

Energy-Aware Task Allocation for Teams of Multi-mode Robots

Takumi Ito¹, Riku Funada¹, Mitsuji Sampei¹, and Gennaro Notomista²

Abstract—This work proposes a novel multi-robot task allocation framework for robots that can switch between multiple modes, e.g., flying, driving, or walking. We first provide a method to encode the multi-mode property of robots as a graph, where the mode of each robot is represented by a node. Next, we formulate a constrained optimization problem to decide both the task to be allocated to each robot as well as the mode in which the latter should execute the task. The robot modes are optimized based on the state of the robot and the environment, as well as the energy required to execute the allocated task. Moreover, the proposed framework is able to encompass kinematic and dynamic models of robots alike. Furthermore, we provide sufficient conditions for the convergence of task execution and allocation for both robot models.

I. INTRODUCTION

Multi-robot systems perform increasingly complex tasks in numerous application fields, such as environmental monitoring [1], rescue [2], and delivery [3]. As such applications expand, today’s growth in robot technology raises the variety and complexity of robotic systems. Many robots are now designed with switchable operation or locomotion modes, such as flying, driving, and walking [4]–[6]. This multi-mode property makes robots more flexible, scalable, and resilient.

Switching between multiple modes provides advantages in both task execution and energy consumption. For instance, the convertible Unmanned Aerial Vehicle (UAV) surveyed in [4] has two flight modes: cruise and hovering. Cruise mode is energy-efficient in high forward velocity while hovering mode has advantages in takeoff spaces and static hovering. Similarly, robots with wheels and legs [5], or UAVs with wheels [6], are more energy efficient when using wheels. However, walking or flying offers advantages in executing tasks on harsh terrain.

In a scenario where multiple robots execute multiple tasks, Multi-Robot Task Allocation (MRTA) methods play a central role (see, e.g., [7], [8]). One key factor of MRTA is *energy*. Many MRTA problems were formulated to minimize travel distance [9] or energy consumption [10], or to manage limited energy resources [11]. These approaches can be regarded as energy-saving strategies. Another important

factor of MRTA is *resilience*. Heterogeneous multi-robot systems enhance resilience since if one robot fails, another can take over. The robots’ heterogeneity is encoded in [12] and [13]. Our previous work [10] expanded to address component failures and facilitate task reallocation. The robot’s multimodality has the potential to enhance both factors by selecting the most suitable mode according to capability and energy consumption. Nevertheless, to the best of the authors’ knowledge, no studies within the context of MRTA have explicitly considered the heterogeneity of multi-robot systems resulting from the multimodality of robotic units.

Numerous approaches have accomplished MRTA, including *market-based approaches* [14] and *behavior-based approaches* [15]. In contrast to many approaches, our previous work [16] formulated an *optimization problem* where each robot performs multiple tasks simultaneously with prioritizing, which is highlighted by lower computational complexity and dynamic allocation. Furthermore, we provided theoretical guarantees to the convergence of task execution and stable allocation in [10]. Moreover, when the optimization problem is convex, the formulation is scalable with respect to the number of robots and tasks (see, e.g., the constrained-based task execution approach in [17] based on Control Barrier Functions (CBFs) [18], which is, however, limited to velocity-controlled robots).

This letter proposes a novel task allocation framework that can switch between multiple modes of robots. The multimodality concept can be interpreted as a form of heterogeneity, and this analogy can be used to encode it. However, encompassing robots with multiple modes in task allocation frameworks is not straightforward. We further consider prioritizing between robot modes not only based on their capability, but also accounting for the energy consumption during task execution.

Although most existing works formulate task execution with velocity-controlled robots with the CBF-based method [18], many real robots have dynamics with high relative degree (e.g., acceleration-controlled robots executing position-related tasks). This makes formulating task executions difficult and requires a more sophisticated approach, such as integral CBF [19] or cascaded CBF [18]. Our proposed approach is able to encompass tasks and robot models, which result in high relative degree constraints. We theoretically examine a convergence condition of the task execution and allocation in both kinematics and high relative degree cases.

To summarize, the contributions of our work can be written as follows: 1) We propose a new MRTA framework that can handle robots with multiple modes. 2) We propose task execution strategies that can be applied to robot and

*This work was supported by JSPS KAKENHI, Grant Number 21H01348 and JST SPRING, Grant Number JPMJSP2180 (T. Ito, R. Funada, M. Sampei), and by the NSERC Discovery Grant RGPIN-2023-03703 (G. Notomista).

¹ Department of Systems and Control Engineering, Institute of Science Tokyo, Tokyo 152-8550, Japan, takumi.ito@sl.sc.e.titech.ac.jp, {funada, sampei}@sc.e.titech.ac.jp.

² Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada, gennaro.notomista@uwaterloo.ca.

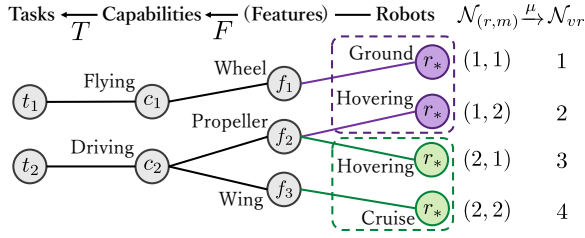


Fig. 1. Example of mapping, including two tasks, two capabilities, three features, and two robots with two modes. One robot is a UAV with wheels, which has ground vehicle and ground modes, and the other is a convertible UAV, which has hovering and cruise modes.

task models resulting in high relative degree constraints. 3) We provide theoretical conditions for convergence of task execution and allocation, considering both kinematic and dynamic robot models.

II. MULTI-MODE ROBOT MODELING

A. Task-to-mode Mapping

The overview of the proposed encoding mapping is depicted in Fig. 1. An encoding framework for robot heterogeneity is proposed in [10] with multiple layers: robots, features, capabilities, and tasks. The layer of robots incorporates two factors: heterogeneity of robots (distinguished by purple and green colors in Fig. 1) and modes of each robot (nodes in each of the dashed boxes). Each node expressing a mode is denoted by a *virtual robot* hereafter.

Let us exemplify how this framework incorporates a scenario in which a wheel-equipped UAV, which is depicted as purple nodes in Fig. 1, conducts two tasks: 1) monitoring a targeted area from the air and 2) traversing a region not allowed flying due to law regulations. Each task requires a robot to own a specific capability. This requirement is expressed by a connection between the nodes of tasks and capabilities. Similarly, each capability is connected by features that grant a robot its capability. A wheel-equipped UAV switches its mode between hovering and ground by selecting a virtual robot according to its features.

Let the number of robots, features, capabilities, and tasks be n_r , n_f , n_c , and n_t , respectively. The following index sets are subsets of natural number \mathbb{N} . The ordered sets of robot and task indices are denoted as $\mathcal{N}_r = \{1, \dots, n_r\}$ and $\mathcal{N}_t = \{1, \dots, n_t\}$. Let robot i 's number of modes and a set of its indices be m_i and $\mathcal{N}_{m_i} = \{1, \dots, m_i\}$, respectively.

From the robot and mode indices, we now define a set of ordered pairs of robot and mode as $\mathcal{N}_{(r,m)} = \{(1, 1), \dots, (1, m_1), \dots, (n_r, 1), \dots, (n_r, m_{n_r})\}$, whose number of elements is denoted as $n_{vr} = \sum_{i \in \mathcal{N}_r} m_i$. Then, we define a set of virtual robot indices as $\mathcal{N}_{vr} = \{1, \dots, n_{vr}\}$, which is order isomorphic to $\mathcal{N}_{(r,m)}$. Let us denote “virtual robot of robot i in mode k ” as “virtual robot $\mu(i, k)$ ” with a bijection mapping $\mu : \mathcal{N}_{(r,m)} \rightarrow \mathcal{N}_{vr}$. For instance, the mapping μ in the case of two robots with two modes is shown in the right part of Fig. 1. Now, we can leverage the robot-to-task mapping introduced in [10], using virtual robots, as explained in the following.

1) *Mapping Matrices*: First, let us define a *robot-to-capability mapping matrix* as $F \in \{0, 1\}^{n_c \times n_{vr}}$, where $F_{\ell, \mu(i, k)} = 1$ if and only if virtual robot $\mu(i, k)$ possesses capability ℓ . Note that this robot-to-capability mapping is formally defined using *robot-to-feature mapping* and a *feature-to-capability mapping* defined as bipartite hypergraph; refer to [10] for a complete definition. Then, a *capability-to-task mapping matrix* is defined as $T \in \{0, 1\}^{n_t \times n_c}$, where $T_{j, \ell} = 1$ if and only if task j requires capability ℓ .

