

Received 29 July 2023; revised 10 November 2023; accepted 29 November 2023. Date of publication 6 December 2023; date of current version 27 December 2023. Recommended by Senior Editor Dr. Yue Wang.

Digital Object Identifier 10.1109/OJCSYS.2023.3340034

# A Computational Framework for Optimal Adaptive Function Allocation in a Human-Autonomy Teaming Scenario

SOOYUNG BYEON (Member, IEEE), JOONWON CHOI (Graduate Student Member, IEEE),
AND INSEOK HWANG (Member, IEEE)

(Modeling, Control, and Learning Approaches for Human-Robot Interaction Systems)

School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907 USA

CORRESPONDING AUTHOR: SOOYUNG BYEON (e-mail: sbyeon@purdue.edu).

This work was supported by the NSF under Grant CNS-1836952.

**ABSTRACT** This article proposes a quantitative framework for optimally allocating task functions in human-autonomy teaming (HAT). HAT involves cooperation between humans and autonomous agents to achieve common goals. As humans and autonomous agents possess different capabilities, function allocation plays a crucial role in ensuring effective HAT. However, designing the best adaptive function allocation remains a challenge, as existing methods often rely on qualitative rules and intensive human-subject studies. To address this limitation, we propose a computational function allocation approach that leverages cognitive engineering, computational work models, and optimization techniques. The proposed optimal adaptive function allocation method is composed of three main elements: 1) analyze the teamwork to identify a set of all possible function allocations within a team construction, 2) numerically simulate the teamwork in temporal semantics to explore the interaction of the team with complex environments using the identified function allocations in a trial-and-error manner, and 3) optimize the adaptive function allocation with respect to a given situation such as physical conditions, available information resources, and human mental workload. For the optimization, we utilize performance metrics such as task performance, human mental workload, and coherency in function allocations. To illustrate the effectiveness of the proposed framework, we present a simulated HAT scenario involving a human work model and drone fleet for last-mile delivery in disaster relief operations.

**INDEX TERMS** Computational work model, function allocation, human-automation interaction, human-autonomy teaming, human-vehicle systems.

#### I. INTRODUCTION

Human-Autonomy teaming (HAT) is a collaborative working strategy involving at least one human and one autonomous agent [1]. An autonomous agent is a system capable of responding to the environment with a degree of self-governance [2]. Recent studies on HAT have demonstrated that autonomous agents can enhance task performance when working as teammates with humans rather than mere tools. However, introducing autonomous agents does not imply that humans can simply delegate team tasks to them in complex environments [3]. Cognitive scientists emphasize the need to coordinate and distribute teamwork among the

different teammates by considering their interactions and interdependencies, as the introduction of autonomous agents creates a new cognitive system [4], [5]. An effective HAT design should consider the unique capabilities, limitations, and interdependencies of each teammate to foster symbiotic teamwork, especially in handling off-nominal conditions. For example, humans can contribute with their high-level cognitive control in unseen situations, while autonomous agents can excel within operational boundary conditions [6].

Function allocation coordinates task functions within a human-autonomy team. Well known traditional methods are Men-are-better-at/Machines-are-better-at



(MABA-MABA) [7], Levels of Automation (LOA) [8], and adaptive function allocation [9]. MABA-MABA laid a solid foundation for HAT function allocation but it has been criticized for assuming fixed human and autonomous agent capabilities [10]. LOA uses a continuum of automation levels in four function classes (information acquisition, information analysis, decision selection, and action implementation). It offers valuable insights and design flexibility. However, LOA faces challenges due to its broad classifications [5]. Adaptive function allocation methods have been actively explored to specify who does what and when in HAT [9]. These methods enable human-autonomy teams to assign functions based on relevant variables such as task performance and human mental workload. However, one major concern is the need for computational models and simulation methods for effective and efficient quantitative analysis [11].

#### A. RELATED WORK

## 1) COMPUTATIONAL MODELING

State-of-the-art findings in computational modeling techniques and cognitive engineering have been integrated to address function allocation issues. Computational work models (CWMs) focus on examining multi-agent concepts of operation by analyzing work and team constructions [12]. The CWM is a simulation engine that can propagate the state of the system operated by a HAT design while accounting for constraints and interdependencies of physical and information resources among heterogeneous agents [13]. CWM frameworks can incorporate cognitive work analysis (CWA) to identify work requirements systematically [14] and agent-based simulation [15] to demonstrate function allocations in temporal semantics [5]. This approach helps avoid the substitution myth (the wrong assumption that an autonomous agent can seamlessly replace a human in HAT [16]) in function allocation through clarifying the teamwork dynamics instead of focusing solely on who does what. The analysis provides quantified measures of teamwork and interactions, even in complex scenarios like air traffic control under off-nominal conditions [12], [17]. However, the current work is limited to fixed function allocations in HAT and lacks an effective approach to analyze adaptive function allocations using CWMs. Although the computational models in [12], [17] have laid a robust groundwork for function allocation studies, their models are not formulated as a mathematical representation conducive to standard optimization techniques.

# 2) CYBER-PHYSICAL-HUMAN SYSTEMS

Cyber-physical-human systems (CPHSs) encompass systems that involve cyber, physical, and human layers, interacting in time and space [18], [19]. Many applications in CPHS are closely related HAT such as human-in-the-loop control systems and human-robot interaction. Regarding the function

(or task) allocation problem, several CPHS studies have modeled human behaviors and cognitive states to assist humans. For example, shared control approaches between pilots and autopilots have been explored to enhance safety by allocating functions based on the timeline of off-nominal conditions [20]. In other shared control approaches, cyber-physical systems can assist humans by partially or completely taking on control functions based on observed human behaviors and inferred cognitive states [21], [22]. Physical humanrobot interaction (pHRI) techniques ensure safe and efficient working spaces in manufacturing by providing necessary assistive functions based on inferred human intention [23]. However, optimal function allocation are not the primary focus in these applications. In human-robot collaboration studies, various optimization frameworks aim to maximize production performance in manufacturing [24], [25], [26]. Nevertheless, most of the existing work concentrates on interactions in the same physical location. The function allocation in terms of information flow for remotely operated systems may require further investigation for enhancing human perception [18] (p.259).

#### **B. CONTRIBUTIONS**

In this article, we propose a quantitative framework for optimal adaptive function allocation in the HAT context to address limitations in existing methods. Our main contributions can be summarized in two parts. First, we introduce a systematic approach for formulating the CWM as a discrete-time stochastic control process. The formulated work model offers a concise representation of complex team interactions to facilitate the simulation of HAT designs. We employ a parameterized human work model in place of real human-subjects to efficiently assess a range of feasible function allocations in a computationally tractable manner and eliminate the need for exhaustive human-subject studies during the early HAT design phase. Second, we present a computational optimization framework for determining an optimal function allocation policy that adapts to various team situations or states. This framework can accommodate diverse performance metrics, including mission completion time, human mental workload, and the coherency of function allocations [17]. It is versatile enough to consider a wide array of simulated situations, such as initiating teamwork, gathering information, and accounting for off-nominal conditions. Furthermore, it offers flexibility in updating computational models to enable the incorporation of modified scenarios, human work models, and other factors for re-optimizing function allocation in new situations or under new models.

