A Hybrid Systems-Based Hierarchical Control Architecture for Heterogeneous Field Robot Teams

Chanyoung Ju, *Student Member, IEEE*, and Hyoung Il Son, *Senior Member, IEEE*

***Abstract*—Field robot systems have recently been applied in a wide range of research fields. Further automation, development, and activation of such systems require cooperation among heterogeneous robots. Classical control theory is inefficient in managing large-scale complex dynamic systems. Therefore, a discrete-event system based on the supervisory control theory must be introduced to overcome this limitation. In this article, we propose a hybrid system-based hierarchical control architecture using a supervisory control-based high-level controller and a traditional control-based low-level controller. The hybrid system and its dynamics are modeled through a formal method, called hybrid automata, and the behavior specifications are designed to express the control objectives for cooperation. In addition, modular supervisors that are more scalable and maintainable than a centralized supervisory controller were synthesized. The proposed hybrid system and hierarchical control architecture were implemented, validated, and evaluated for performance through a physics-based simulator and field tests. The experimental results confirmed that the robot team satisfied the given specifications and presented systematic results, validating the efficiency of the proposed control architecture.**

***Index Terms*—Field robotics, heterogeneous multirobot, hybrid automata, hybrid systems, supervisory control.**

I. Introduction

In RECENT years, field robots have been widely deployed in nuclear plants, agriculture [1], construction, underwater [2], and military applications. Their usefulness has attracted a considerable amount of attention from environmental scientists and robotics engineers. However, field robotics research is complicated because agents need to perform tasks in outdoor, unstructured, and unknown environments. Nevertheless, various types of robots, such as unmanned ground vehicles (UGVs), unmanned aerial vehicles (UAVs), mobile manipulators, humanoids, and hybrid UAVs [3]), have been developed to cope with disasters and substitute human labor (e.g., by automating agricultural work, such as spraying, cultivation, grafting, and harvesting) [4]–[6]. However, issues of application and commercialization have emerged due to the considerable gap between the current robot technology and the requirements for the use in the field. In particular, the development of control systems for the cooperation of homogeneous and heterogeneous field robots remains a significant challenge.

To achieve the eventual autonomy of temporal–spatial missions (e.g., mapping, sensing, sampling, and monitoring) using field robots, novel control frameworks that enable multiple heterogeneous robots to collaborate must be developed. However, the classical control theory based on differential equations has some limitations in handling large-scale complex dynamic systems, such as heterogeneous field robots. For example, the traditional control approach has been used to study multiple UAV systems for remote sensing [7], autonomous underwater vehicles (AUVs) for distributed formation tracking [8], path planning [9], and heterogeneous multirobot systems for exploration, sampling [10], and mapping [11]. These approaches are inadequate for handling reactivity, scalability, maintainability, modularity, and systematic analysis when field robots are added to or removed from a team, or when a given mission is modified (e.g., a change of tasks in local or global areas), even if robustness and adaptiveness are considered in the controller. In addition, field robot systems need to meet the practical requirements of a dynamic environment, which reflects asynchronous events.

One approach to solving this problem involves the use of discrete-event systems (DESs) and formal methods to systematically analyze the states, events, and behaviors of large-scale dynamic systems and design powerful controllers [12]. Formal methods typically include automata, Petri nets, temporal logic, behavior trees, and min–max algebra. Recently, the supervisory control theory (SCT), also known as the Ramadge–Wonham framework (RW framework), based on automata, has become an intensive approach to controlling DES behavior [13]. SCT has been proven to be efficient for large-scale dynamic systems, as this approach can design supervisors that meet the behavior specifications of plants and maximally enable events [14]. Consequently, SCT-based control systems have been developed for sumo robots [15], industrial UGVs [16], multicopters [17], patient support tables [18], and fuzzy DESs [19]. However, challenges pertaining to the debugging, usage, and application of formal methods to dynamic systems remain. More importantly, automata-based SCT, which is a formal method-based approach, requires an

Manuscript received February 1, 2021; revised June 26, 2021 and October 18, 2021; accepted November 26, 2021. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2018R1D1A1B07046948. This article was recommended by Associate Editor Y. Pan. *(Corresponding author: Hyoung Il Son.)*

The authors are with the Department of Rural and Biosystems Engineering and Interdisciplinary Program in IT-Bio Convergence System, Chonnam National University, Gwangju 61186, South Korea (e-mail: cksdud15@gmail.com; hison@jnu.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3133631.

Digital Object Identifier 10.1109/TCYB.2021.3133631

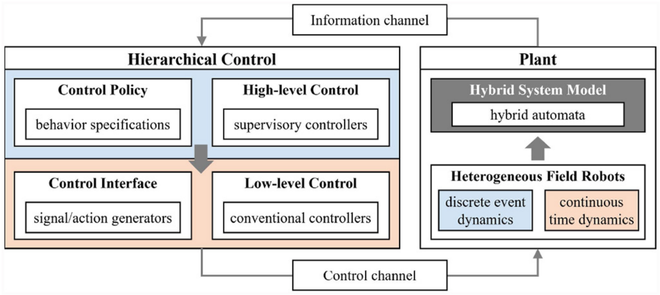

Fig. 1. Concept of the HSHC architecture composed of a high-level control and a low-level control.

implementable solution without event generators because the supervisory controller enables and disables controllable events while observing events occurring in the plant.

A new approach using DES and SCT is therefore required for the explicit cooperation of heterogeneous field robots. In this article, we propose a hybrid system-based hierarchical control (HSHC) architecture that combines classical control theory based on a continuous-time system (CTS) together with DES-based SCT to overcome these weaknesses. Because HSHC attempts to integrate the advantages of each system and control theory, it overcomes many of the existing limitations (i.e., computational complexity, modeling uncertainties, or differences in dynamics) of large-scale complex dynamic systems with unstructured and unknown environments. In addition, it has the advantage of systematically analyzing the system behaviors (i.e., state transitions), event occurrence and sequence, and control behaviors under any condition. Hence, HSHC is an ideal control architecture for heterogeneous field robot teams, where system modeling and controller design are extremely challenging.

A. Objective of the Study

The objective of this study is to control complex dynamic systems, such as heterogeneous field robot teams. To achieve this, an HSHC architecture is proposed for heterogeneous field robot cooperation. The hybrid system is a coupling scheme consisting of continuous-time dynamics and discrete-event dynamics. The concept of hierarchical control architecture is illustrated in Fig. 1. The plant is modeled with hybrid automata, taking into account the continuous-time and discrete-event dynamics of the system. Moreover, the behavior specifications are designed to represent the desired goals and control policies for the synthesis of supervisory controllers in the high-level control domain. The behavior specification is also expressed in the automata modeling method, and more details are introduced in Section IV. However, when controlling the robot system with continuous-time dynamics taken into consideration, a low-level controller is required based on conventional controllers. Therefore, we designed distributed swarm controllers using ordinary differential equations considering the continuous-time dynamics in the low-level control domain. The command of the high-level control is transmitted to the low-level control through a control interface that generates signals and actions. The control input generated by the hierarchical control architecture is passed to the plant via the control channel. The output of the hybrid system model-based heterogeneous field robots (measured values, state variables, events, etc.) is transmitted to the hierarchical control architecture through the information channel. Consequently, the proposed control architecture based on the supervisory controllers and hybrid system model can provide a feedback control loop through the control and information channels.

In our previous work [20], [21], we also designed a centralized supervisory controller to control a heterogeneous agricultural field robot system. These studies constituted our first investigation of modeling and control based on DESs for multirobot systems. In addition, the finite-state automata theory was applied to model the entire system, focusing only on the behavior of field robots for collaboration. In other words, these works served as preliminary studies to verify the feasibility of the HSHC proposed in this article. Here, we modeled a field robot system based on the dynamics of each mobile robot and extended the previous research using hybrid automata and a hybrid system. Furthermore, the modular supervisor was designed to be more scalable, maintainable, and superior to the centralized supervisory controller to manage large-scale dynamic systems. In summary, the goal of our research is to develop an HSHC system that can systematically model, control, and analyze heterogeneous field robots based on a novel approach. The proposed hybrid system and modular supervisor were implemented, validated, and evaluated in a physics-based simulator and in a field environment, and its systematic results and effectiveness are presented in this article.

B. Related Works

In this section, we review related studies based on formal methods for multirobot control, particularly, DES and SCT. SCT, a theory proposed by Ramadge and Wonham in 1987, introduced a methodology for controlling DESs [22]. It has been applied mainly in the field of manufacturing automation [23]; however, it has recently been introduced into a cyber-physical robotic system (i.e., a dynamic system). In task allocation for cooperative robot teams, the supervisory controller was designed to accommodate flexibility in task assignments, robot coordination, and tolerance of failures and repairs [24]. Liu *et al.* [25] designed a control architecture for the application of heterogeneous multirobot teams to urban search and rescue and presented simulation results. They found that this approach was effective for multirobot control in field applications and was robust to varying scenarios and increasing team size. Nevertheless, most research on the supervisory control of multiple robots has been aimed at autonomous navigation [26] through behavior coordination [27] and formation control [28]. These works focus on how to model the DES and analyze the supervisory controller rather than the implementation or demonstration of a robot system.

Examples of the implementation and application of SCT include warehouse automation [29], agricultural UAVs [30], exploration [31], theme park vehicles [32], swarm robotics [33], and attempts to integrate self-development tools and frameworks [34]–[38]. Most of the aforementioned studies employ simple supervisory control through an automata model and do not investigate the implementation problem by combining it with a CTS. In recent research, for example, a learning-based synthesis approach [39], a probabilistic DES under partial observation [40], fuzzy DES-based shared control [41], robust supervisory-based control [42], and a multilevel DES for bus structures [43] have been studied for implementation in the robotics field based on DES and SCT, but these are still in simple simulation stages.

Hybrid systems or hybrid control architectures for controlling robots have been studied as an alternative approach, but these studies did not deal with the heterogeneity, scalability, and systematic approach for complex dynamic systems [44]–[47]. Recently, studies on various processes, such as warehouse automation using logistic robotic networks [48], secure recovery procedures for manufacturing systems [49], and automatic controller code generation for swarm robotics [50] have also been conducted to expand the SCT and apply it to cyber–physical systems. These studies, in which supervisory control and a dynamic system based on DES were developed, do not guarantee solutions to the above-mentioned problems. Therefore, additional research is required to control, implement, and sustain large-scale dynamic systems using DES and SCT beyond the traditional control method in field robotics. To the best of our knowledge, no previous studies have examined the supervisory control of heterogeneous field robots. We have addressed this issue by proposing an HSHC architecture through a new perspective on supervisory control and classic control.

C. Structure of This Article

The remainder of this article is organized as follows. In Section II, we present a system modeling method using the ODE and hybrid automata. Section III introduces the process of designing a CTS-based low-level controller and distributed swarm control algorithm, consisting of obstacle avoidance control, formation control, path-following control, and a high-level controller through SCT. In Section IV, we describe the control architecture proposed in this article, the method of combining CTS and DES, hybrid automata-based system modeling, design of behavior specifications to achieve control objectives, and synthesis of the modular supervisor for heterogeneous field robots. Section V describes the implementation procedure, experimental setup, and experimental results using a physics-based simulator and field tests to evaluate the proposed control system. In Section VI, we summarize the conclusions and provide directions for future research.