2) *Specialization and Penalty*: Additionally, we define the *specialization matrix*, which encodes potential candidates of tasks for virtual robots, as $S_{\mu(i, k)} = \text{diag}(\mathbb{1}_{n_t} - \text{kron}(TF_{-, \mu(i, k)})) \in \mathbb{R}^{n_t \times n_t}$, where $\mathbb{1}_{n_t} \in \mathbb{R}^{n_t}$ is a vector consisting of 1, and kron is the Kronecker delta function. The symbols $X_{i, -}$ and $X_{-, j}$ denote the i th row and the j th column of the matrix X , respectively. The specialization matrix $S_{\mu(i, k)}$ becomes a diagonal matrix whose (j, j) element is 1 if task j is a potential candidate for virtual robot $\mu(i, k)$; otherwise, it is 0. Specifically, when virtual robot $\mu(i, k)$ possesses at least one capability to support task j , the element value becomes 1. Moreover, we define a penalty matrix as $\Pi_{\mu(i, k)} = I_{n_t} - S_{\mu(i, k)}^\dagger S_{\mu(i, k)}^\dagger$, where the dagger is pseudo inverse. The penalty matrix can be regarded as a penalty for task assignment to an infeasible robot since its (j, j) element has a nonzero value if the task j is infeasible, contrasting the specialization matrix.

Example 1: Consider a convertible UAV with cruise and hovering modes, which are depicted as green nodes in Fig. 1. The virtual robots of cruise and hovering modes have *wing* and *propeller*, respectively, and both features support *flying* capability. Although both modes can execute the same task, energy efficiency can be differentiated by specifying distinct energy costs, which are introduced later. By assigning a lower energy cost to cruise mode at high forward velocities than hovering mode, we can enable cruise mode to excel in tasks requiring high forward speed. This robot can adjust energy efficiency by switching its modes while executing the same task.

B. System Models

1) *Motion Model*: Consider the virtual robot $\mu(i, k)$. Let the dimension of the state and input be n_{x_i} and $n_{u_{i, k}}$, respectively. Let the state of the robot be shared for all modes; then, we denote the state only with a subscript i as $x_i \in \mathbb{R}^{n_{x_i}}$. In contrast, the input is denoted as $u_{i, k} \in \mathbb{R}^{n_{u_{i, k}}}$. In this letter, we consider mechanical robotic systems, therefore we model their dynamics via control affine dynamical systems. The state evolution of virtual robot $\mu(i, k)$ is then governed by the following differential equation:

$$\dot{x}_i = f_{i, k}(x_i) + g_{i, k}(x_i)u_{i, k}, \quad (1)$$

where $f_{i, k} : \mathbb{R}^{n_{x_i}} \rightarrow \mathbb{R}^{n_{x_i}}$ and $g_{i, k} : \mathbb{R}^{n_{x_i}} \rightarrow \mathbb{R}^{n_{x_i} \times n_{u_{i, k}}}$ are locally Lipschitz continuous.

2) *Energy Model*: The above discussion can distinguish the capability of task execution of each mode. However, if the capability is the same for multiple modes, how can we allocate tasks? Here, we introduce the energy cost to

differentiate the task execution performance based on the energy consumption.

The previous works (e.g., [10]) represent energy cost as $\|u\|^2$. This can be more generalized for some situations. For instance, an airplane is operated more energy efficiently at high speeds than at low speeds because it can utilize aerodynamic lift. Therefore, in this letter, we consider the energy cost function a more general convex function as

$$\varepsilon_{i,k}(u_{i,k}) = \|u_{i,k} - u_{i,k}^{\text{eff}}\|_{w_{i,k}}^2, \quad (2)$$

where $w_{i,k}$ is a weight for the norm, and u^{eff} is the input that minimizes energy cost (e.g., specific forward velocity). This formulation does not necessarily represent exact energy consumption. However, the quadratic formulation can approximate it and has advantages in handling and implementation.

III. TASK EXECUTION AND ALLOCATION

A. Preliminary: Constraint-Based Task Execution

This subsection introduces a multiple-task execution method based on the extended set-based tasks [17]. That can theoretically guarantee the convergence of the task completion. For the following discussion, we will temporarily omit the subscripts of the indices of robots, modes, and tasks.

Suppose continuously differentiable function $h(x) \leq 0$ represents a task, and maximizing it as $h(x) \rightarrow 0$ means the completion of the task. Let its zero super-level set be $C = \{x \mid h(x) \geq 0\} = \{x \mid h(x) = 0\}$. Then, the control aim is to converge x into C . The following definition and theorem ensure forward invariance and asymptotically stability of C .

Definition 1 (Control Barrier Function (CBF) [18]):

Given a system in (1), and a function $h(x)$ and its zero super-level set C . The function $h(x)$ is a *Control Barrier Function* if there exists a Lipschitz continuous extended class- \mathcal{K} function γ , such that $\sup\{L_f h(x) + L_g h(x)u\} \geq -\gamma(h(x))$, where $L_f h(x)$ and $L_g h(x)$ denote Lie derivative.

Theorem 1 (Forward invariance [18]): If $h(x)$ is a CBF, then any Lipschitz continuous controller $u \in \{u \mid L_f h(x) + L_g h(x)u + \gamma(h(x)) \geq 0\}$ for (1) renders the zero super-level set C forward invariance and asymptotically stable in \mathbb{R}^{n_x} .