## C. APPROACH

The proposed framework comprises three technical elements:
1) identifying teamwork and all allowed function allocations in terms of interaction, constraints, and interdependency using CWA, 2) simulating a wide range of randomized scenarios to demonstrate and evaluate identified teamwork and function

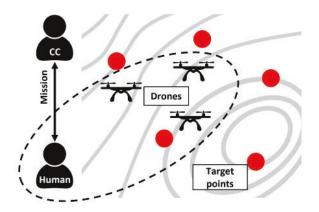


FIGURE 1. An illustrative HAT scenario on disaster relief package delivery with drones (CC: command center).

allocations using CWM, and 3) finding the optimal adaptive function allocation (i.e., situation-dependent function allocation) using an optimization technique.

The remainder of this article is organized as follows. In Section II, an illustrative HAT scenario is presented to contextualize the proposed framework. Section III presents technical elements in detail. Simulation results and discussion are given in Sections IV and V, respectively. Section VI concludes the article.

#### II. HUMAN-AUTONOMY TEAMING SCENARIO

We consider a drone fleet scenario for last-mile delivery in disaster scenes [27]. This scenario has two important features for realistic simulations. First, the system and environment can be simulated under off-nominal conditions that include drone faults (e.g., sensor or actuator faults) and environmental anomalies (e.g., obstacles). These complexities are vital for testing a variety of function allocations. Second, the existing studies in interaction between humans and drones allow us to incorporate empirical findings into constructing a realistic CWM [28].

As shown in Fig. 1, we present a HAT scenario involving a command center, one human operator, and three drones. The team's mission is to deliver disaster relief packages, such as medicine, to designated target points safely and timely. The command center serves as the source of mission information such as target points and environmental conditions, which must be communicated to the team. The human operator is responsible for collaborating with the drones to execute the mission under both nominal and off-nominal conditions. Remotely operating via an interface with the team, the human operator also needs to report the current mission status to the command center. The drones are the only physical entities capable of performing the mission within the environment. The key functions for the drones are guidance, navigation, and control (GNC), which can be autonomously conducted by the drones or managed manually by the human operator.

In complex and uncertain environments, maintaining and sharing a high level of situation awareness (SA) is crucial for the team to handle unexpected problems [29]. Let  $SA_c$  represent the critical SA space required for the team's safe and efficient operation [30]. Its elements are given as:

$$SA_{c} = \{X, F, E, M\} \tag{1}$$

where X denotes the physical state space of the drones, including position, velocity, and attitude. F represents the fault information space, encompassing sensor and/or actuator faults. E denotes the environmental anomalies space, such as unexpected wind gusts or obstacles encountered. M represents the current mission information space, such as the position of target points, which can be updated at any time by the command center.

The team must continually perform GNC functions to achieve its common goal. Navigation provides the physical state of the drones (X). To move the drones to target points, guidance and control must be executed based on the mission information (M). The team is also responsible for monitoring, detecting, and resolving faults (F) and environmental anomalies (E). Therefore, GNC and SA-related actions are considered *generalized functions* for the team. Further information will be provided in Section III.

The detailed setups for the scenario are as follows. The mission space is limited to a 200 m  $\times$  200 m  $\times$  30 m space. The maximum speed of the drones is 5 m/s under nominal conditions and limited to 1 m/s when there is an unresolved fault. The simulation time extends up to 300 seconds with a discrete step of  $\Delta t = 0.1$  seconds. The mission completion time is recorded once all packages are delivered. Each drone is assumed to have a maximum of four visiting points (e.g., due to battery capacity and payload weight limitations).

Throughout this article, the term "human" refers to a simulated human work model. To concentrate on the computational function allocation problem, we make two assumptions. First, autonomous agents suggest a function allocation to the human based on the current situation, and the human accepts it with full trust. Although automation misuse, disuse, and abuse are critical problems in human factors [31], we intentionally omit this part to present an illustrative example with acceptable complexity. Second, we assume that the human can perform the given task without making mistakes. This assumption has been widely accepted in work model simulations [12], [13].

#### III. PROPOSED APPROACH

Fig. 2 presents the architecture of the proposed optimal function allocation framework in HAT. All necessary technical details are followed.

# A. COGNITIVE WORK ANALYSIS

We employ the CWA to analyze the work prior to simulation. The CWA is a formative approach used to analyze sociotechnical systems in complex environments [14]. It is especially suitable for identifying the full range of constraints and the work domain rather than determining who is in charge of each sub-task [5]. The formative nature of the CWA allows us to



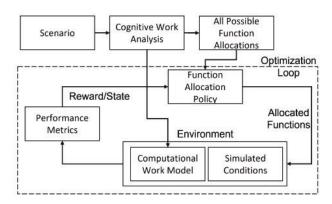


FIGURE 2. The proposed framework for optimal adaptive function allocation in HAT.

focus on describing the system and all possible function allocations in the HAT context. Consequently, the CWA provides valuable insights into the emergent behaviors of sociotechnical systems rather than the simple decomposition of tasks into sub-tasks. We utilize the CWA to analyze the requirements for the HAT design. A well-organized summary of the CWA can be found in [32] (p. 16).

We use a modified work domain analysis (WDA) to model the work in the human-autonomy team [13]. The WDA is the first phase of the CWA, describing the work domain, including tasks, functions, goals, constraints, and the context in which the work is performed. In Fig. 3, the abstraction hierarchy (AH) breaks down the work domain into levels of abstraction for conducting WDA. This structure establishes means-ends links within a systematic hierarchy. Each node represents what it does, an upper-level node answers the question of why it is required, and a lower-level node shows how it can be done. The AH captures the system's flexibility by illustrating that goals can be achieved through multiple approaches within the constraints. The levels are named functional purpose, abstract function, generalized function, physical function, and resource. We also present which entity can undertake each generalized and physical function. Each generalized function and physical function are presented using colors to show the entity responsible for undertaking it in Fig. 3.

We utilize the second phase of the CWA, control task analysis (ConTA), to describe recurring actions in the work domain. The decision-ladder in Fig. 4 represents a sequence of information processes that iterate in the work domain for every time step. There are two different nodes in the process: a rectangular node represents data-processing activities and a oval node denotes the states of knowledge. Navigation is always required to perform the task. SA-related information needs to be observed, identified, interpreted, and resolved if there are any changes in the drones' state (e.g., faults), environmental anomalies, or the mission. These processes produce all essential SA elements in (1). Note that there are shortcuts in the decision-ladder. For instance, if there are no changes in

SA and guidance is already completed, the team can move on to control directly after the SA observation.

#### **B. COMPUTATIONAL WORK MODEL**

In the proposed framework, the CWM aims to simulate the dynamic teamwork based on the identified work domain. The CWM provides further insight into the teamwork in terms of how a specific function allocation induces interactions, dependencies, and constraints that influence the teamwork. The CWM enables our framework to propagate the state of the system over time so that all possible function allocations can be evaluated in terms of the performance metrics. Further details of the CWM can be found in [12], [33].