II. Hybrid System Modeling

A. ODE-Based Low-Level Plant

We used dynamic UAV and UGV models to design a low-level controller based on a CTS. The kinematic and dynamic equations for each vehicle are described in Sections II-A1 and II-A2, respectively.

1) UAV Model: We consider N UAVs with three degrees of freedom (DOFs). The positions of the UAVs are represented by $p_i \in \Re^3$, $i = 1, 2, \ldots, N$. The flight control input of the UAVs is derived from the following dynamics and kinematic equations:

$$m_i \ddot{p}_i = -\lambda_i \zeta_i e_3 + m_i g e_3 + H_i \tag{1}$$

$$\mathcal{J}_i \dot{w}_i + S(w_i)\mathcal{J}_i w_i = \tau_i + a_i \tag{2}$$

$$\dot{\zeta}_i = \zeta_i S(w_i) \tag{3}$$

where $m_i > 0$ denotes the mass, $p_i := [p_1; p_2; \ldots; p_N] \in \Re^{3N}$ is the Cartesian center-of-mass position represented in the north-east-down (NED) inertial frame $\{O\}$, $\lambda_i \in \Re$ indicates the thrust control input along e_3 (with e_1, e_2, and e_3 indicating N, E, and D directions), $\zeta_i \in SO(3)$ denotes the rotational matrix describing the body frame $\{B\}$ of the UAV with respect to the inertial frame $\{O\}$, g is the gravitational constant, $e_3 = [0, 0, 1]^T$ denotes the basis vector representing the down direction and indicating that thrust and gravity act in the D direction, $\mathcal{J}_i \in \Re^{3\times 3}$ indicates the inertia matrix with respect to the body frame $\{B\}$, $w_i \in \Re^3$ is the angular speed of the UAV relative to the inertial frame $\{O\}$ represented in the body frame $\{B\}$, $\tau_i \in \Re^3$ indicates the attitude torque control input, $H_i, a_i \in \Re^3$ denote the aerodynamic perturbations, and $\Upsilon(\diamond) : \Re^3 \to SO(3)$ denotes the skew-symmetric operator defined subject to $\star, \bullet \in \Re^3$, $\Upsilon(\star)\bullet = \star \times \bullet$. For a typical UAV in flight, $H_i, a_i \approx 0$.

The relationships among the angular speed, thrust, and torque of each propeller for low-level control input are as follows:

$$\begin{pmatrix} \lambda_i \\ \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \end{pmatrix} = \begin{bmatrix} \kappa & \kappa & \kappa & \kappa \\ 0 & -L & 0 & L \\ L & 0 & -L & 0 \\ L & -L & L & -L \end{bmatrix} \begin{pmatrix} \omega_{i1}^2 \\ \omega_{i2}^2 \\ \omega_{i3}^2 \\ \omega_{i4}^2 \end{pmatrix} \tag{4}$$

where κ and L are constants that determine the relationship between the angular speed, thrust, and torque, L is the distance between the propeller and the center of the UAV, τ_{ij} denotes the torque in the roll, pitch, and yaw directions, and ω_{ij} denotes the angular speed of each propeller. As a result, when the desired thrust and torque are determined, the control input can be generated by controlling the motor using (4).

2) UGV Model: We consider the position of the ith UGV when the inertial frame is $\{O\}$, and the body frame is $\{D\}$. The kinematic and dynamic equations for the UGV are as follows:

$$\begin{aligned} \dot{x}_i &= \nu_i \cos(\psi_i) \\ \dot{y}_i &= \nu_i \sin(\psi_i) \\ \dot{\psi}_i &= \varphi_i \end{aligned} \tag{5}$$

$$D(x_i)\begin{pmatrix} \dot{\nu}_i \\ \dot{\varphi}_i \end{pmatrix} + Q(x_i, \varphi_i)\begin{pmatrix} \nu_i \\ \dot{\psi}_i \end{pmatrix} = \vartheta_i + \varrho_i \tag{6}$$

where x_i and y_i are the coordinates of the UGV with respect to the inertial frame $\{O\}$, ψ_i denotes the heading angle, ν_i and φ_i are the linear and angular speeds expressed in $\{D\}$, respectively, $\vartheta_i = [\vartheta_i^{\nu}, \vartheta_i^{\omega}]^T$ and $\varrho_i = [\varrho_i^{\nu}, \varrho_i^{\omega}]^T$ are the control input and the external force and torque, respectively, $D(x_i) \in \Re^3$ is the positive-definite symmetric inertia matrix, and $Q(x_i, \varphi_i) \in \Re^{3\times 3}$ is the Coriolis matrix.

B. Hybrid Automata-Based High-Level Plant

Hybrid systems are dynamic systems that combine CTS and DES. For example, the motor (a continuous component) of a mobile robot controlled by a microprocessor (a discrete component) is a hybrid system. Various hybrid systems are deployed in real applications (e.g., elevators, automobile engines, heating systems, and transportation systems). For this purpose, hybrid automatons have been proposed as formal models for hybrid systems [51]. The hybrid automaton $\mathcal{G}_h$ is a tuple comprising the following elements:

$$\mathcal{G}_h = (\mathcal{E}, \mathcal{X}, \Omega, \mathcal{U}, \mathcal{F}, \phi, \text{Inv}, \text{Guard}, \rho, \mathcal{E}_0, \mathcal{X}_0) \tag{7}$$

where $\mathcal{E}$ is the set of discrete states, $\mathcal{X}$ is the set of continuous states, Ω is the set of events, $\mathcal{U}$ is the set of admissible controls, $\mathcal{F}$ is the vector field of $\mathcal{G}_h$ ($\mathcal{F} : \mathcal{E} \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$), ϕ is the discrete state transition function of $\mathcal{G}_h$ ($\phi : \mathcal{E}\times\mathcal{X}\times\Omega \to \mathcal{E}$), Inv is the set defining an invariant condition (Inv $\subseteq \mathcal{E} \times \mathcal{X}$), Guard is the set defining a guard condition (Guard $\subseteq \Omega \times \mathcal{E} \times \mathcal{X}$), ρ is the reset function ($\rho : \Omega \times \mathcal{E} \times \mathcal{X} \to \mathcal{E} \times \mathcal{X}$), $\mathcal{E}_0$ is the initial discrete state, and $\mathcal{X}_0$ is the initial continuous state. In this study, we also attempted to model heterogeneous field robots using hybrid automata models that combine continuous-time models and discrete-event models (DEMs). Therefore, we extended the SCT to propose an HSHC structure applicable to complex dynamic systems. The detailed supervisor design for the hybrid systems is discussed in the following sections.

III. Hybrid Controller Design

A. CTS-Based Low-Level Control

We define the following distributed swarm control for the low-level control of heterogeneous field robots. Heterogeneous field robots consist of a group of N UAVs and M UGVs. We denote $\mathbf{r}_i \in \Re^3$ as the position of the ith robot, $i = 1, \dots, N+M$. Here, we define the position of the virtual point (VP) d_i to be followed by $\mathbf{r}_i$.

We define the dynamic undirected connectivity graph $\mathcal{C} := \{\mathcal{V}, \mathcal{K}\}$ by the vertex set $\mathcal{V} := \{1, 2, \dots, N+M\}$, representing the heterogeneous robots, and the edge set $\mathcal{K} := \{e_{ij} : i = 1, 2, \dots, N+M, j \in \mathcal{N}_i\}$ representing the connectivity among heterogeneous robots, and the dynamic neighbor set $\mathcal{N}_i$ of the ith robot is defined as follows:

$$\mathcal{N}_i := \{j \in \mathcal{V} : i \text{ receives information from } j, i \neq j\}. \tag{8}$$

Subsequently, the kinematic evolution of VP d_i generated by a distributed swarm control is as follows: for the ith robot

$$\dot{d}_i(t) := u_i^f + u_i^o + u_i^u, \qquad d_i(0) = \mathbf{r}_i(0) \tag{9}$$

where the three control inputs $u_i^u \in \Re^3$, $u_i^f \in \Re^3$, and $u_i^o \in \Re^3$ represent the velocity terms.

The first term $u_i^f \in \Re^3$ denotes a control input to avoid collisions among heterogeneous robots, preserves connectivity, and achieves the desired formation as specified by the desired distances $\mathcal{D}_f \in \Re^+$ $\forall i = 1, \dots, N+M$, and $\forall j \in \mathcal{N}_i$, as defined by

$$u_i^f := -\sum_{j\in\mathcal{N}_i} \frac{\partial \Phi_{ij}^f\left(\|d_i - d_j\|^2\right)^T}{\partial d_i} \tag{10}$$

where Φ_{ij}^f denotes a designed potential function that produces an attractive behavior if $\|d_i - d_j\| > \mathcal{D}_f$, a repulsive behavior if $\|d_i - d_j\| < \mathcal{D}_f$, and a null behavior if $\|d_i - d_j\| = \mathcal{D}_f$. There are two vertical asymptotes in $u_i^f(\Phi_{ij}^f)$ that correspond to the minimum and maximum permitted distances and, thus, avoidance of collisions between robots is guaranteed, and connectivity between robots is preserved.

The second term $u_i^o \in \Re^3$ is a control input based on a potential field that allows heterogeneous robots to avoid obstacles through a certain distance threshold $\mathcal{D}_o \in \Re^+$, expressed by the following equation:

$$u_i^o := -\sum_{j\in\mathcal{O}_i} \frac{\partial \Phi_{ij}^o\left(\left\|d_i - d_j^o\right\|\right)^T}{\partial d_i} \tag{11}$$

where $\mathcal{O}_i$ denotes the set of obstacles of the ith VP with an obstacle point d_j^o that corresponds to the position of the rth obstacle in the environment, and Φ_{ij}^o denotes a specific artificial potential function that produces a repulsive behavior if $\|d_i - d_j^o\| < \mathcal{D}_o$ and a null behavior if $\|d_i - d_r^j\| \geq \mathcal{D}_o$. When the distance between the VP and the obstacles becomes closer to D_o, the repulsive behavior increases to infinity. However, if $\|d_i - d_j^o\| \to \mathcal{D}_o$, then Φ_{ij}^o gradually converges to 0.

The final term $u_i^u \in \Re^3$ represents the desired velocity input of the VP, which is controlled by the planning algorithm defined as follows:

$$u_i^u = K_P(t)e_i(t) + K_I \int e_i(t)dt + K_D \frac{d}{dt} e_i(t) \tag{12}$$

where $\mathcal{T}_i \in \Re^3$ denotes the target velocity, $e_i(t) = \mathcal{T}_i - \dot{d}_i$ indicates the velocity error between the target and the VP, and K_P, K_I, and K_D are the parameters of the desired velocity controller.