It is shown in [20] that the maximization of $h(x)$ while reducing the energy cost is realized by solving the following optimization problem

$$\begin{aligned} & \underset{u, \delta}{\text{minimize}} \quad \varepsilon(u) + \delta^2 \\ & \text{subject to} \quad L_f h(x) + L_g h(x)u \geq -\gamma(h(x)) - \delta. \end{aligned} \quad (3)$$

Consider the virtual robot $\mu(i, k)$ now in a multi-task situation. Each task is encoded by a CBF, $h_j, j \in \mathcal{N}_t$. Then, the constrained optimization problems can be combined into a single optimization problem as follows [10]:

$$\begin{aligned} & \underset{u_{i,k}, \delta_{\mu(i,k)}}{\text{minimize}} \quad \varepsilon_{i,k}(u_{i,k}) + \|\delta_{\mu(i,k)}\|^2 \\ & \text{subject to} \quad L_{f_{i,k}} h_j(x_i) + L_{g_{i,k}} h_j(x_i)u_{i,k} \\ & \quad \geq -\gamma(h_j(x_i)) - \delta_{\mu(i,k)j}, \quad \forall j \in \mathcal{N}_t, \end{aligned}$$

where $\delta_{\mu(i,k)} = [\delta_{\mu(i,k)1} \dots \delta_{\mu(i,k)n_t}]^\top$ denotes the slack variables for the all tasks.

B. Task Allocation for Multi-mode Robots

Now, we are ready to formulate an optimization problem that allocates tasks to multi-mode robots. Let $\alpha \in \{0, 1\}^{n_t \times n_{vr}}$ be an allocation matrix such that

$$\alpha_{j,\mu(i,k)} = \begin{cases} 1 & \text{if task } j \text{ is assigned to virtual robot } \mu(i,k) \\ 0 & \text{otherwise} \end{cases}$$

We need to ensure no two virtual instances of the same robot—corresponding to its modes—are assigned tasks simultaneously, which means the following condition must be satisfied.

$$\sum_{k=1}^{m_i} \sum_{j=1}^{n_t} \alpha_{j,\mu(i,k)} \leq 1, \quad \forall i \in \mathcal{N}_r \quad (4)$$

Considering this condition, the optimization problem for task allocation to multi-mode robots is defined as follows.

$$\begin{aligned} & \underset{u, \delta, \alpha}{\text{minimize}} \quad \sum_{i=1}^{n_r} \sum_{k=1}^{m_i} \left(l_1 \|\Pi_{\mu(i,k)} \alpha_{-, \mu(i,k)}\|^2 \right. \\ & \quad \left. + \varepsilon_{i,k}(u_{i,k}) + l_2 \|\delta_{\mu(i,k)}\|_{S_{\mu(i,k)}}^2 \right) \end{aligned} \quad (5a)$$

$$\begin{aligned} & \text{subject to} \quad L_{f_{i,k}} h_j(x_i) + L_{g_{i,k}} h_j(x_i)u_{i,k} \\ & \quad \geq -\gamma(h_j(x_i)) - \delta_{\mu(i,k)j} \end{aligned} \quad (5b)$$

$$\Theta_i \delta_{[i]} + \Phi_i \alpha_{[i]} \leq \Psi_i \quad (5c)$$

$$\mathbb{1}_{n_t m_i}^\top \alpha_{[i]} \leq 1 \quad (5d)$$

$$F \alpha_{j,-}^\top \geq T_{j,-}^\top \quad (5e)$$

$$n_{j,\min} \leq \mathbb{1}_{n_{vr}}^\top \alpha_{j,-}^\top \leq n_{j,\max} \quad (5f)$$

$$\|\delta_{\mu(i,k)}\|_\infty \leq \delta_{\max} \quad (5g)$$

$$\alpha \in \{0, 1\}^{n_t \times n_{vr}} \quad (5h)$$

$$\forall k \in \mathcal{N}_{m_i}, \quad \forall j \in \mathcal{N}_t, \quad \forall i \in \mathcal{N}_r,$$

where $\delta_{[i]} = [\delta_{\mu(i,1)}^\top \dots \delta_{\mu(i,m_i)}^\top]^\top$ and $\alpha_{[i]} = [\alpha_{-, \mu(i,1)}^\top \dots \alpha_{-, \mu(i,m_i)}^\top]^\top$, and the other undefined symbols will be defined hereafter. The program in (5) is a mixed-integer quadratic program.