Let  $s_k$  be the state of the human, drones, and environment at time step  $k \in \mathbb{Z}_{>0}$ :

$$s_k = h(x_k, f_k, e_k, m_k, w_k) \in \mathbb{R}^n, \tag{2}$$

where  $x_k \in X$  denotes the physical states and  $f_k \in F$  denotes the fault of the drones, respectively.  $e_k \in E$  represents the environmental anomalies and  $m_k \in M$  is the mission information.  $w_k \in \mathbb{Z}_{\geq 0}$  denotes the quantified (human mental) workload. h is the user-defined mapping function from CWM variables to the state  $s_k$ . For the HAT scenario in Section II, we can define the state  $s_k$  as the concatenated vector of all elements of  $x_k$ ,  $f_k$ ,  $e_k$ ,  $m_k$ , and  $w_k$ . The CWM can propagate the state with respect to the current state and function allocation  $a_k$ :

$$s_{k+1} = g(s_k, a_k), \tag{3}$$

where g is the state propagation function that is modeled as a discrete-time stochastic control process in the CWM. In the target scenario, the function allocation  $a_k \in \{0, 1\}^m$  is a binary vector of dimension m. Its elements are zero or one, indicating whether the corresponding function is allocated to the human or the drones, respectively. We assume that the assigned function cannot be reallocated to another agent until it is completed by the corresponding agent, in order to prevent unnecessary complexity and inefficiency, such as reallocating the function at every single time step. Table 1 provides a detailed representation of the state  $s_k$ . For example, fault information for a drone can be represented as a 4-dimensional vector, including binary variables on fault occurrence (0 or 1), fault detection (0 or 1), fault isolation (0 or 1), and fault recovery (0 or 1).

To propagate the state, the CWM consists of the agents with physical functions that interact with a given environment. The agents exert their actions on the environment to *get* resources (e.g., read information values) or *set* resources (e.g., drone position changes). The environment includes the dynamics of drones in the three-dimensional space, drone faults, any anomalies such as obstacles, and initial and updated mission information from the command center. Thus, the CWM can simulate any changes in physical or information resources given by any work. The CWM can incorporate dependencies and constraints. For example, in the target scenario, the drones should obtain their physical state through the *get drone state* 

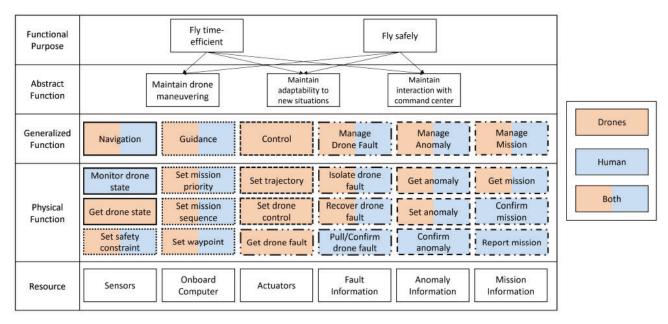


FIGURE 3. The abstraction hierarchy (AH) of the HAT scenario. Each physical function is assigned to a generalized function. Different outlines are used to represent the assignment.

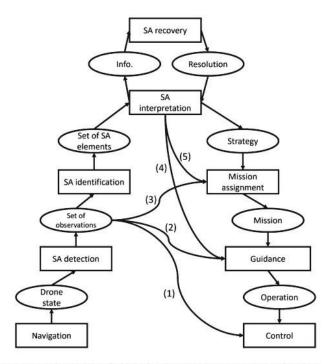


FIGURE 4. The decision-ladder. The shortcuts represent (1) no situation awareness (SA) needs to be addressed except for the navigation, and no need to update guidance; (2) no SA needs to be addressed, but guidance needs updating; (3) no SA needs to be addressed, but the mission needs to be assigned among the drones; (4) an existing SA issue is resolved, but no need to update guidance; and (5) an existing SA issue is resolved, and guidance needs to be updated, respectively.

function before they control their position using the *set drone* control function.

Additional teamwork actions are necessary when there is an authority-responsibility mismatch [13], [17]. When a specific

**TABLE 1.** The state  $s_k \in \mathbb{R}^{93}$  of the human-drones CWM.

State	Value	Dimension
Initialization	0 or 1	1
Human: remaining action time	Integer	16
Human: delayed action	Integer	1
Drone 1: generalized function	0 or 1	6
Drone 1: SA $(F, E, M)$	0 or 1	3
Drone 2: generalized function	0 or 1	6
Drone 2: SA $(F, E, M)$	0 or 1	3
Drone 3: generalized function	0 or 1	6
Drone 3: SA $(F, E, M)$	0 or 1	3
Drone 1: distance to target	Integer	4
Drone 2: distance to target	Integer	4
Drone 3: distance to target	Integer	4
Drone 1: fault information	0 or 1	4
Drone 2: fault information	0 or 1	4
Drone 3: fault information	0 or 1	4
Drone 1: Anomaly information	0 or 1	7
Drone 2: Anomaly information	0 or 1	7
Drone 3: Anomaly information	0 or 1	7
Drones: Anomaly proximity	0 or 1	1
Termination	0 or 1	1
Sum	8*.	93

function is allocated to the drones but the human is responsible for that, the human needs to monitor or confirm the function. A total of six teamwork actions are considered in the CWM. *Monitor* and *Control* teamwork actions are required simultaneously with the work. *Command* and *Confirm* actions need to be conducted before and after the work. Information *Pull* and *Push* are known to play an important role in teamwork when teammates want to proactively share information before it is actually necessary [2], [34]. For instance, when the drones are taking the mission information directly from



the command center, the human is requested to confirm the taken mission information since the human is responsible for obtaining mission information in the HAT design.

Workload is known to be one of the key considerations in function allocation [35]. In the existing work model literature, the workload has been modeled in simple forms. For instance, the workload was modeled as the total number of tasks that the human undertook [12] or the total busy time in teamwork [13]. However, these models cannot reflect situations with low-demanding but time-consuming tasks. Thus, we are inspired by cognitive architecture-based workload models that can distinguish the required time and intensity of a task [36]. We assume that low workload is imposed when a human action is mainly involved with perceptual-model modules (visual, aural, and motor modules). Likewise, a human action that primarily activates central modules (procedural and goal modules) is assumed to cause mid workload. Human actions immersed in memory modules (declarative and imaginal modules) are assumed to invoke high workload. Low, mid, and high workloads correspond to quantified workload 1, 2, and 4, respectively [36]. In the CWM, we assume that there is a maximum allowed workload  $w_{\text{max}} = 6$  for the human. If the current function allocation requires a higher workload for the human than  $w_{\text{max}}$ , overflowed functions would be delayed. The priority of the function follows the decision-ladder in Fig. 4. Note that we have the flexibility to consider different values for  $w_{\text{max}}$  since the proposed framework allows for parameter variation. For example, if we choose a value for  $w_{\text{max}}$ larger than 6, the mission completion time would decrease as the human can handle more functions simultaneously without causing delays.

