, 3, 4, and 5 in sub-figures (a), (b), (c), (d), and (e), respectively. Also, actual trajectory of each co-worker is shown by black dots in sub-figures (a)-(e).

Note that $\mathbf{w}$ is related to the control input of multicopter UAS, denoted by $\mathbf{u}$, by

$$\mathbf{w} = \mathbf{M}_1\mathbf{u} + \mathbf{M}_2, \tag{40}$$

where

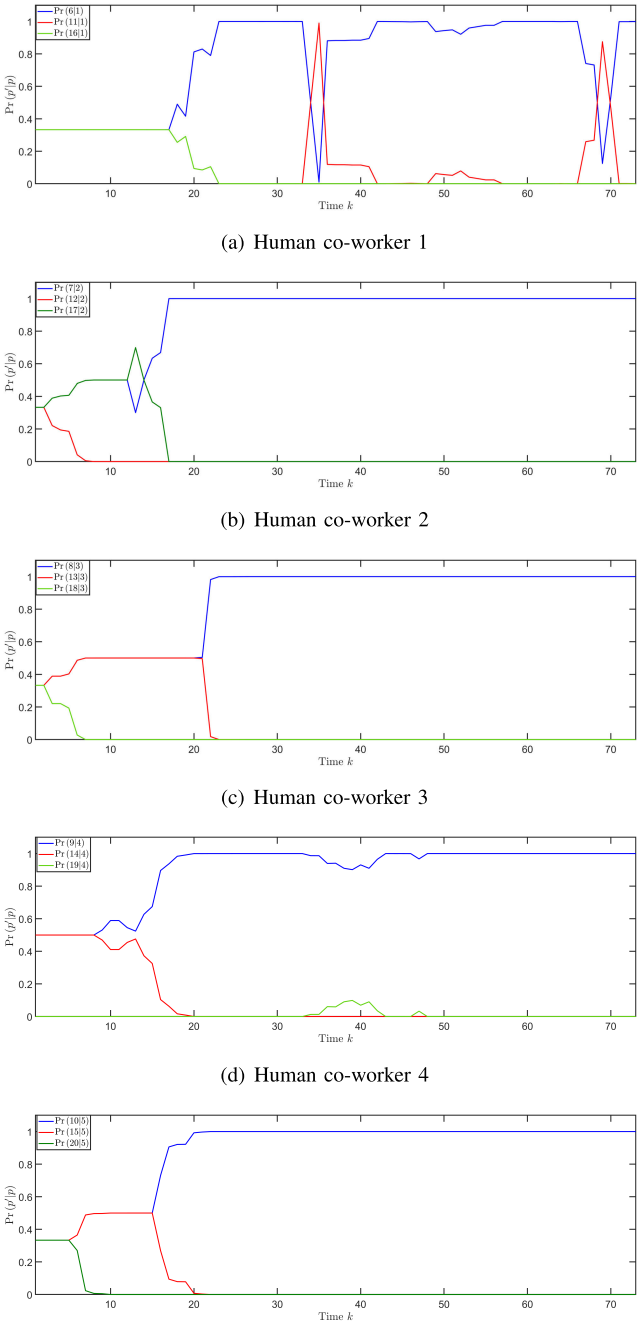
$$\mathbf{M}_1 = \begin{bmatrix} L_{\mathbf{g}_1}L_{\mathbf{f}}^3 x & L_{\mathbf{g}_2}L_{\mathbf{f}}^3 x & L_{\mathbf{g}_3}L_{\mathbf{f}}^3 x & L_{\mathbf{g}_4}L_{\mathbf{f}}^3 x \\ L_{\mathbf{g}_1}L_{\mathbf{f}}^3 y & L_{\mathbf{g}_2}L_{\mathbf{f}}^3 y & L_{\mathbf{g}_3}L_{\mathbf{f}}^3 y & L_{\mathbf{g}_4}L_{\mathbf{f}}^3 y \\ L_{\mathbf{g}_1}L_{\mathbf{f}}^3 z & L_{\mathbf{g}_2}L_{\mathbf{f}}^3 z & L_{\mathbf{g}_3}L_{\mathbf{f}}^3 z & L_{\mathbf{g}_4}L_{\mathbf{f}}^3 z \\ L_{\mathbf{g}_1}L_{\mathbf{f}}\psi & L_{\mathbf{g}_2}L_{\mathbf{f}}\psi & L_{\mathbf{g}_3}L_{\mathbf{f}}\psi & L_{\mathbf{g}_4}L_{\mathbf{f}}\psi \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{41a}$$

$$\mathbf{M}_2 = \begin{bmatrix} L_{\mathbf{f}}^4 x & L_{\mathbf{f}}^4 y & L_{\mathbf{f}}^4 z & L_{\mathbf{f}}^2\psi \end{bmatrix}^T \in \mathbb{R}^{4\times 1}. \tag{41b}$$