The objective function (5a), with weighting parameters $l_1, l_2 \in \mathbb{R}_{>0}$, aims to minimize the penalty of task allocation, energy cost, and slack variables. The first term prevents tasks from being assigned to infeasible virtual robots using the penalty matrix $\Pi_{\mu(i,k)}$. The second term minimizes the energy costs. The third term minimizes the weighted norm of slack variables with a measure $S_{\mu(i,k)}$. This weighting can relax the constraint on the slack variable of a task with an infeasible virtual robot. The constraint (5b) ensures task execution as introduced in Section III-A. The constraint (5c) is a prioritization between modes and tasks. The matrices $\Theta_i, \Phi_i \in \mathbb{R}^{\frac{n_t m_i (n_t - 1)(m_i - 1)}{2} \times n_t m_i}$ and $\Psi_i \in \mathbb{R}^{\frac{n_t m_i (n_t - 1)(m_i - 1)}{2}}$ are obtained by collecting the inequalities:

$$\delta_{j,\mu(i,k)} \leq \kappa^{-1} \delta_{j,\mu(i,k')} + \delta_{\max}(1 - \alpha_{j,\mu(i,k)}), \quad (6a)$$

$$\begin{aligned} \delta_{j,\mu(i,k)} & \leq \kappa^{-1} \delta_{j',\mu(i,k)} + \delta_{\max}(1 - \alpha_{j,\mu(i,k)}), \\ & \quad \forall k' \in \mathcal{N}_{m_i} \setminus \{k\}, \quad \forall j' \in \mathcal{N}_t \setminus \{j\}, \end{aligned} \quad (6b)$$

with $\kappa \gg 1$. Inequality (6a) prioritizes between modes. When task j is assigned to robot i in mode k , the second term on the right-hand side of (6a) becomes zero. This means the slack variable for mode k needs to be smaller than $1/\kappa$ times one for mode k' , prioritizing task execution represented in (5b) of mode k over mode k' . Otherwise, this term becomes δ_{\max} , which makes the constraint redundant with (5g). Analogously, inequality (6b) prioritizes tasks.

Remark 1: This prioritization aims to switch modes based on energy cost. The modes that can satisfy the task constraint (5b) with low energy input should be prioritized. If a less efficient mode is prioritized, it will try to satisfy the constraint (5b) with inefficient input, causing a large energy cost. The solution α , which minimizes the total cost, including energy, will be selected (see also discussion in [16]).

Inequalities (5d), (5e), and (5f) are constraints for the number of robots. The constraint (5d) ensures that no more than one task is assigned to each robot, which is equivalent to (4). Inequality (5e) ensures that the required features are assigned to a task. Finally, via (5f) we can set the minimum and maximum number of robots for each task.

C. Analysis of Convergence

This section provides a condition for the optimization problem (5) to complete tasks and generate a stable task allocation.

As the task allocation problem for multi-mode robots (5) is formulated resembles one proposed in [10], adopting the convergence condition in [10] is facilitated. We only need to redefine variables to fit our multi-mode framework as follows. Let $h(x) = [h_1(x) \dots h_{n_t}(x)]^\top$ with $x = [x_1^\top \dots x_{n_r}^\top]^\top$, $u = [u_{1,1}^\top \dots u_{1,m_i}^\top \dots u_{n_r,1}^\top \dots u_{n_r,m_{n_r}}^\top]^\top$, $\delta = [\delta_1^\top \dots \delta_{n_r}^\top]^\top$, and $\bar{\alpha} = [\alpha_1^\top \dots \alpha_{n_r}^\top]^\top$. Then, define $\varphi = [\gamma(h(x))^\top u^\top \delta^\top \bar{\alpha}^\top 1]^\top$. Additionally, let $\bar{\Phi} = \text{diag}_b(\Phi_1, \dots, \Phi_{n_r})$, $\bar{\Theta} = \text{diag}_b(\Theta_1, \dots, \Theta_{n_r})$, and $\bar{\Psi} = \text{diag}_b(\Psi_1, \dots, \Psi_{n_r})$, where diag_b returns a block diagonal matrix. The constraints (5d)-(5g) can be rewritten as $\bar{\alpha}^\top A_\alpha^\top A_\alpha \bar{\alpha} \leq \bar{\alpha}^\top A_\alpha^\top b_\alpha$ and $\delta^\top A_\delta^\top A_\delta \delta \leq \delta^\top A_\delta^\top b_\delta$.

Then, the following proposition provides a condition to establish the convergence of the task allocation problem, which is represented as a linear matrix inequality (LMI).