To account for the distinct capabilities of each agent, we adopt the skill-rule-knowledge (SRK) taxonomy [37]. Cognitive engineering approaches recognize that autonomous agents may require human intervention when faced with unforeseen situations [6], [38]. Humans can efficiently handle uncertainties, such as effectively isolating and recovering drone faults, detecting environmental anomalies in longer ranges, and managing complex communication for mission updates. The drones excel in skill and rule-based tasks, including sensor-based navigation, tracking control, avoiding detected anomalies, and optimizing drone assignments to individual target points using an optimization technique. In this study, we implement obstacle avoidance for the drones using an artificial potential approach [39] and the brute-force optimization for target assignments of drones.

Flowcharts are provided to explicitly illustrate the structure of the CWM. Fig. 5 represents the state propagation function in (3) and the information processes over time. Note that this figure is closely linked to Fig. 4. Fig. 5 illustrates how the team state  $s_k$  evolves based on the occurrence of off-nominal conditions. Each function in Fig. 5 must be allocated to either the human or drones based on the function allocation policy. The flowchart in Fig. 6 demonstrates how the functions allocated to the human can be categorized as either delayed or active functions depending on the situation. As shown in Fig. 7, if

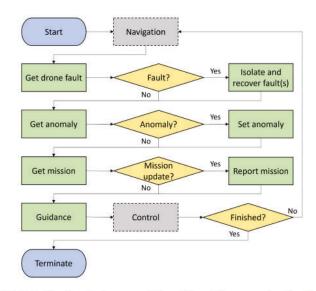


FIGURE 5. The flowchart representation of the state propagation function in (3) over time. The navigation and control functions (dotted boxes) are active at every time step, while the others are conditionally activated based on the current states.

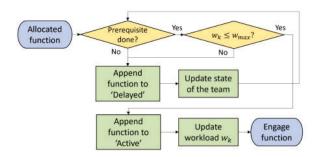


FIGURE 6. The flowchart illustration for the allocated function to the human. The allocated function can be categorized as either a delayed function or an active function based on the satisfaction of prerequisite functions and workload constraint.

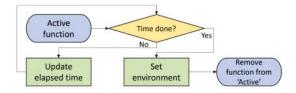


FIGURE 7. The flowchart for handling active functions. Once a function is categorized as active, its time duration is taken into account to determine whether the function is completed. This completion can have an impact on the environment, for example, changing the position of drones based on their dynamics.

an active function is completed (i.e., its elapsed time is equal to or greater than the time duration), it has an impact on the environment. In contrast, delayed functions need to wait until the human can perform the corresponding action (i.e., all prerequisite functions are completed and the workload has room for the delayed function). In this example, the drones

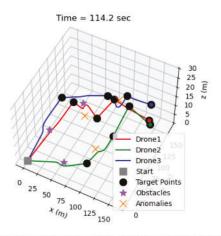


FIGURE 8. The drone trajectories with environment elements. The drones fly to and land on each target point while avoiding obstacles and environmental anomalies.

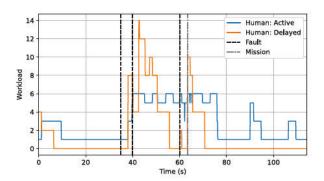


FIGURE 9. The workload pertains to both the active functions and delayed functions allocated to the human. The timing of drone faults and mission updates is represented using the vertical lines.

do not have a workload constraint, but prerequisites are still required.

In summary, there are total 19 functions and 9 of them can be allocated to both human and drones, i.e., m = 9. An illustrative example of teamwork with adaptive function allocation is presented in Fig. 8 through Fig. 11. Figs. 8 and 9 depict drone trajectories and human workload over simulation time, respectively. In Fig. 10, the majority of guidance functions and fault functions are allocated to the drones and humans, respectively. However, some environmental anomaly functions, such as Get anomaly and Set anomaly, are allocated to both. It shows the adaptability of the function allocation policy to the team's state. Even though the human is not involved in Get mission in this specific case, they are still required to perform Confirm mission and Report mission to interact with the drones and the command center, respectively. Fig. 11 illustrates the distance to each next target point for the drones. This figure demonstrates that physical variables, such as drone positions, can be simulated alongside function allocation decisions over time. We present the details of the CWM elements for the target scenario in Table 2.

There may be criticism that the presented model is too simple and may not fully reflect complex human factors.

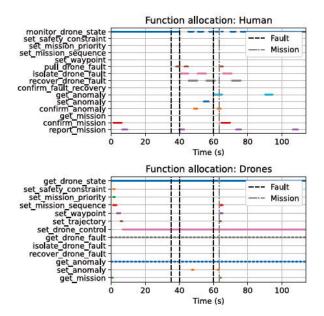


FIGURE 10. The allocated functions (represented as solid lines) over time for the human and drones.

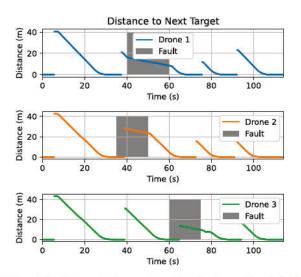


FIGURE 11. The distance to the next target for the drones. The shaded areas represent the time duration affected by the fault before recovery by the team.

However, we would like to note that the strength of the proposed framework lies in its flexibility. Any parts of the CWM can be updated without affecting other parts. Thus, the function allocation analysis can be re-assessed without the need for exhaustive rebuilding of the entire model. This flexibility provides a fast and effective initial HAT design process.

#### C. OPTIMIZATION USING REINFORCEMENT LEARNING

The purpose of the optimization is to allocate functions while maximizing the reward from the teamwork. We present the problem formulation to formally address the optimal adaptive function allocation. The optimization incorporates the

Function	GF	Teamwork	Responsibility	Duration by H D	Workload	Update Rate
Monitor drone state	Navigation	Monitor	Human	0.1 sec   N/A	Low	0.1 sec
Get drone state	Navigation	Monitor	Drone	N/A   0.1 sec	N/A	0.1 sec
Set safety constraint	Guidance	Command	Human	5 sec   1 sec	Mid	N/A
Set mission priority	Guidance	Command	Human	5 sec   1 sec	Mid	N/A
Set mission sequence	Guidance	Command	Drone	10 sec   2 sec	High	N/A
Set waypoint	Guidance	Command	Drone	10 sec   2 sec	High	N/A
Set trajectory	Control	N/A	Drone	N/A 1 sec	N/A	N/A
Set drone control	Control	Control	Drone	N/A   0.1 sec	Mid	0.1 sec
Get drone fault	Fault	N/A	Drone	1 sec N/A	N/A	1 sec
Pull drone fault	Fault	Pull	Human	2 sec N/A	Mid	N/A
Isolate drone fault	Fault	Monitor	Human	5 sec   30 sec	High	N/A
Recover drone fault	Fault	Push	Human	5 sec   30 sec	High	N/A
Confirm drone fault	Fault	Confirm	Human	2 sec N/A	Mid	N/A
Get anomaly	Anomaly	Monitor	Human	3 sec   1 sec	Mid	30 sec
Set anomaly	Anomaly	Push	Human	3 sec 1 sec	Mid	N/A
Confirm anomaly	Anomaly	Confirm	Human	2 sec N/A	Mid	N/A
Get mission	Mission	Pull	Human	5 sec   1 sec	High	N/A

Human

Human

5 sec

3 sec

N/A

N/A

TABLE 2. The list of physical functions in the computational work model (GF: generalized function and H | D: human or drone).