B. Modular Supervisors-Based High-Level Control

The supervisor is defined as the automaton $\mathcal{S} = (X, \Omega, \delta, x_0, X_m)$, where X, Ω, δ, x_0, and X_m indicate the state set, event set, state transition function, initial state, and marker states, respectively. The plant is defined as hybrid automaton $\mathcal{G}$ and the behavior and generated language of the plant $\mathcal{G}$ under supervision are defined as

$$S/\mathcal{G} = \{X \times \mathcal{E}, \Omega, \delta \times \phi, (x_0, \mathcal{E}\mathcal{E}_0), X_m \times \mathcal{E}_m\} \tag{13}$$

$$\begin{aligned} L(S/\mathcal{G}) &: \varepsilon \in L(S/\mathcal{G}) \quad \forall s \in \Omega^*, \varepsilon \in \Omega \\ &s \in L(S/\mathcal{G}), s\upsilon \in L(\mathcal{G}), \upsilon \notin \mathcal{U} \Rightarrow s\upsilon \in L(S/\mathcal{G}) \end{aligned} \tag{14}$$

where υ is an eligible event and $\mathcal{U}$ is a control map defined as $\mathcal{U} : L(\mathcal{G}) \mapsto 2^{\Omega_c}$.

In addition, the projection map $P : \Omega^* \mapsto \Omega_o^*$ is defined as

$$\begin{aligned} P(\varepsilon) &= \varepsilon \\ P(\upsilon) &= \begin{cases} \varepsilon, & \text{if } \upsilon \notin \Omega_o \\ \upsilon, & \text{if } \upsilon \in \Omega_o \end{cases} \\ P(s\upsilon) &= P(s)P(\upsilon) \quad \forall s \in \Omega^* \quad \forall \upsilon \in \Omega. \end{aligned} \tag{15}$$

The controllability and observability of $L(S)$ with respect to $\mathcal{G}$ are defined as follows.

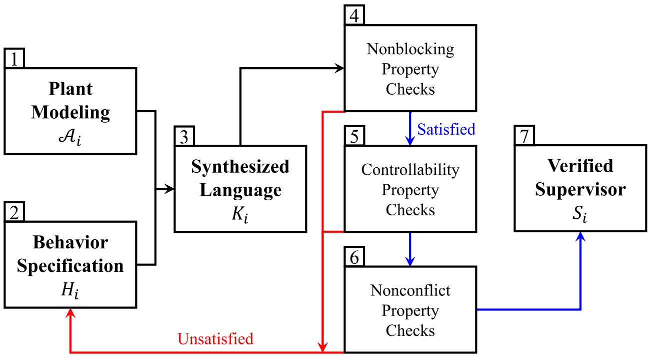

Fig. 2. Process of designing and verifying the supervisory controllers $\mathcal{S}_i$.

Definition 1: $\mathcal{S}$ is defined as *controllable* with respect to $(\mathcal{G}, \Omega_{uc})$ when the following condition is satisfied:

$$(\forall s, \upsilon) s \in \overline{L(S)},\ \upsilon \in \Omega_{uc},\ s\upsilon \in L(\mathcal{G}) \Rightarrow s\upsilon \in \overline{L(S)}. \quad (16)$$

In other words, s, which is allowable by $\mathcal{S}$ and an uncontrollable event υ, is eligible in $\mathcal{G}$ if the sequence $s\upsilon$ is eligible in $\mathcal{G}$; furthermore, if $\mathcal{S}$ also allows $s\upsilon$, then $\mathcal{S}$ is controllable with respect to $\mathcal{G}$.

Definition 2: For $\mathcal{S} \subset \mathcal{G}$, $\mathcal{S}$ is said to be *observable* with respect to $(\mathcal{G}, P, \Omega_{uo})$ when the following condition is satisfied:

$$\left(\forall s, s', \upsilon \in \overline{L(\mathcal{S})}\right) \upsilon \in \Omega_{uo}, P(s) = P\left(s'\right)$$
$$s\upsilon \in \overline{L(\mathcal{S})}, s'\upsilon \in L(\mathcal{G}) \Rightarrow s'\upsilon \in \overline{L(\mathcal{S})}. \quad (17)$$

In other words, if the string $s\upsilon$ is permissible by the supervisor S and $s'\upsilon$ is eligible in the plant $\mathcal{G}$, then $\mathcal{S}$ also has to permit $s'\upsilon$, where two strings $s, s' \in \Omega^*$ are recognized as the same string by the projection map P and are also permissible by $\mathcal{S}$. When υ is an unobservable event, then $\mathcal{S}$ is observable with respect to $\mathcal{G}$. Note that we consider only controllability for the DES-based high-level control design in this study.

Moreover, the supervisory control problem (SCP) used to design the supervisor (the procedure presented in Fig. 2) is defined as follows.

Definition 3: For a given $K \subseteq \mathcal{G}$, we determine a supremal language $\mathcal{S}$ that is controllable with respect to $(\mathcal{G}, \Omega_{uc})$, satisfying $L(S/\mathcal{G}) = K$ and $L(S/\mathcal{G}) = \overline{L_m(S/\mathcal{G})}$.

Therefore, if K is defined as the specification for $\mathcal{G}$, then the SCP has to identify a supervisory controller that satisfies $L(S/\mathcal{G}) = K = \overline{L_m(S/\mathcal{G})}$ and is nonblocking and controllable with respect to $\mathcal{G}$. Here, a plurality of supervisors that satisfy the specifications exists and is controllable. Among these K, a supremal controllable sublanguage of K is determined as the solution of the SCP. Therefore, $\mathcal{S}$ can maximally allow the eligible language to occur in $\mathcal{G}$. The application of the proposed hybrid system modeling and controller design for heterogeneous field robot teams is described in the following sections.

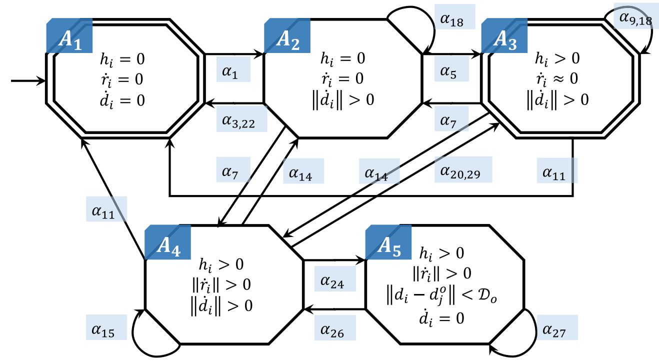

Fig. 3. Hybrid automata model for UAV $\mathcal{G}_{\mathcal{A}}$.

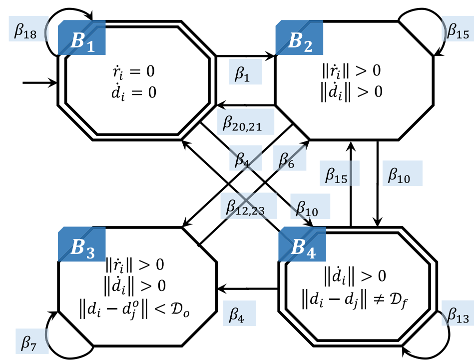

Fig. 4. Hybrid automata model for UGV $\mathcal{G}_{\mathcal{B}}$.

IV. Hybrid System-Based Heterogeneous Field Robots

A. Plant Modeling of Heterogeneous Field Robots

The objective is to enable collaboration among heterogeneous field robots so that a hybrid automaton model and behavior specifications of heterogeneous field robots consisting of UAVs and UGVs can be designed. The hybrid models for UAVs and UGVs are shown in Figs. 3 and 4, respectively. In these figures, the initial state is shown first (i.e., at the upper left position in the figure). The marked states are indicated by a double line, the discrete states are listed at the top left of each shape, the continuous states are described in the middle of the shape, and the events and state transition functions are specified with alphabetical designations and arrows, respectively. Here, we modeled the states of hybrid models by focusing on the dynamics of heterogeneous robots with a CTS, unlike in previous studies [20]. In other words, hybrid modeling was considered for heterogeneous field robots by combining the CTS-based modeling and the DES-based modeling introduced in Section II. As a result, the UAV model $(\mathcal{G}_{\mathcal{A}})$ consists of five states, 23 transitions, and 15 events, whereas the UGV model $(\mathcal{G}_{\mathcal{B}})$ consists of four states, 13 transitions, and 12 events. The events for each model are described in detail in Table I. Here, odd-numbered events are controllable events, and even-numbered events are uncontrollable events. Finally, we obtained the hybrid automata model of the entire plant system (i.e., the heterogeneous field robot team) through

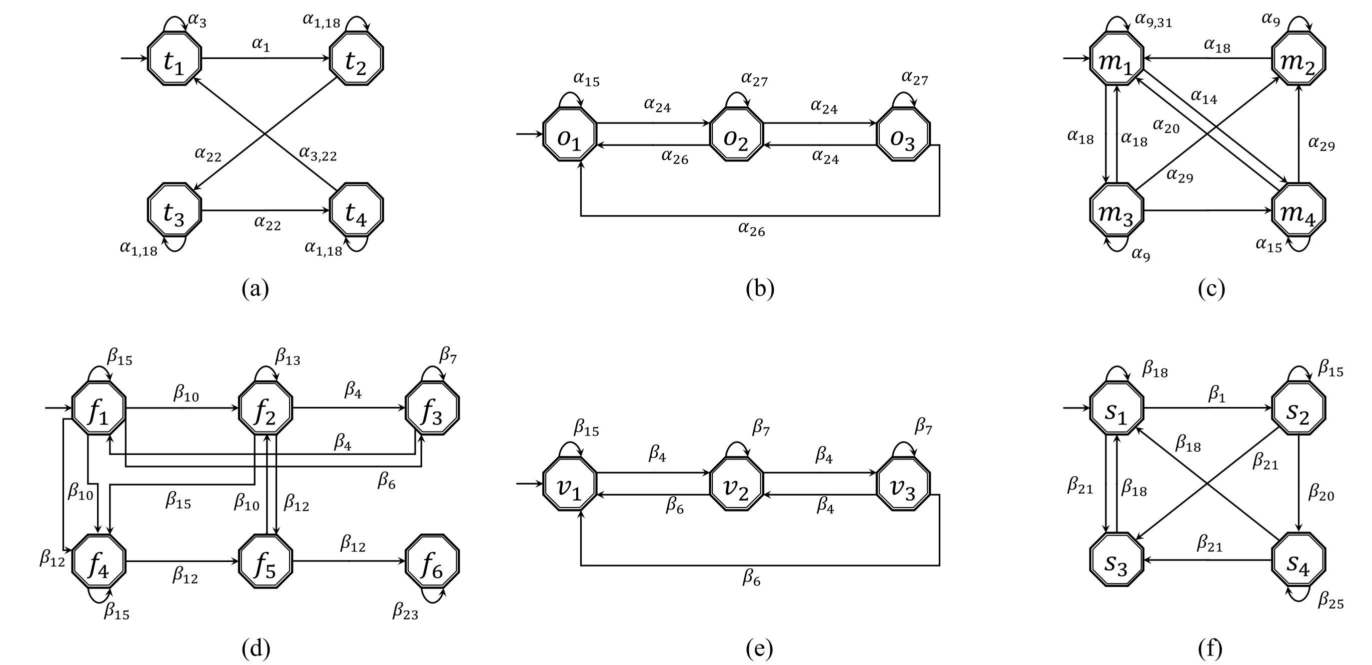

Fig. 5. Specifications for heterogeneous robots: (a) navigation for UAV H_A^1, (b) obstacle avoidance for UAV H_A^2, (c) mission management for UAV H_A^3, (d) navigation for UGV H_B^1, (e) obstacle avoidance for UGV H_B^2, and (f) mission management for UGV H_B^3.