Proposition 1: If there exist positive scalars τ_1 , τ_2 , and τ_3 for all time $t \geq 0$ that satisfy

$$B_0 \leq \tau_1 B_1 + \tau_2 B_2 + \tau_3 B_3, \quad (7)$$

where B_0 , B_1 , B_2 , and B_3 with $c \in \mathbb{R}_{>0}$ are given in Appendix, then the sequences $u(t)$, $\delta(t)$, $\alpha(t)$, and solutions of (5) converge as $t \rightarrow \infty$.

Proof: Similar to [10]. ■

IV. HIGH RELATIVE DEGREE TASK EXECUTION AND ALLOCATION

This section extends the proposed framework to a scenario with task execution constraints that lead to high relative degree. As a motivating example, we utilize a convertible UAV introduced in Example 1 with tasks represented by its position. In such scenarios, $L_{g_{i,k}} h_j(x_i)$ in (5b) becomes

zero; hence, the constraint (5b) cannot be employed as a constraint for the input. To address this issue, Section IV-B and Section IV-C extend the constraint (5b) and Proposition 1, respectively.

A. Modeling a Convertible UAV

1) *Dynamics:* The convertible UAVs described in Example 1 have two modes, 1: cruise and 2: hovering. Each of these two modes has different control inputs that stem from their features. Specifically, the cruise mode is driven by a forward velocity and a yaw rate input, while the hovering mode is governed by horizontal velocity input.

Let us first derive a generalized equation of motion shared between both modes. Let $x_i \in \mathbb{R}^2$ and $\eta_i = [v_{i,x} \ v_{i,y} \ \theta_i]^\top$ be the position and the input consisting of horizontal velocity and yaw orientation of the UAV i , respectively. Then, the state derivative can be written as

$$\dot{x}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} v_{i,x} \\ v_{i,y} \end{bmatrix} = f_i(\eta_i) \quad (8)$$

Additionally, to prevent a discontinuous change in the linear velocity and to model the transition phase between the two modes, we introduce the input dynamics as

$$\dot{v}_{i,x} = k_v(v_{i,x}^{\text{ref}} - v_{i,x}), \quad \dot{v}_{i,y} = k_v(v_{i,y}^{\text{ref}} - v_{i,y}), \quad \dot{\theta}_i = \omega_i, \quad (9)$$

with positive gain $k_v \in \mathbb{R}_{>0}$. Then, (9) can be rewritten as

$$\dot{\eta}_i = \begin{bmatrix} -k_v v_{i,x} \\ -k_v v_{i,y} \\ 0 \end{bmatrix} + \begin{bmatrix} k_v v_{i,x}^{\text{ref}} \\ k_v v_{i,y}^{\text{ref}} \\ \omega_i \end{bmatrix} = \psi(\eta_i) + g_i \begin{bmatrix} v_{i,x}^{\text{ref}} \\ v_{i,y}^{\text{ref}} \\ \omega_i \end{bmatrix}. \quad (10)$$

Because $\psi(\eta_i)$ can be regarded as a viscous friction, this formulation mitigates a discontinuity in vehicle velocity.

Next, we derive dynamics specific to each mode based on (8) and (10). For the cruise mode, its input is composed of a forward velocity and yaw rate as $u_{i,1} = [v_{i,x}^{\text{ref}} \ \omega_i]^\top$. To incorporate this new input into (10), we substitute $v_{i,y}^{\text{ref}} = 0$ and replace g_i with $g_{i,1}$ which is obtained by eliminating the second column of g_i . Then, we get the state space equation of the cruise mode as

$$\frac{d}{dt} \begin{bmatrix} x_i \\ \eta_i \end{bmatrix} = \begin{bmatrix} f_i(\eta_i) \\ \psi(\eta_i) + g_{i,1} u_{i,1} \end{bmatrix}. \quad (11a)$$

By following similar procedures, the state space equation of hovering mode is derived as

$$\frac{d}{dt} \begin{bmatrix} x_i \\ \eta_i \end{bmatrix} = \begin{bmatrix} f_i(\eta_i) \\ \psi(\eta_i) + g_{i,2} u_{i,2} \end{bmatrix}. \quad (11b)$$

where $u_{i,2} = [v_{i,x}^{\text{ref}} \ v_{i,y}^{\text{ref}}]^\top$ and $\omega_i = 0$. Note that $g_{i,2}$ is obtained by removing the third column from g_i .

2) *Energy model:* We define energy cost functions as

$$\varepsilon_{i,1}(u_{i,1}) = (v_{i,x}^{\text{ref}} - v_x^{\text{eff}})^2 + \omega_i^2, \quad (12a)$$

$$\varepsilon_{i,2}(u_{i,2}) = v_{i,x}^{\text{ref}2} + v_{i,y}^{\text{ref}2}, \quad (12b)$$

where $v_x^{\text{eff}} \in \mathbb{R}_{>0}$ is the forward velocity of the cruise mode that minimizes energy cost. If we consider only $v_{i,x}^{\text{ref}}$, under high forward velocity reference, cruise mode is regarded as more energy-efficient than hovering mode. That means cruise mode has an advantage in executing a task that requires high-velocity forward movement.