Anomaly detection ranges are set to 100 m and 20 m for human and drones (from the drone position), respectively.

Confirm

Push

CWM in (3) and the performance metrics to find the statedependent optimal function allocation. For the target scenario, we formulate a weighted reward function with three elements: mission completion time T as the task performance, workload  $w_k$ , and the coherency in function allocations  $c_k \in \mathbb{R}$  for  $k \in \{0, 1, ..., T\}$ :

Confirm mission

Report mission

$$r_k = -\left(\mu_1 \cdot 1 + \mu_2 w_k^2 + \mu_3 c_k\right) \tag{4}$$

Mission

Mission

where

$$c_k := ||a_k - a_0||^2, \quad \forall k \in \{0, \dots, T\}$$
 (5)

and  $\sum_{i=1}^{3} \mu_i = 1, \mu_i \ge 0 \ \forall i \in \{1, 2, 3\}$  are the weight parameters of the reward function. The workload is squared (i.e., using  $w_k^2$ ) in the reward function (4) to penalize the high workload. The coherency  $c_k$  is designed to generate penalty for any function allocation changes from the initial allocation to avoid frequent function reallocation. In other words, the coherency  $c_k$  is a measure of the stability in function allocation. For instance, if the function allocation is erratic, it may negatively impact human cognition [17]. Note that the initial function allocation,  $a_0$ , is used instead of the previous function allocation,  $a_{k-1}$ , in (5) to compute coherency based on our pre-pilot human-subject experiment. During the experiments, the human-subjects tended to be more familiar with the initial function allocation rather than the function allocation in the previous time step. As an initial condition, we assign all guidance functions to the drones and SA related functions to the human. The navigation and and control functions are fixed to each agent. By modifying  $\mu_i$ , we can investigate the trade-off space of the function allocation.

The CWM presented in (3) and the reward function in (4) can be formulated as a Markov decision process (MDP). As a result, we can leverage reinforcement learning (RL) to determine the optimal adaptive function allocation based on

the given CWM and reward function. Note that our proposed framework is not limited to a specific optimization technique. Instead, any alternative optimization methods capable of solving the MDP optimization problem can be applied. We opt for RL due to its generality and widespread accessibility within the research community. The optimization problem is formulated as:

Mid

Mid

N/A

30 sec

$$\max V_{\pi}(s_0)$$
subject to  $s_{k+1} = g(s_k, a_k), \quad w_k \le w_{\max}$ 

$$\forall k \in \{0, 1, \dots, T-1\}$$
 (6)

where

$$V_{\pi}(s) := \mathbf{E}\left[\sum_{k=0}^{T} \gamma^{k} r_{k} \mid s_{0} = s\right]$$
 (7)

is called the value function.  $\gamma \in [0,1]$  is the discount-rate. We use  $\gamma = 1$  since we focus on an episodic scenario with a finite termination time and consider all time history equally.  $w_{\max} = 6$  is the workload constraint.  $\pi(s_k)$  denotes the state-dependent policy that is equivalent to the function allocation in this optimization problem. In other words, the proposed framework decides the function allocation based on the current situation. The final output of the proposed framework is the optimal function allocation  $a^*$  and the corresponding optimal policy  $\pi^*$ :

$$a^* = \pi^*(s) := \arg\max_{\pi} V_{\pi}(s).$$
 (8)

The proposed framework uses the model to simulate the scenarios, but the model might be too complex to be represented explicitly as the function in (3). Thus, we choose a RL method that utilizes experience (i.e., sample sequences of the states, function allocations, and rewards). Since the state and function allocation are also complex, we can apply a

linear method to approximate the action-value function [40]. It simplifies the problem while maintaining information that allows the optimization problem to be solved efficiently. The action-value function Q is linearized with the feature and weight parameters as  $\hat{Q}$ :

$$Q_{\pi}(s, a) := \mathbf{E} \left[ r_{k+1} + \gamma V_{\pi}(s_{k+1}) \mid s_k = s, a_k = a \right]$$
 (9)

$$\hat{Q}_{\pi}(s,a) = \sum_{i=1}^{d} \theta_i \phi_i(s,a) = \theta^T \phi(s,a)$$
 (10)

where  $\theta = [\theta_1, \dots, \theta_d]^T \in \mathbb{R}^d$  denotes the unknown weight vector and  $\phi_i(s, a) = [\phi_1, \dots, \phi_d]^T \in \mathbb{R}^d$  is the predefined feature. The feature is designed as:

$$\phi(s, a) = \begin{bmatrix} 1, s^T, a^T, s_1 a_1, \dots, s_i a_j, \dots, s_n a_m \end{bmatrix}^T$$
 (11)

where  $i \in \{1, ..., n\}$ ,  $j \in \{1, ..., m\}$ ,  $s = [s_1, ..., s_n]^T$ , and  $\mathbf{a} = [a_1, ..., a_m]^T$ . Note d = 1 + n + m + nm for the feature in (11). For each episode, we can update the weight using the episodic semi-gradient Sarsa method [40] (p. 244).

$$\theta_{k+1} = \theta_k + \alpha \left[ r_k - \hat{Q}(s_k, a_k) \right] \phi(s_k, a_k) \tag{12}$$

where  $\theta_k \in \mathbb{R}^d$  denotes the weight vector at time step k and  $\alpha > 0$  is the step size. The  $\epsilon$ -greedy method is used for training to balance between exploration and exploitation of the current value function, i.e., choosing any of non-optimal function allocations with the probability of  $\epsilon$ . Then, the optimal function allocation is obtained as:

$$a_k^* = \arg\max_{a_k} \hat{Q}(s_k, a_k)$$
 (13)

with breaking ties randomly.

## IV. NUMERICAL SIMULATION

We present a series of numerical simulation results to demonstrate the effectiveness of the proposed framework in conducting a trade-off study. The specific details of the HAT scenario are provided in Table 3. The objective of the trade-off study is to evaluate a set of weight parameters  $\{\mu_1, \mu_2, \mu_3\}$  in (4) upon specific demands. To establish baselines for team performance, we compare two fixed function allocations. The study cases are outlined as follows.

- (Human) All functions are assigned to the human.
- · (Autonomy) All functions are assigned to the drones.
- (Time) The mission completion time is the only performance metric, i.e.,  $\{\mu_1, \mu_2, \mu_3\} = \{1, 0, 0\}$ .
- (Workload) Workload is the only performance metric, i.e., {μ<sub>1</sub>, μ<sub>2</sub>, μ<sub>3</sub>} = {0, 1, 0}.
- (Coherency) Coherency in function allocations is the only performance metric, i.e., {μ<sub>1</sub>, μ<sub>2</sub>, μ<sub>3</sub>} = {0, 0, 1}.
- (Balance) All performance metrics are balanced with  $\{\mu_1, \mu_2, \mu_3\} = \{0.8, 0.1, 0.1\}.$

For each case, we used 3000 random episodes for training and an identical set of 100 random episodes for testing. We thoroughly analyzed all performance metrics, including the mission completion time, workload, and coherency in function allocations for each case in Fig. 12. The workload and

TABLE 3. The HAT scenario is used for RL training and testing.