TABLE I
DESCRIPTION OF EVENTS FOR HYBRID UAV MODEL $\mathcal{G}_\mathcal{A}$ AND HYBRID UGV MODEL $\mathcal{G}_\mathcal{B}$

Plant	Event	Controllability	Description
UAV ($\mathcal{G}_\mathcal{A}$)	α_1	O	Arm
	α_3	O	Disarm
	α_5	O	Take off
	α_7	O	Land
	α_9	O	Keep hovering
	α_{11}	O	Return home
	α_{14}	X	Start mission
	α_{15}	O	Keep mission
	α_{18}	X	Receive mission
	α_{20}	X	Finish mission
	α_{22}	X	Time out
	α_{24}	X	Detect obstacles
	α_{26}	X	Detect free space
	α_{27}	O	Keep avoiding
	α_{29}	X	Clear mission
	α_{31}	O	Keep waiting formation
UGV ($\mathcal{G}_\mathcal{B}$)	β_1	O	Start mission
	β_4	X	Detect obstacles
	β_6	X	Detect-free space
	β_7	O	Keep avoiding
	β_{10}	X	Network connected
	β_{12}	X	Network disconnected
	β_{13}	O	Keep formation
	β_{15}	O	Keep mission
	β_{18}	X	Receive mission
	β_{20}	X	Finish mission
	β_{21}	O	Clear mission
	β_{23}	O	Break formation
	β_{25}	O	Keep waiting formation

parallel composition [13] as follows:

$$\mathcal{G}_{\text{plant}} = \mathcal{G}_\mathcal{A}||{\mathcal{G}_\mathcal{B}}^1||{\mathcal{G}_\mathcal{B}}^2||{\mathcal{G}_\mathcal{B}}^3. \tag{18}$$

1) Hybrid Automata Model for UAVs: The hybrid automata $\mathcal{G}_\mathcal{A}$ (shown in Fig. 3) models the dynamic behavior of the UAV, and the states are as follows: $\mathcal{E}_\mathcal{A} = \{A_1, A_2, A_3, A_4, A_5\}$, where A_1: *Ideal*, A_2: *Arming*, A_3: *Hovering*, A_4: *Flying*, and A_5: *Avoiding*. The eligible events Ω_A are listed in Table I, and the discrete transition function is $\phi_A : f(\mathcal{E}_\mathcal{A}, \Omega_A) = \mathcal{E}_\mathcal{A}$, as indicated by the arrow in Fig. 3. The initial state $\mathcal{E}_0$ is A_1, and the desired state is $\mathcal{E}_m = \{A_1, A_3\}$. From the perspective of the CTS, h_i represents the height of the ith robot, $\mathbf{r}_i$ is the position, and d_i is the position of the ith VP. For example, if an event α_5 occurs in state A_2, the state transitions to A_3, where the continuous state $\mathcal{X}_\mathcal{A}$ is a state in which the height h_i of the UAVs is greater than 0, the velocity $\dot{\mathbf{r}}_i$ is close to 0, and a control command is input to the VP ($||d_i|| > 0$). The initial continuous state $\mathcal{X}_0$ of this model is as follows: $h_i, \dot{\mathbf{r}}_i, \dot{d}_i = 0$.

2) Hybrid Automata Model for UGVs: Hybrid automata $\mathcal{G}_\mathcal{B}$ (shown in Fig. 4) model the dynamic properties of the UGV, and the states are as follows: $\mathcal{E}_\mathcal{B} = \{B_1, B_2, B_3, B_4\}$, where B_1 is *stationary*, B_2: *navigation*, B_3: *Safety*, and B_4: *formation*. The discrete transition function is $\phi_B : f(\mathcal{E}_\mathcal{B}, \Omega_B) = \mathcal{E}_\mathcal{B}$, as indicated by the arrows in Fig. 3, the initial state $\mathcal{E}_0$ is B_1, and the desired state is $\mathcal{E}_m = \{B_1, B_4\}$. For example, when UGVs form the desired formation (i.e., the state is B_4) and an eligible event β_4 occurs that detects an obstacle, the state transitions to B_3. In the B_3 state, the event of avoiding obstacles (β_7) is allowed by the supervisory controller $\mathcal{S}_i$. Here, in the CTS, the velocity of the UGV ($||\dot{\mathbf{r}}_i||$) and the control input to the VP ($||\dot{d}_i||$) are greater than zero, and the distance between the UGV and the obstacle ($||d_i - d_j^o||$) is less than the desired safe distance ($\mathcal{D}_o$).

B. Design of Behavioral Policies

The supervisor is the result of establishing a legal language (i.e., specification, as shown in Fig. 5) and then finding a supremal controllable sublanguage for the plant $\mathcal{G}_{\text{plant}}$. An alternate description is that we find a controllable language that meets the conditions for becoming an appropriate modular supervisor $\mathcal{S}_i$ with respect to the plant model $\mathcal{G}_i$. The behavior specifications are designed such that they satisfy the conditions of nonblocking, controllability, and nonconflict for the cooperation of heterogeneous field robots. The $\text{POL}_1, \text{POL}_2, \ldots, \text{POL}_{10}$ policies for the design specifications are as follows.

1) *POL_1:* Arming must be in operation before the UAV takes off (i.e., hovering).
2) *POL_2:* The mission is assigned while the UAV is arming and performed while the UAV is hovering.
3) *POL_3:* When a UGV completes its mission, the UAV lands and waits for the next mission.
4) *POL_4:* UGVs move into a desired formation when the network is connected.
5) *POL_5:* UGVs receive a mission while the UGV is stopped.
6) *POL_6:* UGVs perform missions that involve moving into formations.
7) *POL_7:* At the end of the mission, the UAV remains hovering and the UGV remains stopped.
8) *POL_8:* If an obstacle is found while the robot is moving, obstacle avoidance is the priority and is guaranteed.
9) *POL_9:* Duplicate missions are not assigned to the robots.
10) *POL_{10}:* When the mission is over, the robots form a desired formation and wait for the next mission.

We modeled the specifications H_i that met the aforementioned policies before synthesizing the modular supervisory controller $\mathcal{S}_i$. Six specifications were designed to achieve the desired control objectives for each robot (Fig. 5). Fig. 5(a)–(c) are the control specifications for the UAV, modeling for arming and disarming, obstacle avoidance, and mission management, respectively. In addition, Fig. 5(d)–(f) are control specifications for navigation, obstacle avoidance, and mission management for UGVs. A detailed description is provided below. We built the automaton H_A^1, shown in Fig. 5(a), which considers policies POL_1, POL_2, and POL_7. The states of H_A^1 are as follows: 1) state t_1 represents the UAV in a stopped state; 2) state t_2 represents the arming state in which the UAV moves and is assigned a desired goal; 3) the UAV travels with a given mission in t_3; and 4) state t_4 shows the situation in which the UAV avoids obstacles. We also built an automaton H_B^1, as shown in Fig. 5(d), using the specifications of the UGV, considering POL_3 to POL_7. Specifications H_A^2 and H_B^2 considering POL_8 are shown in Fig. 5(b) and (e). In other words, if obstacles are found in states o_1 and v_1 while the robot is in motion, the states will be transitioned to states o_2 and v_2, respectively. The obstacle avoidance controls, controllable events, α_{27} and β_7, are allowed in those states. When heterogeneous robots avoid obstacles, the state transitions from o_2 to o_3, and v_2 to v_3 are repeated until free-space detection events α_{26} and β_6 occur. Finally, specifications H_A^3 and H_B^3 reflecting POL_9 and POL_{10} are depicted in Fig. 5(c) and (f). In the case of UGVs, state s_1 that receives the mission transitions to state s_2 when the mission begins. In state s_2, the mission is performed according to the controllable event β_{15}. Here, state s_2 transitions to state s_4 when the mission is completed, and transitions to state s_3 when the mission is cleared. In states s_3 and s_4, when a new mission is received, the state transitions to s_1 and iterates through these mission loops. When the desired waiting formation is input to the control system, the heterogeneous robot maintains that formation until it receives a new mission.

Therefore, the control objectives model which events are allowed and disallowed correctly by the supervisor when heterogeneous robots avoid collision with each other when given a mission. In addition, when an obstacle is found, it assigns the highest priority; in the case of UGVs, it is designed to maintain the formation while following the given path. The design specifications $\mathcal{H}_i$ were modeled to reflect the desired policies and meet the conditions for a proper formal language.

C. Hybrid Controller of Heterogeneous Field Robots

For modular supervisory control, plant $\mathcal{G}_i$ and specification H_i are synthesized to obtain the legal language K_i. If $\mathcal{S}_i,\ i = 1, 2, \ldots, m$, is verified for the legal specifications $K_i,\ i = 1, 2, \ldots, n$ $(m \leq n)$, then $\mathcal{S}_i$ is defined as a modular supervisory controller. The modular supervisor $\mathcal{S}_i$ satisfies the nonconflict condition defined as

$$\bar{S} = \bar{S_1} \wedge \bar{S_2} \wedge \cdots \wedge \bar{S_m} \tag{19}$$

where $\wedge$ is the *meet* [13] [defined as $S_1 \wedge S_2 = L(S_1) \bigcap L(S_2)$]. In (17), S is a centralized supervisor designed with respect to $K = K_1 \wedge \cdots \wedge K_n$. If $\mathcal{S}_i$ satisfies the nonconflict condition, all modular supervisors can control plant $\mathcal{G}_{\text{plant}}$ in the same way as the centralized supervisor $\mathcal{S}$. If $\mathcal{S}_i$ is conflicting, modular supervisors do not satisfy the specifications $K_i,\ i = 1, 2, \ldots, n$, in that they allow unnecessary events. Therefore, the modular supervisory $\mathcal{S}_i$ must satisfy the conditions of controllability, nonblocking, and nonconflict, as explained in Section III. The solution of the modular supervisory SCP is presented in Algorithm 1 (see [14] for a formal proof).

The subcomponent $\mathcal{G}_{\text{sub},i}$ of the entire plant $\mathcal{G}_{\text{plant}}$ consists of UAV $\mathcal{G}_{\mathcal{A}}$ and UGV $\mathcal{G}_{\mathcal{B}}$. First, we check whether each legal language K_i satisfies the controllability, nonblocking, and nonconflict conditions for the plant. According to Algorithm 1, the legal languages K_i that satisfy the requirements for subplant UAVs $\mathcal{G}_{\mathcal{A}}$ and UGV $\mathcal{G}_{\mathcal{B}}$, $K_{\mathcal{A}}^3$, $K_{\mathcal{B}}^1$, and $K_{\mathcal{B}}^2$, are defined as modular supervisors $\mathcal{S}_{\mathcal{A}}^3$, $\mathcal{S}_{\mathcal{B}}^1$, and $\mathcal{S}_{\mathcal{B}}^2$, respectively. In contrast, the supremal controllable sublanguage $\mathcal{S}_i$ was synthesized for the legal specifications K_i that do not satisfy controllability, that is, $K_{\mathcal{A}}^1$, $K_{\mathcal{A}}^2$, and $K_{\mathcal{B}}^3$. The obtained supremal controllable sublanguage $\mathcal{S}_i$ is the solution to the SCP for K_i and G_i; hereafter, it was determined to be supervisors $\mathcal{S}_{\mathcal{A}}^1$, $\mathcal{S}_{\mathcal{A}}^2$, and $\mathcal{S}_{\mathcal{B}}^3$, respectively, as shown in Figs. 6 and 7, respectively. As a result, we found a modular supervisor $\mathcal{S}_i,\ i = 1, 2, \ldots, 6$ for specification language $K_i,\ i = 1, 2, \ldots, 6$ with respect to $\mathcal{G}_i$. Finally, we verified that the designed modular supervisory controllers satisfy nonconflict by using a supervisory control synthesis tool. These modular supervisors $\mathcal{S}_i$ can control the dynamic complex system in the desired behavior by selectively allowing or disallowing controllable events Ω_c generated by the plant $\mathcal{G}_{\text{plant}}$.