B. High Relative Degree Task Execution

Here, we extend the condition for task execution (5b) to a high relative degree case. The task used in the simulations is to minimize the distance between a robot and a target point. Hence, we define a CBF as

$$h_j(x_i) = -\|x_i - p_j\|^2, \quad (13)$$

where p_j is the task j 's target position. The convergence to its zero super-level set $C_1 = \{x_i \mid h_j(x_i) \geq 0\} = \{p_j\}$ means the accomplishment of the task j .

Because the UAV model has input dynamics, we need to employ integral CBFs [19]. According to [19], C_1 is rendered to be forward invariant when the input satisfies

$$\begin{aligned} \dot{h}_j + \gamma_1(h_j) \\ = \frac{\partial h_j}{\partial x_i} f_i(\eta_i) + \underbrace{\frac{\partial h_j}{\partial \eta_i} (\phi_i(\eta_i) + g_{i,k} u_{i,k})}_{=0} + \gamma_1(h_j) \geq 0, \end{aligned} \quad (14)$$

where γ_1 is an extended class- \mathcal{K} function. However, as $\partial h_j / \partial \eta_i$ is zero, (14) does not constrain the actual input $u_{i,k}$. Then, we define a second CBF candidate as

$$h'_j(x_i, \eta_i) = \frac{\partial h_j}{\partial x_i} f_i(\eta_i) + \gamma_1(h_j). \quad (15)$$

The input $u_{i,k}$ which satisfies

$$\begin{aligned} \dot{h}'_j + \gamma_2(h'_j) &= \frac{\partial}{\partial x_i} \left(\frac{\partial h_j}{\partial x_i} f_i(\eta_i) \right) f_i(\eta_i) \\ &+ \frac{\partial}{\partial \eta_i} \left(\frac{\partial h_j}{\partial x_i} f_i(\eta_i) \right) (\phi_i(\eta_i) + g_{i,k} u_{i,k}) \\ &+ \frac{\partial \gamma_1}{\partial h_j} \dot{h}_j + \gamma_2(h'_j(x_i)) \geq 0 \end{aligned} \quad (16)$$

renders $C_2 = \{(x_i, \eta_i) \mid h'_j(x_i, \eta_i) \geq 0\}$ forward invariant; simultaneously, this input also renders C_1 forward invariant (see discussion in [18]). Therefore, we can utilize (16) in (5), instead of the constraint (5b).

C. Convergence of High Relative Degree Task Allocation

In this section we give sufficient conditions for the convergence of the task allocation algorithm (5) for the case of high relative degree tasks.

Proposition 2: If there exist positive scalars τ_1 , τ_2 , and τ_3 for all time $t \geq 0$ that satisfy

$$B'_0 \leq \tau_1 B_1 + \tau_2 B_2 + \tau_3 B_3, \quad (17)$$

where B'_0 , B_1 , B_2 , and B_3 are reported in the Appendix, then the sequences $u(t)$, $\delta(t)$, $\alpha(t)$, and solutions of (5) converge as $t \rightarrow \infty$.

Proof: Let the candidate Lyapunov function be

$$V(x) = h(x)^\top h(x). \quad (18)$$

Then, we want the following condition on its time derivative to hold:

$$\dot{V}(x, \eta) = 2h(x)^\top \frac{\partial h}{\partial x} f(\eta) \leq -c_1 V(x). \quad (19)$$

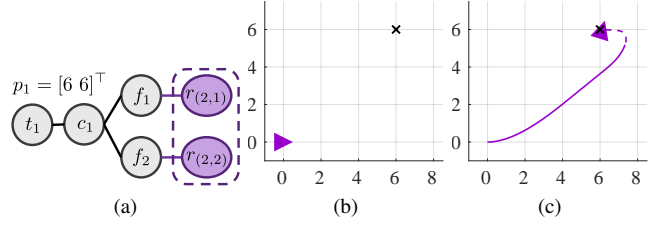


Fig. 2. Single UAV simulation. (a) Encoding: Both modes can execute the task. (b) Initial state. (c) Result trajectory: The trajectories of the cruise and hovering mode are shown as solid and dashed lines, respectively.