Element	Value
Start position	$[0,0,0]^T$
Termination time	300 sec
Initial target points	$ \begin{bmatrix} [100,5,10]^T, [5,100,10]^T, \\ [100,95,20]^T, [110,70,10]^T, \\ [75,120,15]^T, [30,90,10]^T, \\ [80,75,15]^T, [110,120,20]^T, \\ [120,110,15]^T \\ (\pm [5,5,5]^T \text{ for each}) \end{bmatrix} $
Static obstacle position, radius	$ \begin{bmatrix} 45, 105, 12 \end{bmatrix}^T, 10 \\ [10, 55, 5]^T, 10 \\ [60, 10, 5]^T, 10 $
Faults timing	Drone 1: $45 \pm 10$ sec Drone 2: $35 \pm 10$ sec Drone 3: $55 \pm 10$ sec Occurrence probability: 0.5 for each fault
Anomaly position, radius	Anomaly 1: $[60, 80, 13]^T$ , 10 Anomaly 2: $[100, 40, 10]^T$ , 10 Anomaly 3: $[100, 120, 17]^T$ , 20
Update mission position, timing	$[75, 120, 15]^T$ , $[30, 90, 10]^T$ , $[160, 100, 15]^T$ , at 65 sec $(\pm [5, 5, 5]^T$ for each)
Step size	$\alpha = 10^{-4}$
$\epsilon$ -greedy	$\epsilon = 0.05$

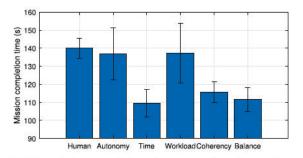
The values denoted by  $\pm$  represent uniformly distributed random values within the corresponding range. The position unit is measured in meters (m).

coherency levels were averaged for each episode, resulting in 100 samples for each performance metric (one sample for each episode). Furthermore, a post-analysis was conducted to offer additional insights and information.

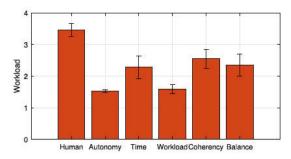
In Fig. 12(a), the mission completion time T results are presented for each case. Among the two baseline cases, Human and Autonomy, the relatively worse mission completion time is observed, which indicates their failure to effectively leverage the diverse capabilities of the team members to reduce mission completion time. Notably, the Time case achieves the best mission completion time, demonstrating the framework's ability to allocate functions to meet specific requirements. The Workload case, which allocates a majority of functions to drones to reduce workload but sacrifices team performance, exhibits a very high mission completion time. Conversely, the Coherency case, with nearly fixed function allocation due to its reward function, demonstrates a lower mission completion time compared to the other fixed function allocation cases. The observation supports that the initial guess of function allocation (i.e., guidance functions by the drones and SA related functions by the human) is a reasonable choice for team performance. The Balance case achieves the second-best mission completion time despite its weighted considerations on workload and coherency. This fact shows its effectiveness in achieving a balance among performance metrics.

To further investigate the trade-off, the averaged work-load is presented in Fig. 12(b). The Human case records

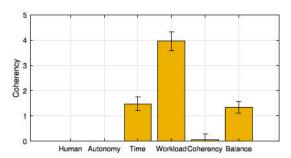




(a) The mission completion time comparison for each case.



(b) The averaged workload comparison for each case.



(c) The averaged coherency in function allocations comparison for each case.

FIGURE 12. The mission completion time, averaged workload, and averaged coherency in function allocations comparison results for each case. The error bars denote 1-σ standard deviations.

the highest workload, requiring significant human interventions. On the other hand, the lowest workload levels are observed in the Autonomy and Workload cases, where most functions are allocated to the drones, minimizing human engagement but leading to higher mission completion time. The remaining three cases (Time, Coherency, and Balance) demonstrate moderate workload levels compared to the extreme cases. Interestingly, the Time case, which does not explicitly account for workload, results in relatively low workload. This finding suggests a potential correlation between mission completion time and workload in the HAT scenario. For instance, high workload could induce delayed functions for the human, which deteriorates the mission completion time. Hence, workload should be maintained within the saturation range to expedite the mission progress.

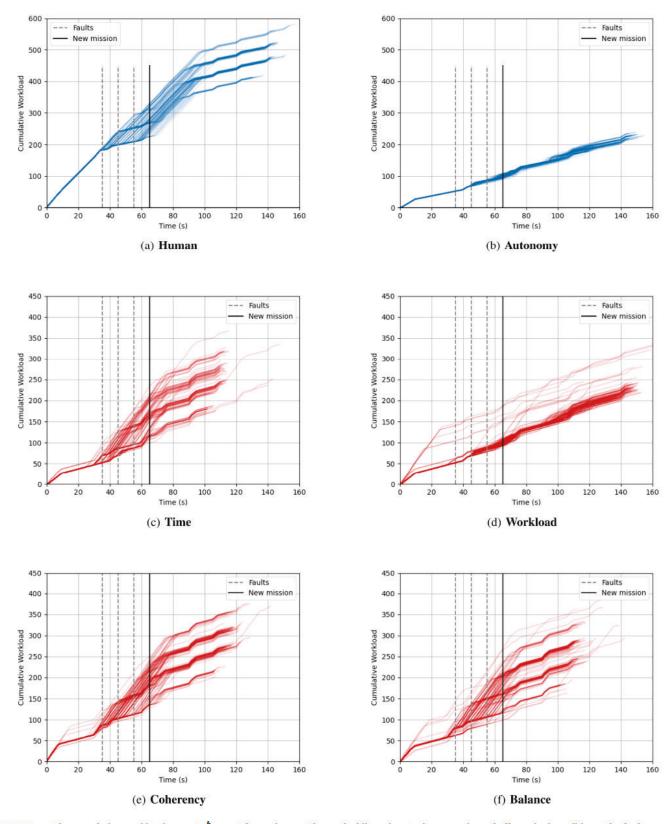
The coherency of function allocations  $c_k$  examines the stability of the team structure. A lower averaged  $c_k$  indicates that the team adheres more closely to the initial function allocation. The comparison results are shown in Fig. 12(c). The Human and Autonomy cases represent two extremes with a consistently zero averaged coherency variable  $c_k$  at every time step. The Coherency case exhibits a very low coherency level, though not precisely zero, which may be attributed to approximation errors in the action-value function in (10). In contrast, the Workload case consistently allocates all functions to the drones to reduce workload. This fact results in the high  $c_k$  in average since the SA related functions are initially allocated to the human. The Time and Balance cases occasionally deviate from the initial function allocation, but they maintain stable function allocations over time. The Balance case is even more stable compared to the Time case since it directly considers the coherency as a part of its reward.

The cumulative workload results  $cw_k := \sum_{i=0}^k w_i \Delta t$  are presented in Fig. 13. Note that the branches of the results are induced by the probability of the fault occurrence in Table 3. These results reveal that each case chooses different function allocation strategies. For instance, in time-wise efficient cases, including Time and Balance, the human engages in Get anomaly at the beginning phase more often. This policy enables the team to detect anomalies early by leveraging the human's capability. Then, the drones can obtain time-optimal trajectories based on the obtained anomaly information. Even though the Human case also allocates Get anomaly to the human at the early phase, it is not time-efficient since the human is overloaded, and the delayed functions cause degradation in mission completion time. We can examine the workload saturation by observing the slope of each case in Fig. 13. The adaptive function allocation cases can alleviate the workload saturation by delegating some functions to the drones based on the current situation.