Algorithm 1 Solutions of Modular Supervisory SCP

Input: Plant $\mathcal{G}_i$ and specification $\mathcal{H}_i$, $i = 1, 2, \ldots, n$.
Output: Modular supervisors $\mathcal{S}_j$ or $\mathcal{S}'_j$, $j = 1, 2, \ldots, m$.

- Step 1: Define the subplant automaton $\mathcal{G}_{sub,i}$ and specification automaton $\mathcal{H}_i$.
- Step 2: Synthesize the subplant automaton $\mathcal{G}_{sub,i}$ and specification automaton $\mathcal{H}_i$ for the legal language K_i
- Step 2: Determine $\mathcal{S}_j$ (the supremal controllable sublanguage) of K_i and $\mathcal{G}_{sub,i}$.
- Step 3: Verify that K_i controllable with respect to $\mathcal{G}_{sub,i}$ using Ω_c in K_i.
- Step 4: Verify that K_i is nonblocking.
- Step 5: Verify that K_i is nonconflicting with respect to $\mathcal{G}_{sub,i}$. If K_i is nonconflicting, then $\mathcal{S}'_j$ exists, where $\mathcal{S}'_j = K_i$. Go to Step 8.
- Step 6: If the specification K_i does not satisfy Steps 3, 4, and 5, go back to Step 1 and redesign. Nevertheless, if K_i does not meet the proper conditions, then $\mathcal{S}'_j$ does not exist. Go to Step 7.
- Step 7: $\mathcal{S}_j$ is the solution of the SCP with respect to $(K_i, G_{sub,i})$. **return** $\mathcal{S}_j$
- Step 8: If $L(\mathcal{S}_j) = L(\mathcal{S}'_j/G_{sub,i})$, then $\mathcal{S}'_j$ is the solution of the SCP with respect to $(K_i, G_{sub,i})$.
  return $\mathcal{S}'_j$, **else return** $\mathcal{S}_j$.

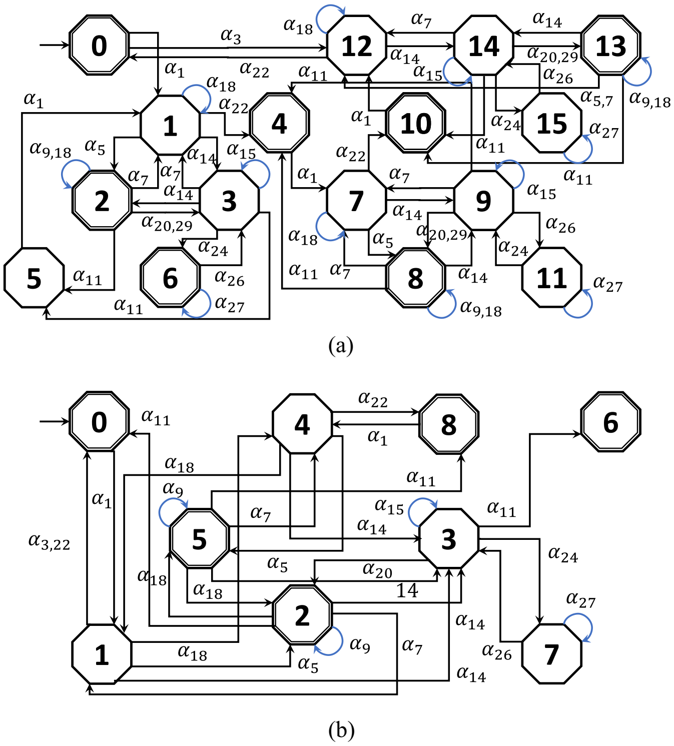

Fig. 6. UAV modular supervisors: (a) supervisor 1 ($\mathcal{S}^1_{\mathcal{A}}$) and (b) supervisor 3 ($\mathcal{S}^3_{\mathcal{A}}$).

V. Experimental Results and Discussions

A. Verification Through Dynamic Simulations

1) Environmental Setup: A physics-based simulator was constructed to implement and verify the designed modular

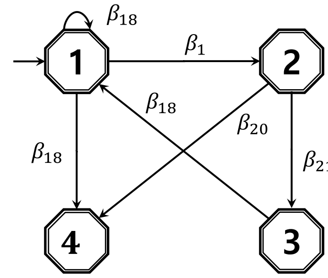

Fig. 7. UGV modular supervisor ($\mathcal{S}^3_{\mathcal{B}}$).

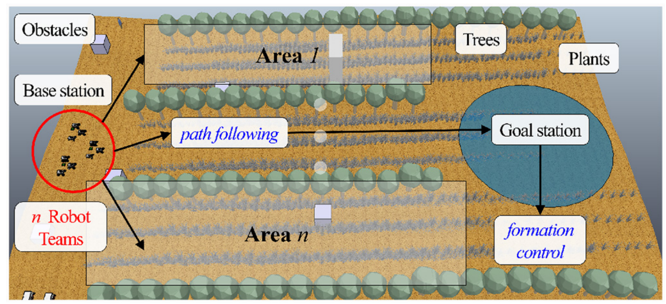

Fig. 8. Simulation environments for HSHC-based heterogeneous field robot teams.

supervisory controller and HSHC architecture for controlling heterogeneous field robots, as shown in Fig. 8. The heterogeneous robot system included multiple UAVs and UGVs. The virtual environment is described as an agricultural field. Each robot was equipped with LIDAR and camera sensors to recognize obstacles and assign a desired path to the starting and target points through a start algorithm. The entire system was implemented using V-REP, a robot simulator, and MATLAB for low-level and high-level controllers. For the hybrid system, a DES and SCT computation tool, TCT, was combined with CST-based MATLAB to send and receive real-time data to and from the simulator. The lower controller and the modular supervisor communicated with each other through the DES-to-CTS and CTS-to-DES interfaces, including the control logic channel and the information channel, and sent or received command signals through the mapping function. To evaluate the performance of the proposed HSHC system, the measured output was recorded at 50 Hz. Furthermore, to analyze the experimental results, the states of the plant, the occurrence of events, the position and velocity of the heterogeneous robot, and the VP were recorded.

2) Experimental Task: In these experiments, the heterogeneous robots were required to carry out collaborative mapping and driving in an unknown field. The experimental task executed was as follows: a UAV first navigated in the agricultural environment and then shared the constructed map with the UGVs. The UGVs then drove based on the shared map for autonomous navigation and obstacle avoidance. We set up cubes and trees as obstacles, and the heterogeneous robots mapped and drove using onboard sensors and controllers. The following four experiments were conducted by extending the number of robots in an agricultural environment with obstacles.

Case 1: One team (1 UAV and 3 UGVs).

Case 2: Three teams (3 UAVs and 12 UGVs).

Case 3: One team (Occurrence of network disconnected and connected events).

Case 4: Three teams (Occurrence of keep waiting formation events).

For case 2, the extended plant is computed by simply synthesizing the existing plants as follows:

$$\mathcal{G}_{\text{new}} = \mathcal{G}_{\text{plant}} || \mathcal{G}_{\text{plant}} || \mathcal{G}_{\text{plant}}. \tag{20}$$

In terms of implementation, we synthesized the plant and supervisory controller using the TCT function as follows:

1) $\mathbf{G}_{\text{new}} = \mathbf{sync}(\mathbf{G}, \mathbf{G}, \mathbf{G})$;
2) $\mathbf{SUP}_{\text{new}} = \mathbf{supcon}(\mathbf{G}_{\text{new}}, \mathbf{SPEC})$

where **G** and **SPEC** are the automata models of the plant and the specifications computed in case 1, respectively. In addition, **sync** and **supcon** are functions used for synthesis in the TCT. Therefore, the model and high-level controller of the heterogeneous robot plant were computed promptly. To check the disabled events, the control action of $\mathbf{SUP}_{\text{new}}$ is determined by the **condat** function as follows:

$$\mathbf{SUP}.\text{DAT}_{\text{new}} = \mathbf{condat}(\mathbf{G}_{\text{new}}, \mathbf{SUP}_{\text{new}}). \tag{21}$$

In this study, we input the desired waiting formation only for experimental case 4 to verify the scalability of the proposed control architecture. The desired formation was set by dividing the shape of the goal station by the number of robots. In this experimental task, we focused on whether heterogeneous robots performed a given cooperative task while achieving the control objectives of the modular supervisors. In addition, we confirmed that the hybrid system satisfied the given behavioral specifications even as the number of robots increased.

3) Simulation Results: The experimental results of the agricultural scenarios for cases 1 and 2 are shown in Figs. 9 and 10, respectively. Figs. 9 and 10(a)–(c) show the path, state transitions, and observed events of the hybrid systems consisting of UAVs and UGVs controlled by the designed modular supervisors.

First, Fig. 9(b) shows that the state of the UAV transitions to A_2, A_3, and A_4 by the occurrence of events, such as α_1, α_{18}, α_5, α_9, α_{14}, and α_{15}. The UAV performs the given mission in state A_4 and hovers in A_3. In Fig. 9(c), UGVs also change state from B_1 to B_4 and B_2 through permissible events, such as β_{10}, β_{13}, β_{18}, β_1, and β_{15}. The UGVs maintain the desired formation in state B_4 and perform the mission by changing state to B_2 between 100 and 150 s after the mission of the UAV is over. Because the mission ends in 300 s, the states of the UAV and UGVs become the ideal states of A_1 and B_4, respectively. Similarly, the experimental results for case 2 are presented in Fig. 10(b) and (c). Here, the UAV missions are set to perform at 15 s intervals. As shown in Fig. 10(b), the states of the UAVs transition from A_1 to A_4 at constant intervals. In addition, because the length of each given path is different, the figures show that the duration time of the UAV state A_4 and the UGV state B_4 are different. Furthermore, the mission time performed by the UGVs is also different for each team, and the result can be confirmed through state B_2 in Fig. 10(c). Fig. 10(b) and (c) show that the UAVs and UGVs have very similar state transitions for each team. In case 2, an obstacle was observed by the third team