The control design objective is objective is achieved by choosing $\mathbf{w}$ such that $\mathbf{z}$ stably tracks

$$\mathbf{z}_d = \begin{bmatrix} \mathbf{r}_d^T & \dot{\mathbf{r}}_d^T & \ddot{\mathbf{r}}_d^T & \dddot{\mathbf{r}}_d^T & \psi_d & \dot{\psi}_d \end{bmatrix}^T,$$

where $\mathbf{r}_d$ is the UAS desired trajectory, defined by Eq. (33), and ψ_d is the desired yaw angle of multicopter $i \in \mathcal{V}$. Without loss of generality, this paper assumes that $\psi_d(t) = 0$ at any



(a) Human co-worker 1

(b) Human co-worker 2

(c) Human co-worker 3

(d) Human co-worker 4

(e) Human co-worker 5

Fig. 9. Intention prediction of human co-workers.

time t, which in turn implies that $\dot{\psi}_d(t) = 0$. To achieve the control objective, we choose

$$\mathbf{w} = \mathbf{K}_U(\mathbf{z}_d - \mathbf{z}) \tag{42}$$

such that $\mathbf{A}_U - \mathbf{B}_U\mathbf{K}_U$ is Hurwitz. Then, the control input of the UAS is obtained by

$$\mathbf{u} = \mathbf{M}_1^{-1}(\mathbf{w} - \mathbf{M}_2). \tag{43}$$

Figure 6 shows the block diagram of the multicopter control design based on the above feedback linearation method.

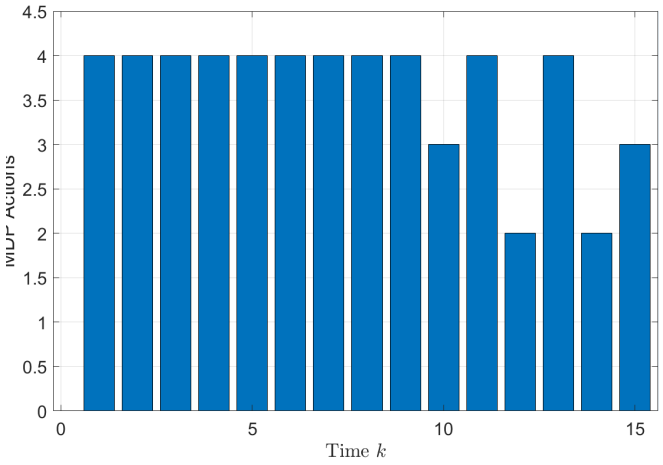


Fig. 10. Optimal actions taken by the UAS to safely plan its trajectory.

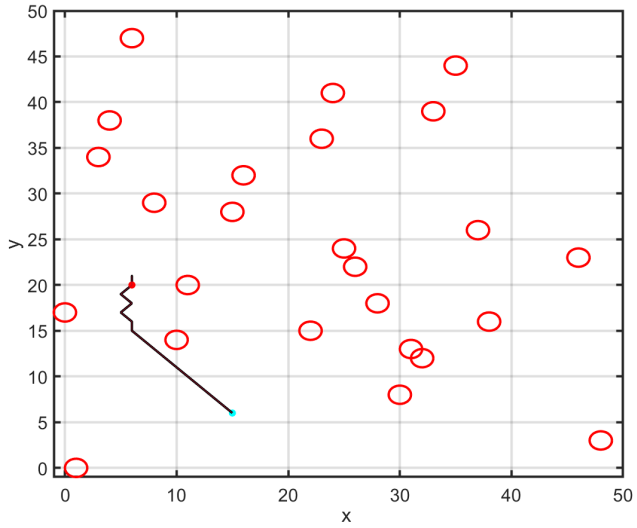


Fig. 11. The optimal path of the UAS in the workplace.