A summary of the numerical results is presented in Table 4. The comparison includes two baselines (Human and Autonomy) alongside the adaptive function allocations (Time, Workload, Coherency, and Balance), based on the mission completion time, workload, and coherency. The results indicate that there is no single dominant function allocation for all metrics in the given scenario. Instead, the proposed framework allows for the consideration of various performance metrics with different weights in the trade-off study. The proposed framework empowers the HAT designer to identify the best adaptive function allocation strategy by accommodating various performance metrics quantitatively.

#### V. DISCUSSION

We propose a computational framework for optimizing function allocation in HAT to enhance cooperation between human and autonomous agents. The proposed framework identifies and explores all possible function allocations using a computational work model to seek the optimal function allocation



**FIGURE 13.** The cumulative workload  $cw_k = \sum_{i=0}^k w_i \Delta t$  for each case. The vertical lines denote the mean time of off-nominal conditions. The fault occurrence probability is 0.5 for each one. Note that (a) and (b) have wider y-axis ranges.



TABLE 4. The performance metric differences (mean) between two baselines and four adaptive function allocations for the mission completion time (T), workload (W), and coherency in function allocations (C).

	Human	Autonomy
Time	T: -30.4660	T: -27.3110
	W: -1.1768	W: +0.7563
	C: +1.4838	C: +1.4838
Workload	T: -2.7400	T: +0.4150
	W: -1.8654	W: +0.0677
	C: +3.9622	C: +3.9622
Coherency	T: -24.4320	T: -21.2770
	W: -0.9075	W: +1.0256
	C: +0.0599	C: +0.0599
Balance	T: -28.4430	T: -25.2880
	W: −1.1048	W: +0.8283
	C: +1.3510	C: +1.3510

A decrease (-) indicates improvement, while an increase (+) represents degradation, respectively. Top two results are presented in bold among four row cases.

that can adapt for the dynamically changing situation. A key strength of the proposed framework lies in its flexibility to build and update the computational work model's parameters while considering any performance metrics.

The proposed framework offers the advantage of personalizing function allocation due to its flexible structure. For instance, the work model parameters in Table 2 can be customized based on individual user characteristics. An expert human may exhibit shorter duration and lower workload for specific physical functions [21]. However, this flexibility also implies that determining the human model parameters may pose additional challenges. As a result, the proposed framework may not provide an immediate solution for determining the best function allocation. Instead, it serves as a valuable tool to explore function allocations based on the available work model and performance metrics. The work model can be updated through human-subject experiments in the final design phase, considering factors such as personality, interface design, and scenarios. Then, the proposed framework can perform a computational trade-off study of the HAT design instead of conducting it with additional human-subject experiments that are expensive in terms of time and cost.

We acknowledge that the proposed optimal function allocation framework would benefit from further elaboration of its human work model, particularly concerning cognitive state modeling. The significance of human cognitive states in function allocation has been studied [35]. For workload and SA, computational models based on the well-known cognitive architecture adaptive control thought-rational (ACT-R) are available [36], [41]. These cognitive models could enhance the CWM by simulating human behaviors, such as making mistakes and unintentionally losing SA. Another crucial cognitive state is human trust in autonomy. While we assumed that humans always accept the allocated functions by autonomy, this may not hold if trust is low. Implementing computational

approaches for dynamic trust in the CWM can calibrate trust in HAT scenarios [35].

#### VI. CONCLUSION

We proposed a quantitative framework for optimal adaptive function allocation in human-autonomy teaming (HAT). The framework leverages systematic approaches in cognitive engineering and computational work modeling. By considering performance metrics such as mission completion time, workload, and coherency in function allocations, the proposed framework can compute the optimal situation-dependent function allocation. Additionally, the framework allows for flexible HAT design and easy adaptation to changes in the computational work model during the early design phase. An illustrative numerical example with a simulated human and drone fleet teaming scenario demonstrated the effectiveness of the proposed optimal function allocation framework.

#### REFERENCES

- T. O'Neill, N. McNeese, A. Barron, and B. Schelble, "Humanautonomy teaming: A review and analysis of the empirical literature," *Hum. Factors*, vol. 64, no. 5, pp. 904–938, 2022.
- [2] J. B. Lyons, K. Sycara, M. Lewis, and A. Capiola, "Human-autonomy teaming: Definitions, debates, and directions," *Front. Psychol.*, vol. 12, 2021, Art. no. 589585.
- [3] M. Johnson, M. Vignati, and D. Duran, "Understanding humanautonomy teaming through interdependence analysis," in *Proc. Symp. Hum. Autonomy Teaming*, 2018, pp. 1–20.
- [4] K. Christoffersen and D. D. Woods, "How to make automated systems team players," Adv. Hum. Perform. Cogn. Eng. Res., vol. 2, pp. 1–12, 2001.
- [5] E. M. Roth, C. Sushereba, L. G. Militello, J. Diiulio, and K. Ernst, "Function allocation considerations in the era of human autonomy teaming," J. Cogn. Eng. Decis. Mak., vol. 13, no. 4, pp. 199–220, 2019.
- [6] M. M. Cummings, "Man versus machine or man + machine?," IEEE Intell. Syst., vol. 29, no. 5, pp. 62–69, Sep./Oct. 2014.
- [7] P. M. Fitts et al., Human Engineering for an Effective Air Navigation and Traffic Control System. Washington, DC, USA: Nat. Res. Council, 1951
- [8] R. Parasuraman, T. Sheridan, and C. Wickens, "A model for types and levels of human interaction with automation," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans.*, vol. 30, no. 3, pp. 286–297, May 2000.
- [9] P. A. Hancock and M. H. Chignell, "Adaptive function allocation by intelligent interfaces," in *Proc. 1st Int. Conf. Intell. User Interfaces*, 1993, pp. 227–229.
- [10] S. W. A. Dekker and D. D. Woods, "MABA-MABA or Abracadabra? Progress on human-automation co-ordination," *Cogn. Technol. Work*, vol. 4, pp. 240–244, 2002.
- [11] T. Inagaki, "Adaptive automation: Sharing and trading of control," in *Handbook of Cognitive Task Design*, Mahwah, NJ, USA: Lawrence Erlbaum Associates Publishers, 2003, pp. 147–169, doi: 10.1201/9781410607775.ch8.
- [12] A. R. Pritchett, K. M. Feigh, S. Y. Kim, and S. K. Kannan, "Work models that compute to describe multiagent concepts of operation: Part 1," J. Aerosp. Inf. Syst., vol. 11, no. 10, pp. 610–622, 2014.
- [13] M. IJtsma, L. M. Ma, A. R. Pritchett, and K. M. Feigh, "Computational methodology for the allocation of work and interaction in human-robot teams," J. Cogn. Eng. Decis. Mak., vol. 13, no. 4, pp. 221–241, 2019.
- [14] K. J. Vicente, Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work. Boca Raton, FL, USA: CRC Press, 1999
- [15] C. M. Macal and M. J. North, "Tutorial on agent-based modelling and simulation," J. Simul., vol. 4, no. 3, pp. 151–162, 2010.
- [16] J. M. Bradshaw, R. R. Hoffman, D. D. Woods, and M. Johnson, "The seven deadly myths of "autonomous systems"," *IEEE Intell. Syst.*, vol. 28, no. 3, pp. 54–61, May/Jun. 2013.