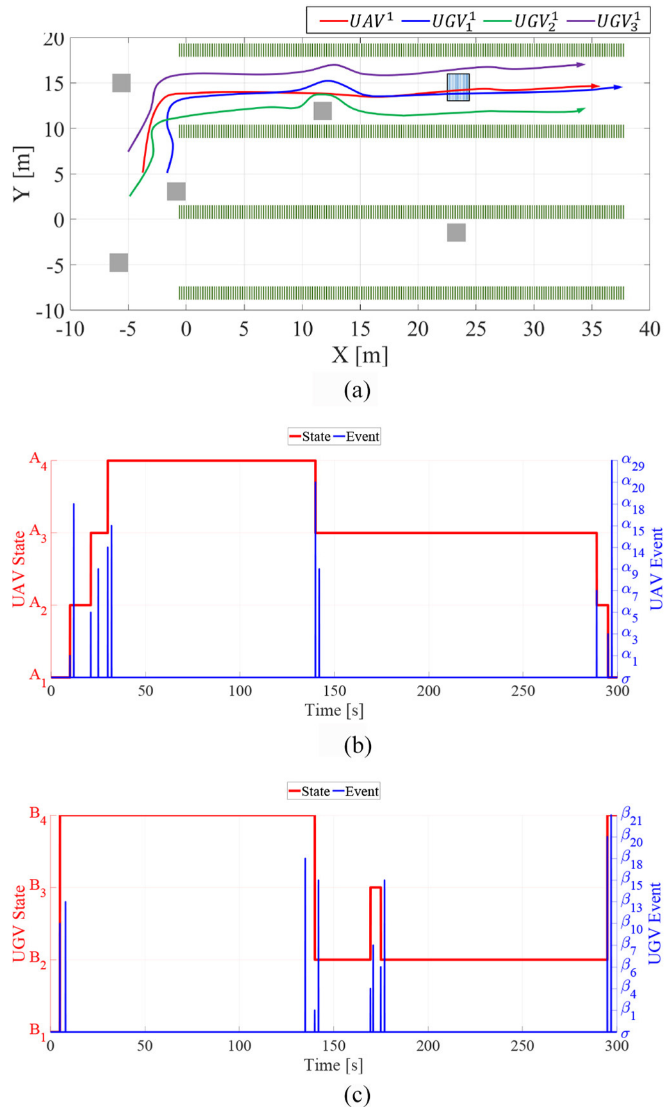

Fig. 9. Experimental results for case 1: (a) path, (b) UAV, and (c) UGVs.

between 200 and 250 s. Therefore, Fig. 10(c) shows that the state of the UGVs State[3] transitioned to B_3. Finally, Figs. 9(a) and 10(a) show the path of the heterogeneous field robots. Through these results, the hybrid systems consisting of the large-scale dynamic robots are seen to perform the given mission, avoid obstacles, and achieve the desired formation under the modular supervisor-based hierarchical control architecture.

Fig. 11 shows the results of experimental case 3. Fig. 11(b) represents the average relative distance between the UGVs during the cooperative task. The jamming obstacle in the new environment causes the occurrence of network disconnection and reconnection. While the communication network is disconnected, it can be seen that the control objective of forming the desired formation (i.e., formation state B_4) is not achieved and that UGVs are simply traveling on an allocated path (i.e., navigation state B_2).

Furthermore, the path of the heterogeneous field robots for experimental case 4 is shown in Fig. 12. This experiment was designed to prove that the behavior specification of POL_{10} is satisfied by the modular supervisors $\mathcal{S}_i$. Through

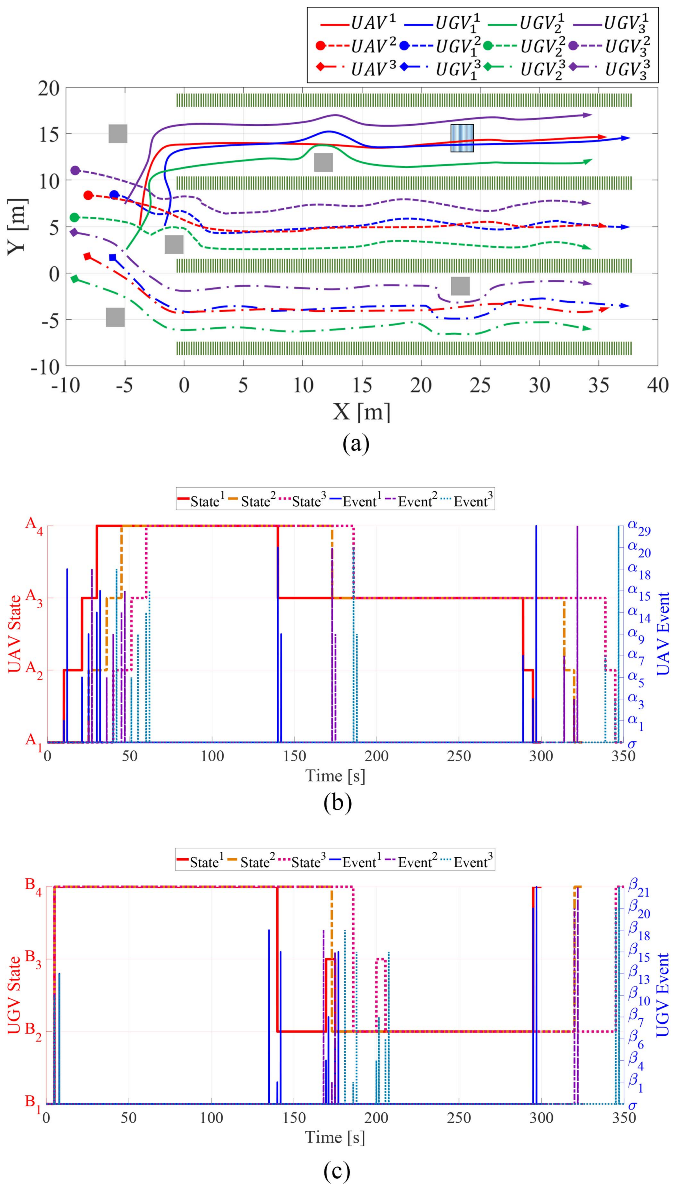

Fig. 10. Experimental results for case 2: (a) path, (b) UAV, and (c) UGV.

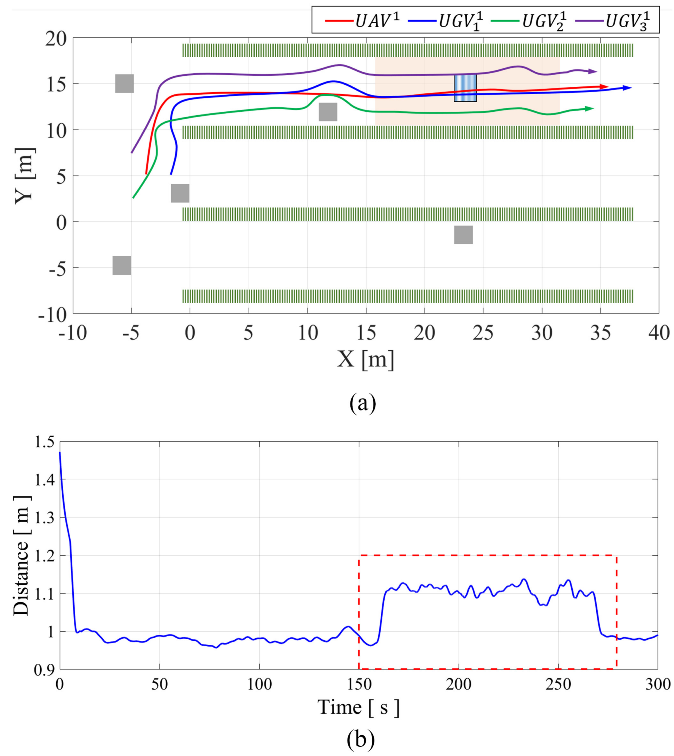

Fig. 11. Experimental results for case 3: (a) path and (b) relative distance of UGVs.

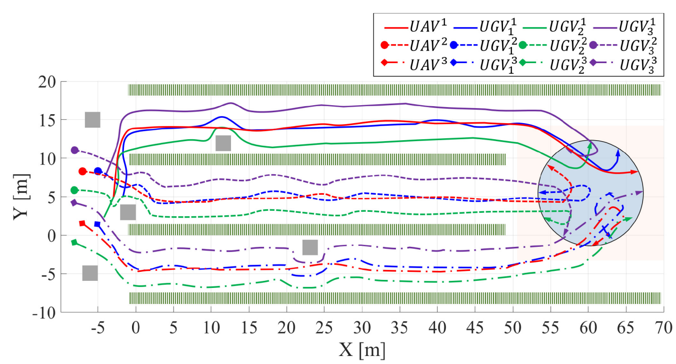

Fig. 12. Experimental result for case 4: path.

the systematic results, we found that events α_{31} and β_{25} were allowed by modular supervisors $\mathcal{S}_{\mathcal{A}}{}^3$ and $\mathcal{S}_{\mathcal{B}}{}^3$, respectively. The experimental results show that the hybrid systems composed of UAVs and UGVs controlled by the proposed controllers form the desired waiting formation. As a result, we proved that hybrid systems consisting of large-scale dynamic robots can perform the given mission, avoid obstacles, and achieve the desired formation under the modular supervisor-based hierarchical control architecture.

B. Field Tests for HSHC of Heterogeneous Field Robots

We used the UAV (3DR Solo) and UGVs (Jackal and Husky robots) to test the proposed HSHC-architecture-based heterogeneous field robot teams in a real environment. Each robot was equipped with an onboard computer (Nvidia Jetson TX2 or AGX Xavier) for computing, and a laptop was used for high-level control and logging. In addition, a robot operating system (ROS) was used to communicate between the heterogeneous field robots based on the user datagram protocol. Field tests were conducted so that heterogeneous robots performed tasks designed in the simulation in an open field. Fig. 13 represents the field robots, path, state transition of the UAV and the UGVs, and event occurrence, and we can clearly identify how the state of the heterogeneous field robots changed during the field tests. For the field tests, we only considered obstacle-free conditions as shown in Fig. 13(a).

We can observe the occurrence of an event in Fig. 13(d), and the states of the UAV and UGVs [Fig. 13(c)] based on these event strings. For example, due to the occurrence of the events of arm α_1, take off α_5, and keep hovering α_9, the UAV state sequentially transitions from A_1 to A_2 and A_3. Subsequently, the UAV flies the given path through the events of the start mission α_{14} and keep mission α_{15} with the mission received by the event of α_{18}. In the case of the UGVs, keep formation