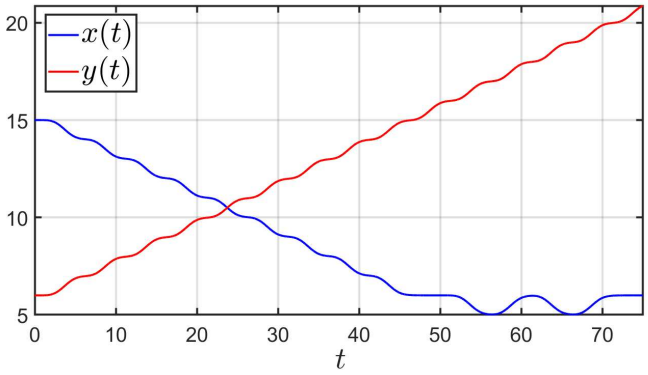


Fig. 12. x and y components of actual trajectory of the multicopter UAS from the initial position $(6, 15)$ to final destination $(21, 6)$.

V. Simulation Results

We consider a single UAS and five human co-workers in a shared environment (motion space). The motion space is a rectangle and represented by a uniform grid of 50×50 size. The motion space consists a finite number of obstacles that are shown by red circles in Fig. 7. The workplace has 33 WSs that are identified by set $\mathcal{P} = \{1, \cdots, 33\}$ and represented by the places shown in Fig. 3. The schematic of the motion space (workplace) is shown in Fig. 7 where WSs 1 through 22 are marked by blue as they represent either the current WSs or the possible next WSs for the humans and UAS existing in the workplace. We consider a UAS-human interaction scenario that is formally specified by the Petri Nets shown in Fig. 3 where UAS aims to move from $21 \in \mathcal{P}$ to $22 \in \mathcal{R}_U(21)$. This UAS motion interacts with the motion of five human co-workers at $1, 2, 3, 4, 5 \in \mathcal{P}$ in the shared workplace. Motion of human co-workers deals with uncertainty as each has three possible destinations. More specifically,

$$\begin{aligned}
\mathcal{R}_H(1) &= \{6, 11, 16\}, \qquad 1 \in \mathcal{P}, \\
\mathcal{R}_H(2) &= \{7, 12, 17\}, \qquad 2 \in \mathcal{P}, \\
\mathcal{R}_H(3) &= \{8, 13, 18\}, \qquad 3 \in \mathcal{P}, \\
\mathcal{R}_H(4) &= \{9, 14, 19\}, \qquad 4 \in \mathcal{P}, \\
\mathcal{R}_H(5) &= \{10, 15, 20\}, \qquad 5 \in \mathcal{P},
\end{aligned}$$

define the possible next WSs for the human co-workers. The schematic of the proposed human-UAS interaction is shown in Fig. 7.

Given origin and possible WSs for every human co-worker, the desired trajectory are obtained by using A* search method and shown by dashed plots in Fig. 8. Also, the actual trajectory of each human co-worker is shown by black dots in Figs. 8(a), (b), (c), (d), and (e), for human co-workers 1, 2, 3, 4, and 5, respectively. By using the human intention prediction approach, presented in Section IV-B, We obtain $\Pr(p'|p)$ for human-co-workers 1 through 5 and plot them in Figs. 9(a) through 9(e). Optimal actions of the UAS are plotted versus discrete time k in Fig. 10, where 1, 2, 3, 4, 5, 6, 7, 8, and 9 represent "E," "NE," "N," "NW," "W," "SW," "S," "SE," and "O," respectively. The optimal path of the UAS in the workplace is shown in Fig. 11.

For UAS trajectory planning, we use $\Delta t = 5s$, therefore, β, $\dot{\beta}$, $\ddot{\beta}$, and $\dddot{\beta}$ plotted in Fig 5 are used to obtain $\mathbf{r}(t)$ $\dot{\mathbf{r}}(t)$, $\ddot{\mathbf{r}}(t)$, and $\dddot{r}(t)$ for $t \in [t_k, t_k]$ and $k \in \mathbb{N}$. Figure 12 plots UAS actual position components x and y versus time t.

VI. Conclusion and Future Work

We develop a novel method for abstraction of UAS and human interaction in the same workplace. We consider a workplace that consists of a finite number of WSs and apply Petri Nets to: (i) abstractly represent WSs and transition between WSs, and (ii) leverage incomplete knowledge about human intentions. We develop a variable-size non-stationary MDP, with time-invariant transition function, state space, actions, and discount factor and time-varying cost, to safely plan the UAS trajectory in the presence of human co-workers. In particular, the MDP cost function is updated based on real-time observation data so that human intention and distraction of human co-workers are properly incorporated in UAS motion planing when UAS closely intercat with human co-workers.