- [17] A. R. Pritchett, S. Y. Kim, and K. M. Feigh, "Measuring humanautomation function allocation," *J. Cogn. Eng. Decis. Mak.*, vol. 8, no. 1, pp. 52–77, 2014.
- [18] A. M. Annaswamy, P. P. Khargonekar, F. Lamnabhi-Lagarrigue, and S. K. Spurgeon, Cyber-Physical-Human Systems: Fundamentals and Applications. Piscataway, NJ, USA: Wiley-IEEE Press, 2023.
- [19] A. M. Annaswamy and Y. Yildiz, Cyber-Physical-Human Systems. Cham, Switzerland: Springer, 2021, pp. 497–508.
- [20] E. Eraslan, Y. Yildiz, and A. M. Annaswamy, "Shared control between pilots and autopilots: An illustration of a cyberphysical human system," *IEEE Control Syst. Mag.*, vol. 40, no. 6, pp. 77–97, Dec. 2020.
- [21] S. Byeon, J. Choi, Y. Zhang, and I. Hwang, "Stochastic-skill-level-based shared control for human training in urban air mobility scenario," ACM Trans. Hum.-Robot Interaction, Jun. 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3603194
- [22] M. S. Yuh, S. Byeon, I. Hwang, and N. Jain, "A heuristic strategy for cognitive state-based feedback control to accelerate human learning," *IFAC-PapersOnLine*, vol. 55, no. 41, pp. 107–112, 2022.
- [23] Z. Tu, Y. Fang, Y. Leng, and C. Fu, "Task-based human-robot collaboration control of supernumerary robotic limbs for overhead tasks," *IEEE Robot. Automat. Lett.*, vol. 8, no. 8, pp. 4505–4512, Aug. 2023.
- [24] N. Kousi, D. Dimosthenopoulos, S. Aivaliotis, G. Michalos, and S. Makris, "Task allocation: Contemporary methods for assigning human-robot roles," in *Proc. 21st Century Ind. Robot: When Tools Become Collaborators*, 2022, pp. 215–233.
- [25] A. A. Tadesse, K. J. Wang, and C. J. Lin, "Task allocation problem between human–robot collaboration team," in *Intelligent and Transfor*mative Production in Pandemic Times. Cham, Switzerland: Springer, 2023, pp. 473–482.
- [26] A. Pupa, C. T. Landi, M. Bertolani, and C. Secchi, "A dynamic architecture for task assignment and scheduling for collaborative robotic cells," in *Proc. Hum.-Friendly Robot.*, 2021, pp. 74–88.
- [27] B. Rabta, C. Wankmüller, and G. Reiner, "A drone fleet model for last-mile distribution in disaster relief operations," *Int. J. Disaster Risk Reduction*, vol. 28, pp. 107–112, 2018.
- [28] S. Mirri, C. Prandi, and P. Salomoni, "Human-drone interaction: State of the art, open issues and challenges," in *Proc. ACM SIGCOMM Workshop Mobile AirGround Edge Comput.*, Syst., Netw., Appl., 2019, pp. 43–48.
- [29] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Hum. Factors*, vol. 37, no. 1, pp. 32–64, 1995.
- [30] O. McAree, J. Aitken, and S. Veres, "Quantifying situation awareness for small unmanned aircraft: Towards routine beyond visual line of sight operations," *Aeronautical J.*, vol. 122, no. 1251, pp. 733–746, 2018.
- [31] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Hum. Factors*, vol. 39, no. 2, pp. 230–253, 1997.
- [32] D. Jenkins, N. Stanton, and G. Walker, Cognitive Work Analysis: Coping With Complexity, 1st ed. London, U.K.: CRC Press, 2009.
- [33] K. M. Feigh, A. R. Pritchett, S. Mamessier, and G. Gelman, "Generic agent models for simulations of concepts of operation: Part 2," J. Aerosp. Inf. Syst., vol. 11, no. 10, pp. 623-631, 2014.
- [34] B. Mok and W. Ju, "The push vs pull of information between autonomous cars and human drivers," in Proc. Adjunct Proc. 6th Int. Conf. Automot. User Interfaces Interactive Veh. Appl.,, 2014, p. 1–5.
- [35] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "Situation awareness, mental workload, and trust in automation: Viable, empirically supported cognitive engineering constructs," J. Cogn. Eng. Decis. Mak., vol. 2, no. 2, pp. 140–160, 2008.
- [36] S. Jo, R. Myung, and D. Yoon, "Quantitative prediction of mental workload with the ACT-R cognitive architecture," *Int. J. Ind. Ergonom.*, vol. 42, no. 4, pp. 359–370, 2012.
- [37] J. Rasmussen, "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 3, pp. 257–266, May/Jun. 1983.

- [38] K. J. Behymer and J. M. Flach, "From autonomous systems to sociotechnical systems: Designing effective collaborations," She Ji: J. Des., Econ., Innov., vol. 2, no. 2, pp. 105–114, 2016.
- [39] J. Sun, J. Tang, and S. Lao, "Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm," *IEEE Access*, vol. 5, pp. 18382–18390, 2017.
- [40] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
- [41] T. Nguyen, C. P. Lim, N. D. Nguyen, L. Gordon-Brown, and S. Nahavandi, "A review of situation awareness assessment approaches in aviation environments," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3590–3603, Sep. 2019.



SOOYUNG BYEON (Member, IEEE) received the bachelor's and master's degrees in aerospace engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2012 and 2014, respectively. From 2014 to 2019, he was a Senior Research Engineer with Satrec Initiative Company, Ltd., Daejeon. He is currently a Research Assistant with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, USA. His research interests include human-automation interaction, human

behavior modeling, imitation learning, and applied experiments.



JOONWON CHOI (Graduate Student Member, IEEE) received the bachelor's degree in mechanical and aerospace engineering and master's degree in mechanical engineering from the Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea, in 2019 and 2021, respectively. From 2019 to 2020, he was a Researcher with the Autonomous Systems Laboratory, UNIST. He is currently a Research Assistant with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, USA. His research interests

include human-automation interaction, data-driven reachability analysis, and applied experiments.



INSEOK HWANG (Member, IEEE) received the Ph.D. degree in aeronautics and astronautics from Stanford University, Stanford, CA, USA, in 2004. He is currently a Professor with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, USA. His research interests include modeling, estimation, and control of cyber-physical systems and their applications to safety critical systems such as aircraft/spacecraft/unmanned aerial systems, air traffic management, and multi-agent systems. For his

research, he leads the Flight Dynamics and Control/Hybrid Systems Laboratory with Purdue University.