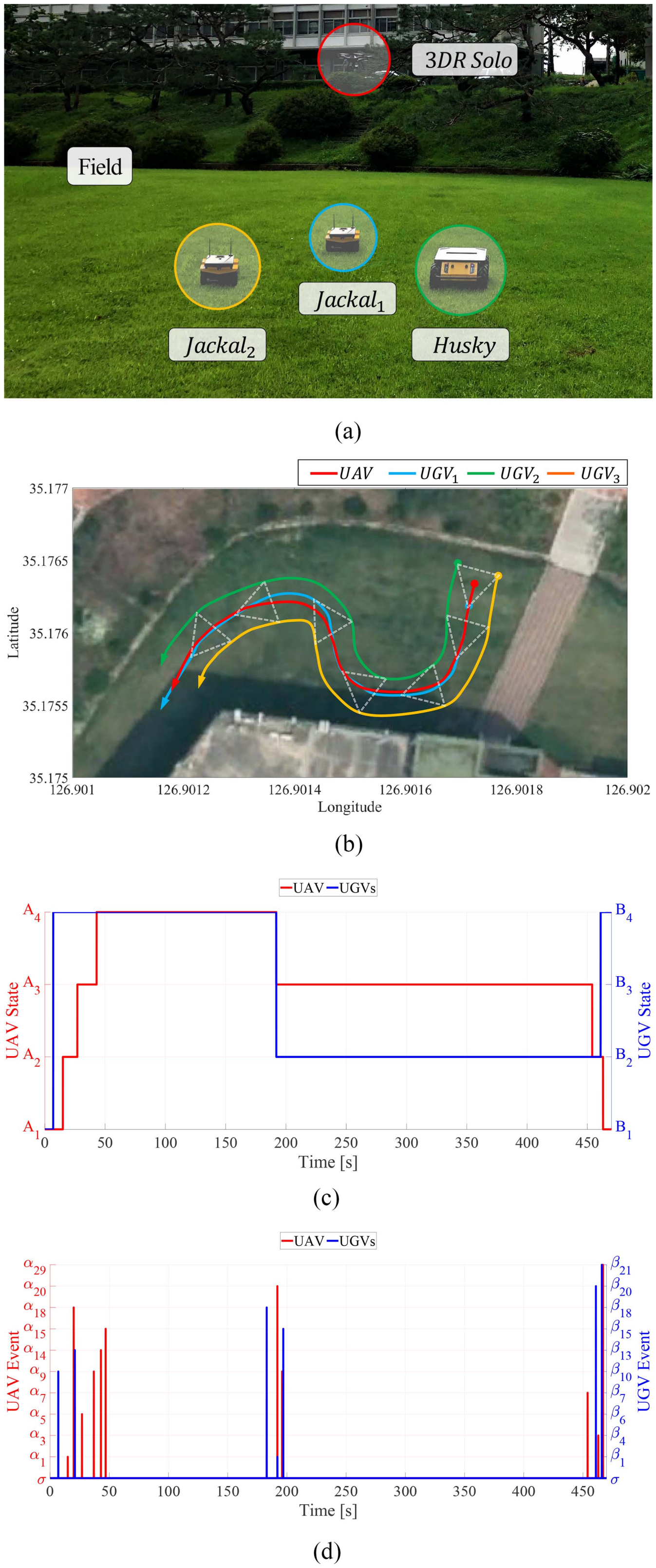

Fig. 13. Field tests and results for HSHC-based heterogeneous field robot teams consisting of UAV and UGVs: (a) system setup, (b) path, (c) state transition, and (d) event occurrence.

β_{13} is allowed by the modular supervisory controller when an event of network connected β_{10} is observed. Therefore, UGVs can achieve their mission while forming the desired formation in state B_2. Finally, the path of the heterogeneous field robot teams controlled by the modular supervisory controller based on the hierarchical control architecture is shown in Fig. 13(b).

In this way, the designed, proposed, and implemented HSHC architecture can effectively control heterogeneous field robot teams and has the advantage of systematic analysis. It is possible to intuitively identify in detail not only the continuous outcomes of the state variables, but also the discrete events and the discrete state of the hybrid systems. On the other hand, in the conventional control system (i.e., continuous-time domain) of previous studies [7], it is impossible to perceive systematic results, such as state transitions and event occurrences. This HSHC architecture is effective for managing and operating large dynamic systems, such as heterogeneous field robot teams. Event-driven systems have been developed and introduced in the real world for advanced automation, such as human-made systems or cyber–physical systems (e.g., elevators, transportation systems, and production/c systems). Therefore, if the proposed HSHC is also introduced into heterogeneous multirobot systems for agriculture and field use, it will be more effective, practical, scalable, and feasible.

C. Discussion

This article presents an HSHC for heterogeneous field robot teams, together with extensive experimental results in both virtual and real environments. The proposed HSHC architecture was based on the authors' earlier studies [21], [30]. First, we studied low-level-based system modeling and controller design to control multirobots [7]. However, we aimed to control more complex dynamic systems, such as heterogeneous multirobot systems. For this purpose, we considered the following points.

1) *Effectiveness:* Can we intuitively control complex dynamic systems according to the desired control objectives?
2) *Scalability:* How easily can we cope with changes in the dynamic system model?
3) *Feasibility:* How feasible is the control system?
4) *Systematic Approach*: Can we systematically observe the behavior of the system and controller?

To achieve this, we determined that a high-level-based control system is required to address the various challenges mentioned above. Therefore, we focused on discrete-event dynamics with SCT and attempted to apply these frameworks to robotics [21], [30]. Nevertheless, the DES had limitations in terms of practicality and feasibility since it has been addressed only in the software domain. Moreover, because continuous dynamics are not considered in defining the robot state, there is a crucial disadvantage in the design of the plant model and the supervisory controller. Therefore, we developed a hybrid system by combining DESs, including discrete-event dynamics, with the conventional CTS. These hybrid systems require a supervisory controller in the high-level domain to handle event-driven features. We developed a hierarchical control architecture based on a hybrid system by designing a modular supervisor suitable for handling complex dynamic systems in the swarm control algorithm developed in the author's previous study [7].

We designed heterogeneous field robots through a hybrid automaton that combines continuous state-based dynamics and discrete-event-based dynamics. In addition, we presented its *effectiveness* by modeling the legal language (i.e., behavior specifications). We therefore established the modeling criteria for the hybrid robot plants considering the continuous dynamics and the specification reflecting control policies. We also presented simulation results that allow the developed system to control the extended dynamic system where the heterogeneous field robot teams consisted of 12 agents with UAVs and UGVs. Even if the model of the complex dynamic system changed, we could simply synthesize the entire plant model and the supervisory controller while ensuring *scalability*. Our proposed HSHC approach, therefore, can be considered as the first step toward multiobjective and multilayer systems for controlling field robots.

We implemented the proposed system using an actual field robot and verified its *feasibility* through field tests. Based on the results of the field tests, the proposed hierarchical control-based hybrid system has sufficient *practicality* and *heterogeneity*. This means that the reliability of SCT, which was limited to specific environments and simulations, has been extended to field robotics. In addition, we present and evaluate the experimental results. Although event occurrences and state transitions have been added to the results, the presented *systemic approach* is excellent for analyzing discrete-event dynamics. Because of this, our proposed HSHC can be considered as simple but, on the other hand, also represents the most fundamental/essential component for the particular case of heterogeneous field robot analysis.

However, the SCT-based HSHC system has various drawbacks and challenges. The number of components of the hybrid plant (e.g., states, transition functions, and control maps) increases exponentially when the robot systems are modeled using hybrid automata. More complex systems require higher computational performance. In addition, when a different type of robot is added to the hybrid system, the corresponding low-level controller, hybrid automata model, and legal behavior model must also be designed as in other control systems. These issues may be more problematic than for conventional systems because these approaches make it challenging to represent plant and behavior specifications through hybrid automata. Nevertheless, if the above-mentioned components for control and implementation are constructed, the HSHC approach is efficient in managing heterogeneous field robot teams.

To address these challenges, there are decentralized, distributed, hierarchical, and top-down approaches to the design of extensive supervisory controllers [13], [52]. In addition, depending on the type of control method, an optimal, robust, adaptive supervisory controller can be considered for nonlinear hybrid dynamical systems [53]. The integration of artificial intelligence with HSHC and SCT for learning poses a future challenge [54]. The method of modeling a hybrid plant is a challenging and crucial problem, but it does not have a clear solution and only relies on the intuition of the designer. Therefore, artificial intelligence-based plant modeling is expected to provide helpful guidance for solving these issues. More detailed and advanced research that considers the observability of events, failure diagnosis [55], and parallel execution of hybrid systems, such as state/behavior trees [56], [57], is of utmost importance, and we intend to pursue these challenges in our future work.

VI. Conclusion

Herein, we proposed an HSHC architecture for the cooperative control of a heterogeneous field robot team using hybrid automata, and a formal modeling approach for a hybrid system consisting of CTS and DES. The performance and effectiveness of this approach were verified and evaluated through a heterogeneous field robot system consisting of UAVs and UGVs. For the modeled large-scale dynamic plant, we designed legal behavior (specifications) to achieve the control objectives. Modular supervisory controllers were synthesized based on these specifications. We also evaluated nonblocking, nonconflict, and controllability to verify that the modular supervisors met the specifications. All modular supervisors could control and manage the entire plant in the same way as a centralized supervisor. To evaluate our proposed HSHC approach and supervisory control system, we implemented and performed experiments in a physics-based simulator and a real field. The experimental results confirmed that the heterogeneous field robots achieved the given control objectives, and systematic results, such as the system behavior and the event strings, are presented. In addition, we validated the scalability of the proposed hybrid system by presenting how to model extended plants and synthesize modular controllers for high-level control architecture. Based on the results obtained from field experiments and the results of previous studies [20], the feasibility and effectiveness of our proposed HSHC system have been sufficiently validated. Therefore, hybrid systems and SCT-based hierarchical control are found to be more efficient and scalable than traditional control systems to control large-scale dynamic systems, such as heterogeneous robot systems. For example, when an additional field robot is included in the entire plant, the subplant is modeled with hybrid automata and synthesized in the plant, and the specifications for the added model are designed to obtain a modular supervisor. This is effective in addressing the complexity, modularity, and scalability. In our future work, we intend to apply more elaborate scenarios and implement the HSHC architecture fully using heterogeneous field robots in real outdoor environments to evaluate their practicality. Implementing this approach could be of significant value to advanced and autonomous field robot systems in the future (e.g., large-scale environmental sampling and monitoring, and smart farming). Research on these HSHC systems is expected to proceed in various directions, such as multitasking, multiareas, observability, and failure diagnosis.

References

[1] J. Kim, S. Kim, C. Ju, and H. I. Son, "Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications," *IEEE Access*, vol. 7, pp. 105100–105115, 2019.

[2] Y. Zhang, X. Liu, M. Luo, and C. Yang, "Bio-inspired approach for long-range underwater navigation using model predictive control," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 4286–4297, Aug. 2021.

[3] U. Ozdemir *et al.*, "Design of a commercial hybrid VTOL UAV system," *J. Intell. Robot. Syst.*, vol. 74, nos. 1–2, pp. 371–393, 2014.