The proposed variable-size non-stationary MDP can efficiently define the state space based on the locations of the origin WS $i \in \mathcal{P}$ and goal WS $g \in \mathcal{R}_U(p)$ to achieve real-time motion planning at a low computing cost. The computation cost of motion planning can be further reduced by efficient structuring of the Petri Nets so that distance between every "origin" WS $i \in \mathcal{P}$ and every goal WS "$g \in \mathcal{R}_U(i)$" not very large.

The other method for reducing the computation cost is to replace the MDP by a time-dependent A* search that can be used for motion planning over spatiotemporal motion spaces. Indeed, A* is much less computationally expensive compared with the MDP and can further reduce the computation cost. However, for the proposed huamn-UAS collaboration application, it needs to incorporate the uncertainty associated with human intention. This will be considered as a plan for the future work.

In the near future, we will also be studying high-level motion planning for UAS under inaccurate sensing and actuation assumption. For this study, we can use model-based and model-free methods to incorporate probabilistic transitions into MDP. Particularly, we can use Gaussian processes to incorporate model-based uncertainty associated with actuation inaccuracy into MDP transition probability function. We can also apply different reinforcement learning approaches to include and learn model-free uncertainty. When sensor reading is uncertain, states are not fully observable. Under this situation, we can apply the hidden Markov models to model evolution over the state space and use Partially Observable Markov Decision Processes (POMDP) method to optimally plan UAS actions.

APPENDIX

This section provides a solution for the optimal control problem defined in Section IV-D1 to assign optimal β over time interval $[t_k, t_{k+1}]$, for every $k \in \mathbb{N}$.

We can rewrite dynamics (35) using the state space form as follows:

$$\dot{\mathbf{x}}_\beta = \mathbf{A}_\beta \mathbf{x}_\beta + \mathbf{B}_\beta s_\beta, \qquad k \in \mathbb{N},\ t \in [t_k, t_{k+1}], \tag{A1}$$

where $\mathbf{x}_\beta = [\beta\ \dot{\beta}\ \ddot{\beta}\ \dddot{\beta}]^T$, $\mathbf{B}_\beta = [\mathbf{0}_{1\times 3}\ 1]^T$, and

$$\mathbf{A}_\beta = \begin{bmatrix} \mathbf{0}_{3\times 1} & \mathbf{I}_3 \\ 0 & \mathbf{0}_{1\times 3} \end{bmatrix}.$$

By rewriting the boundary conditions given by Eq. (37), dynamics (A1) must satisfy the following end-point conditions:

$$\mathbf{x}_\beta(t_k) = \mathbf{0}_{4\times 1}, \qquad k \in \mathbb{N}, \tag{A2}$$

$$\mathbf{x}_\beta(t_{k+1}) = [1\ 0\ 0\ 0]^T, \qquad k \in \mathbb{N}. \tag{A3}$$

Let $\lambda = [\lambda_\beta\ \lambda_{\dot{\beta}}\ \lambda_{\ddot{\beta}}\ \lambda_{\dddot{\beta}}]^T$ be the co-state vector, then, the Hamiltonian is obtained by

$$\mathrm{H}(\mathbf{x}_\beta, \lambda, s_\beta) = \frac{1}{2} s_\beta^2 + \lambda^T (\mathbf{A}_\beta \mathbf{x}_\beta + \mathbf{B}_\beta s_\beta). \tag{A4}$$

By applying the Prontryagin's minimum principle, the optimal control s_β^* is obtained by

$$s_\beta^* = -\mathbf{B}_\beta^T \lambda \tag{A5}$$

where $\mathbf{x}_\beta$ and λ are updated by dynamics

$$\begin{bmatrix} \dot{\mathbf{x}}_\beta \\ \dot{\lambda} \end{bmatrix} = \mathbf{A}_{sys} \begin{bmatrix} \mathbf{x}_\beta \\ \lambda \end{bmatrix} \tag{A6}$$

with

$$\mathbf{A}_{sys} = \begin{bmatrix} \mathbf{A}_\beta & -\mathbf{B}_\beta^T \mathbf{B}_\beta \\ \mathbf{0}_{4\times 4} & -\mathbf{A}_\beta^T \end{bmatrix}. \tag{A7}$$

Therefore,

$$\begin{bmatrix} \mathbf{x}_\beta(t) \\ \lambda(t) \end{bmatrix} = \Phi_{sys}(t, t_k) \begin{bmatrix} \mathbf{x}_\beta(t_k) \\ \lambda(t_k) \end{bmatrix}, \qquad t \in [t_k, t_{k+1}], \tag{A8}$$

where

$$\Phi_{sys}(t, t_k) = \begin{bmatrix} \Phi_{11}(t, t_k) & \Phi_{12}(t, t_k) \\ \Phi_{21}(t, t_k) & \Phi_{22}(t, t_k) \end{bmatrix} = \mathrm{e}^{\mathbf{A}_{sys}(t - t_k)} \tag{A9}$$

is the state transition matrix. By imposing boundary condition (A3), we obtain

$$\lambda(t_k) = \Phi_{12}^{-1}(t, t_k)\big(\mathbf{x}_\beta(t_{k+1}) - \Phi_{11}(t, t_k)\mathbf{x}_\beta(t_k)\big) \tag{A10}$$

By knowing $\mathbf{x}_\beta(t_k)$ from boundary condition (A2) and $\lambda(t_k)$, obtained by (A10), $\mathbf{x}_\beta(t)$ is obtained by (A8) at any time $t \in [t_k, t_{k+1}]$.

REFERENCES

[1] S. Ahirwar, R. Swarnkar, S. Bhukya, and G. Namwade, "Application of drone in agriculture," *Int. J. Current Microbiology Appl. Sci.*, vol. 8, no. 1, pp. 2500–2505, Jan. 2019.

[2] U. R. Mogili and B. B. V. L. Deepak, "Review on application of drone systems in precision agriculture," *Proc. Comput. Sci.*, vol. 133, pp. 502–509, Jan. 2018.

[3] L. Wawrla, O. Maghazei, and T. Netland, "Applications of drones in warehouse operations," in *Proc. Whitepaper. ETH Zurich, D-MTEC*, 2019, p. 212.

[4] S. S. Ali, S. Khan, N. Fatma, C. Ozel, and A. Hussain, "Utilisation of drones in achieving various applications in smart warehouse management," *Benchmarking, Int. J.*, vol. 31, no. 3, pp. 920–954, Apr. 2024.

[5] E. Jones, J. Sofonia, C. Canales Cardenas, S. Hrabar, and F. Kendoul, "Advances and applications for automated drones in underground mining operations," in *Proc. Int. Conf. Deep High Stress Mining*, 2019, pp. 323–334.

[6] E. Jones, J. Sofonia, C. Canales, S. Hrabar, and F. Kendoul, "Applications for the hovermap autonomous drone system in underground mining operations," *J. Southern Afr. Inst. Mining Metall.*, vol. 120, no. 1, pp. 49–56, 2020.

[7] N. Dilshad, J. Hwang, J. Song, and N. Sung, "Applications and challenges in video surveillance via drone: A brief survey," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2020, pp. 728–732.

[8] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Comput. Commun.*, vol. 156, pp. 1–10, Apr. 2020.

[9] T. Umar, "Applications of drones for safety inspection in the Gulf cooperation council construction," *Eng., Construct. Architectural Manage.*, vol. 28, no. 9, pp. 2337–2360, Nov. 2021.

[10] M. Namian, M. Khalid, G. Wang, and Y. Turkan, "Revealing safety risks of unmanned aerial vehicles in construction," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2675, no. 11, pp. 334–347, Nov. 2021.

[11] M. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, vol. 204. Cham, Switzerland: Springer, 2012.

[12] M. Zhou and K. Venkatesh, *Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1999.

[13] A. Marcello Mangini and M. Roccotelli, "Innovative services for electric mobility based on virtual sensors and Petri nets," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 9, pp. 1845–1859, Sep. 2023.

[14] R. Yang, Z. Ding, C. Jiang, and M. Zhou, "Modeling and analysis of three properties of mobile interactive systems based on variable Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 4, pp. 1–13, Aug. 2023.

[15] J. Liang, Y. Tang, R. Hare, B. Wu, and F.-Y. Wang, "A learning-embedded attributed Petri net to optimize student learning in a serious game," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 3, pp. 1–9, Oct. 2022.

[16] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Optimal scheduling of human–robot collaborative assembly operations with time Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 1, pp. 70–84, Jan. 2021.