[4] M. Schwarz *et al.*, "Nimbro rescue: Solving disaster-response tasks with the mobile manipulation robot momaro," *J. Field Robot.*, vol. 34, no. 2, pp. 400–425, 2017.
[5] M. Christiansen, M. Laursen, R. Jørgensen, S. Skovsen, and R. Gislum, "Designing and testing a uav mapping system for agricultural field surveying," *Sensors*, vol. 17, no. 12, p. 2703, 2017.
[6] A. Silwal, J. R. Davidson, M. Karkee, C. Mo, Q. Zhang, and K. Lewis, "Design, integration, and field evaluation of a robotic apple harvester," *J. Field Robot.*, vol. 34, no. 6, pp. 1140–1159, 2017.
[7] C. Ju and H. Son, "Multiple UAV systems for agricultural applications: Control, implementation, and evaluation," *Electronics*, vol. 7, no. 9, p. 162, 2018.
[8] C. Yuan, S. Licht, and H. He, "Formation learning control of multiple autonomous underwater vehicles with heterogeneous nonlinear uncertain dynamics," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2920–2934, Oct. 2018.
[9] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT*," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 993–1004, Jul. 2016.
[10] S. Manjanna, A. Q. Li, R. N. Smith, I. Rekleitis, and G. Dudek, "Heterogeneous multi-robot system for exploration and strategic water sampling," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 1–8.
[11] J. Roldán, P. Garcia-Aunon, M. Garzón, J. de León, J. del Cerro, and A. Barrientos, "Heterogeneous multi-robot system for mapping environmental variables of greenhouses," *Sensors*, vol. 16, no. 7, p. 1018, 2016.
[12] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Cham, Switzerland: Springer, 2021.
[13] W. M. Wonham, "Supervisory control of discrete-event systems," *Encyclopedia of Systems and Control*. Cham, Switzerland: Springer, 2021, pp. 1396–1404.
[14] H. I. Son, "Design and implementation of decentralised supervisory control for manufacturing system automation," *Int. J. Comput. Integr. Manuf.*, vol. 24, no. 3, pp. 242–256, 2011.
[15] C. R. C. Torrico, A. B. Leal, and A. T. Y. Watanabe, "Modeling and supervisory control of mobile robots: A case of a sumo robot," *IFAC-PapersOnLine*, vol. 49, no. 32, pp. 240–245, 2016.
[16] A. G. Gonzalez, M. V. Alves, G. S. Viana, L. K. Carvalho, and J. C. Basilio, "Supervisory control-based navigation architecture: A new framework for autonomous robots in industry 4.0 environments," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1732–1743, Apr. 2018.
[17] Q. Quan, Z. Zhao, L. Lin, P. Wang, W. M. Wonham, and K.-Y. Cai, "Failsafe mechanism design of multicopters based on supervisory control theory," 2017, *arXiv:1704.08605*.
[18] R. J. M. Theunissen, M. Petreczky, R. R. H. Schiffelers, D. A. van Beek, and J. E. Rooda, "Application of supervisory control synthesis to a patient support table of a magnetic resonance imaging scanner," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 20–32, Jan. 2014.
[19] A. Jayasiri, G. K. Mann, and R. G. Gosine, "Behavior coordination of mobile robotics using supervisory control of fuzzy discrete event systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 5, pp. 1224–1238, Oct. 2011.
[20] C. Ju and H. I. Son, "Modeling and control of heterogeneous agricultural field robots based on Ramadge-Wonham theory," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 48–55, Jan. 2020.
[21] C. Ju and H. I. Son, "Hybrid systems based modeling and control of heterogeneous agricultural robots for field operations," in *Proc. ASABE Annu. Int. Meeting*, 2019, pp. 1–5.
[22] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.
[23] H. I. Son and S. Lee, "Failure diagnosis and recovery based on des framework," *J. Intell. Manuf.*, vol. 18, no. 2, pp. 249–260, 2007.
[24] A. Tsalatsanis, A. Yalcin, and K. P. Valavanis, "Dynamic task allocation in cooperative robot teams," *Robotica*, vol. 30, no. 5, pp. 721–730, 2012.
[25] Y. Liu, M. Ficocelli, and G. Nejat, "A supervisory control method for multi-robot task allocation in urban search and rescue," in *Proc. IEEE Int. Symp. Safety Secur. Rescue Robot. (SSRR)*, 2015, pp. 1–6.
[26] J. Dulce-Galindo, M. A. Santos, G. V. Raffo, and P. N. Pena, "Autonomous navigation of multiple robots using supervisory control theory," in *Proc. 18th Eur. Control Conf. (ECC)*, 2019, pp. 3198–3203.
[27] F. J. Mendiburu, M. R. Morais, and A. M. Lima, "Behavior coordination in multi-robot systems," in *Proc. IEEE Int. Conf. Automatica (ICA-ACCA)*, 2016, pp. 1–7.
[28] G. W. Gamage, G. K. I. Mann, and R. G. Gosine, "Discrete event systems based formation control framework to coordinate multiple nonholonomic mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 4831–4836.
[29] Y. Tatsumoto, M. Shiraishi, K. Cai, and Z. Lin, "Application of online supervisory control of discrete-event systems to multi-robot warehouse automation," *Control Eng. Pract.*, vol. 81, pp. 97–104, Dec. 2018.
[30] C. Ju and H. I. Son, "Discrete event systems based modeling for agricultural multiple unmanned aerial vehicles: Automata theory approach," in *Proc. 18th Int. Conf. Control Autom. Syst. (ICCAS)*, 2018, pp. 258–260.
[31] X. Dai, L. Jiang, and Y. Zhao, "Cooperative exploration based on supervisory control of multi-robot systems," *Appl. Intell.*, vol. 45, no. 1, pp. 18–29, 2016.
[32] S. T. J. Forschelen, J. M. van de Mortel-Fronczak, R. Su, and J. E. Rooda, "Application of supervisory control theory to theme park vehicles," *Discrete Event Dyn. Syst.*, vol. 22, no. 4, pp. 511–540, 2012.
[33] Y. K. Lopes, S. M. Trenkwalder, A. B. Leal, T. J. Dodd, and R. Groß, "Supervisory control theory applied to swarm robotics," *Swarm Intell.*, vol. 10, no. 1, pp. 65–97, 2016.
[34] C. Jordan, C. Ma, and J. Provost, "An educational toolbox on supervisory control theory using MATLAB Simulink stateflow: From theory to practice in one week," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, 2017, pp. 632–639.
[35] L. P. Pinheiro, Y. K. Lopes, A. B. Leal, and R. S. U. R. Junior, "Nadzoru: A software tool for supervisory control of discrete event systems," *IFAC-PapersOnLine*, vol. 48, no. 7, pp. 182–187, 2015.
[36] A. M. Rahmani *et al.*, "Spectr: Formal supervisory control and coordination for many-core systems resource management," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 169–183, 2018.
[37] R. C. Hill and S. Lafortune, "Scaling the formal synthesis of supervisory control software for multiple robot systems," in *Proc. Amer. Control Conf. (ACC)*, 2017, pp. 3840–3847.
[38] R. Malik, K. Åkesson, H. Flordal, and M. Fabian, "Supremica–An efficient tool for large-scale discrete event systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5794–5799, 2017.
[39] H. Zhang, L. Feng, and Z. Li, "A learning-based synthesis approach to the supremal nonblocking supervisor of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3345–3360, Oct. 2018.
[40] W. Deng, J. Yang, and D. Qiu, "Supervisory control of probabilistic discrete event systems under partial observation," *IEEE Trans. Autom. Control*, vol. 64, no. 12, pp. 5051–5065, Dec. 2019.
[41] R. Liu, Y.-X. Wang, and L. Zhang, "An FDES-based shared control method for asynchronous brain-actuated robot," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1452–1462, Jun. 2016.
[42] M. Furci, R. Naldi, A. Paoli, and L. Marconi, "Robust supervisory-based control strategy for mobile robot navigation," in *Intelligent Autonomous Systems 13*. Cham, Switzerland: Springer, 2016, pp. 121–133.
[43] M. Goorden, C. Dingemans, M. A. Reniers, J. van de Mortel-Fronczak, W. J. Fokkink, and J. Rooda, "Supervisory control of multilevel discrete-event systems with a bus structure," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 3204–3211.
[44] C. Hu, C. Hu, D. He, and Q. Gu, "A new ROS-based hybrid architecture for heterogeneous multi-robot systems," in *Proc. 27th Chin. Control Decis. Conf. (CCDC)*, 2015, pp. 4721–4726.
[45] M. Milford and G. Wyeth, "Hybrid robot control and slam for persistent navigation and mapping," *Robot. Auton. Syst.*, vol. 58, no. 9, pp. 1096–1104, 2010.
[46] M. Egerstedt and X. Hu, "A hybrid control approach to action coordination for mobile robots," *Automatica*, vol. 38, no. 1, pp. 125–130, 2002.
[47] Z. Huang, D. Chu, C. Wu, and Y. He, "Path planning and cooperative control for automated vehicle platoon using hybrid automata," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 959–974, Mar. 2019.
[48] K. Cai, "Warehouse automation by logistic robotic networks: A cyber-physical control approach," *Front. Inf. Technol. Electron. Eng.*, vol. 21, no. 5, pp. 693–704, 2020.
[49] L. V. R. Alves and P. N. Pena, "Secure recovery procedure for manufacturing systems using synchronizing automata and supervisory control theory," *IEEE Trans. Autom. Sci. Eng.*, early access, Dec. 11, 2020, doi: 10.1109/TASE.2020.3041207.
[50] F. Mirzaei, A. A. Pouyan, and M. Biglari, "Automatic controller code generation for swarm robotics using probabilistic timed supervisory control theory (ptSCT)," *J. Intell. Robot. Syst.*, vol. 100, no. 2, pp. 729–750, 2020.
[51] T. A. Henzinger, "The theory of hybrid automata," in *Verification of Digital and Hybrid Systems*. Berlin, Germany: Springer, 2000, pp. 265–292.

[52] K. Cai and W. M. Wonham, "Supervisor localization: a top-down approach to distributed control of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 605–618, Mar. 2010.

[53] M. Buss, M. Glocker, M. Hardt, O. Von Stryk, R. Bulirsch, and G. Schmidt, "Nonlinear hybrid dynamical systems: Modeling, optimal control, and applications," in *Modelling, Analysis, and Design of Hybrid Systems*. Berlin, Germany: Springer, 2002, pp. 311–335.

[54] B. G. Kang, K.-M. Seo, and T. G. Kim, "Machine learning-based discrete event dynamic surrogate model of communication systems for simulating the command, control, and communication system of systems," *Simulation*, vol. 95, no. 8, pp. 673–691, 2019.

[55] Q. Quan, Z. Zhao, L. Lin, P. Wang, W. M. Wonham, and K.-Y. Cai, "Failsafe mechanism design of multicopters based on supervisory control theory," *IET Cyber-Syst. Robot.*, vol. 2, no. 1, pp. 31–42, 2020.

[56] A. Marzinotto, M. Colledanchise, C. Smith, and P. Õgren, "Towards a unified behavior trees framework for robot control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 5420–5427.

[57] X. Wang, Z. Li, and W. M. Wonham, "Real-time scheduling based on nonblocking supervisory control of state-tree structures," *IEEE Trans. Autom. Control*, vol. 66, no. 9, pp. 4230–4237, Sep. 2021.

Chanyoung Ju (Student Member, IEEE) received the B.S. and M.S. degrees from the Department of Rural and Biosystems Engineering, Chonnam National University, Gwangju, South Korea, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in biosystems engineering.

His research interests include field robotics, supervisory control, discrete event systems, and hybrid systems.

Hyoung Il Son (Senior Member, IEEE) received the B.S. and M.S. degrees from the Department of Mechanical Engineering, Pusan National University, Busan, South Korea, in 1998 and 2000, respectively, and the Ph.D. degree from the Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2010.

He also had several appointments both academia and industry as a Senior Researcher with LG Electronics, Pyeongtaek, South Korea, from 2003 to 2005, Samsung Electronics, Cheonan, South Korea, from 2005 to 2009, a Research Associate with the Institute of Industrial Science, The University of Tokyo, Tokyo, Japan, in 2010, and a Research Scientist with the Max Planck Institute for Biological Cybernetics, Tübingen, Germany, from 2010 to 2012. From 2012 to 2015, he lead the Telerobotics Group, Central Research Institute, Samsung Heavy Industries, Daejeon, as a Principal Researcher. He joined the Faculty of the Department of Rural and Biosystems Engineering, Chonnam National University, Gwangju, South Korea, in 2015, where he is currently an Associate Professor. His research interests include field robotics, agricultural robotics, haptics, teleoperation, and discrete event and hybrid systems.