[17] A. Casalino, E. Mazzocca, M. Grazia Di Giorgio, A. M. Zanchettin, and P. Rocco, "Task scheduling for human–robot collaboration with uncertain duration of tasks: A fuzzy approach," in *Proc. 7th Int. Conf. Control, Mechatronics Autom. (ICCMA)*, Nov. 2019, pp. 90–97.

[18] S. Alirezazadeh and L. A. Alexandre, "Dynamic task scheduling for human–robot collaboration," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8699–8704, Oct. 2022.

[19] R. E. Yagoda and M. D. Coovert, "How to work and play with robots: An approach to modeling human–robot interaction," *Comput. Hum. Behav.*, vol. 28, no. 1, pp. 60–68, Jan. 2012.

[20] S. Hjorth and D. Chrysostomou, "Human–robot collaboration in industrial environments: A literature review on non-destructive disassembly," *Robot. Comput.-Integr. Manuf.*, vol. 73, Feb. 2022, Art. no. 102208.

[21] N. Ali, S. Punnekkat, and A. Rauf, "Modeling and safety analysis for collaborative safety-critical systems using hierarchical colored Petri nets," *J. Syst. Softw.*, vol. 210, Apr. 2024, Art. no. 111958.

[22] N. Höllerich and D. Henrich, "Coloured Petri nets for monitoring human actions in flexible human–robot teams," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 2749–2756.

[23] J. Irizarry, M. Gheisari, and B. N. Walker, "Usability assessment of drone technology as safety inspection tools," *J. Inf. Technol. Construct. (ITcon)*, vol. 17, no. 12, pp. 194–212, 2012.

[24] J. R. Cauchard, K. Y. Zhai, M. Spadafora, and J. A. Landay, "Emotion encoding in human-drone interaction," in *Proc. 11th ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, 2016, pp. 263–270.

[25] H. D. Doran, M. Reif, M. Oehler, C. Stöhr, and P. Capone, "Conceptual design of human-drone communication in collaborative environments," in *Proc. 50th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2020, pp. 118–121.

[26] S. Feuerriegel, R. Geraldes, A. Gonçalves, Z. Liu, and H. Prendinger, "Interface design for human-machine collaborations in drone management," *IEEE Access*, vol. 9, pp. 107462–107475, 2021.

[27] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 81–87.

[28] J. V. Deshmukh, A. Donzé, S. Ghosh, X. Jin, G. Juniwal, and S. A. Seshia, "Robust online monitoring of signal temporal logic," *Form. Methods Syst. Des.*, vol. 51, pp. 5–30, Jul. 2017.

[29] G. Silano, A. Afifi, M. Saska, and A. Franchi, "A signal temporal logic planner for ergonomic human–robot collaboration," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2023, pp. 328–335.

[30] H. Rastgoftar, "Integration of A* search and classic optimal control for safe planning of continuum deformation of a multiquadcopter system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4119–4134, Oct. 2022.

[31] B. Huang, M. Zhou, A. Abusorrah, and K. Sedraoui, "Scheduling robotic cellular manufacturing systems with timed petri net, A* search, and admissible heuristic function," *IEEE Trans. Automat. Sci. Eng.*, vol. 19, no. 1, pp. 243–250, Jan. 2022.

[32] A. E. Asslouj and H. Rastgoftar, "Quadcopter tracking using euler-angle-free flatness-based control," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2023, pp. 1–6.

[33] H. Rastgoftar and I. V. Kolmanovsky, "Safe affine transformation-based guidance of a large-scale multi-quadcopter system (MQS)," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 2, pp. 640–653, May 2021.

Hossein Rastgoftar received the B.Sc. degree in mechanical engineering-thermo-fluids from Shiraz University, Shiraz, Iran, the dual M.S. degrees in mechanical systems and solid mechanics from Shiraz University and the University of Central Florida, Orlando, FL, USA, and the Ph.D. degree in mechanical engineering from Drexel University, Philadelphia, in 2015. He is an Assistant Professor with The University of Arizona. Prior to this, he was an Adjunct Assistant Professor with the University of Michigan, Ann Arbor, from 2020 to 2021. He was also an Assistant Research Scientist (2017–2020) and a Post-Doctoral Researcher (2015–2017) with the Aerospace Engineering Department, University of Michigan. His current research interests include dynamics and control, multiagent systems, cyber-physical systems, and optimization and Markov decision processes.