with $c_1 \in \mathbb{R}_{>0}$, where $f(\eta) = [f_1(\eta)^\top \dots f_{n_r}(\eta)^\top]^\top$ with $\eta = [\eta_1^\top \dots \eta_{n_r}^\top]^\top$. We define the auxiliary Lyapunov function $V'(x, \eta) = -\dot{V}(x, \eta) - c_1 V(x) \geq 0$, and enforce the stability condition $\dot{V}'(x, \eta) \leq -c_2 V'(x, \eta)$, for $c_2 \in \mathbb{R}_{>0}$, which can be written as

$$-\ddot{V}(x, \eta) - (c_1 + c_2) \dot{V}(x, \eta) - c_1 c_2 V(x) \leq 0. \quad (20)$$

The condition (20) can be further simplified to

$$\varphi^\top B'_0 \varphi \leq 0, \quad (21)$$

where B'_0 is given in the Appendix and φ is given in Section III-C. The same steps used in proposition 1 complete the proof. \blacksquare

V. SIMULATIONS

This section provides simulations to demonstrate the proposed method with the settings in Section IV. The parameters are set as $k_v = 4$, $l_1 = 10^6$, $l_2 = 10^{-4}$, $\kappa = 10^4$, $\delta_{\max} = 10^4$, $v_x^{\text{eff}} = 2$ m/s, $\gamma_1(h) = 5h$, and $\gamma_2(h) = h$. We conduct two simulations: one is a single task with a single UAV, and the other is multiple tasks with multiple UAVs under the constraint that cruise mode is prohibited in a given area (modeling, e.g., a regulatory restriction).

The graph encoding the first simulation is shown in Fig. 2 (a). The initial state and the target position of the task are shown in Fig. 2 (b). The result trajectory is shown in Fig. 2 (c). Initially, when the robot flies forward, the cruise mode is assigned to save energy. Then, when the robot gets closer to the target point, the hovering mode is assigned to accomplish the task. This result shows the proposed allocation method can switch robot modes according to energy consumption and task execution ability.

The second simulation features multiple UAVs and tasks. The initial state and the target points of the tasks are set as shown in Fig. 3 (b), where the crosses represent the task target points. Moreover, we enforce a no-cruise area to model a regulatory restriction shown as the brown area in Fig. 3. If a UAV enters this area, the specialization matrix of the cruise mode $S_{\mu(i,1)}$ becomes a zero matrix, preventing the tasks from being assigned to cruise mode. There are three UAVs, and the rightmost UAV can only cruise; therefore, it would immediately stop as it enters the restricted area. The remaining UAVs are able to both cruise and hover. Fig. 3 shows the snapshots of the second simulation result. At first, the nearest UAVs are assigned to each task—an allocation

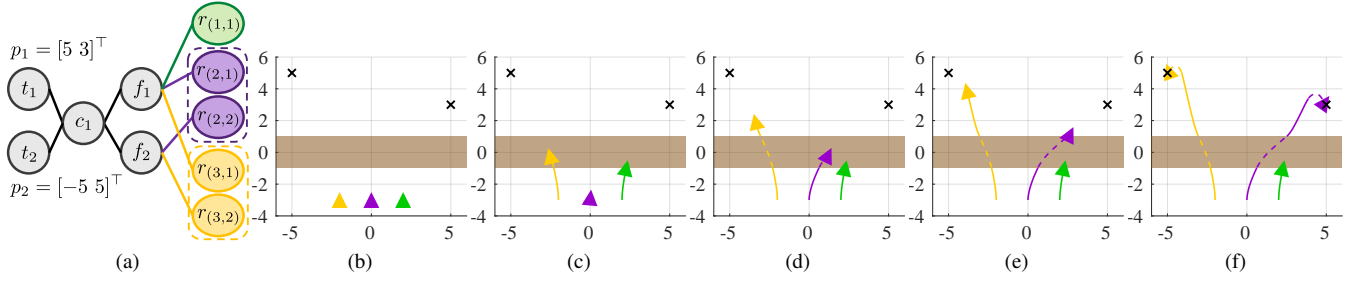


Fig. 3. Multiple UAV simulation with a regulation restriction. Each robot is drawn in the same color in all figures. (a) Encoding: Robot 1 has only cruise mode, while the others have both modes. (b) Snapshot of the result at the time $t = 0$ s. The result trajectories of the cruise and hovering modes are drawn in solid and dashed lines in the same color of its robot. The cruise mode is prohibited in the brown area. (c) $t = 1$ s. (d) $t = 2$ s. (e) $t = 3$ s. (f) $t = 10$ s.

that optimizes energy consumption (Fig. 3 (b)-(c)). However, at time $t = 1$ s (Fig. 3 (c)), the right UAV stops because of the regulation; then, the middle UAV, which has not been assigned to any task, is recruited to go to the rightmost cross. Finally, the two UAVs complete each task by switching their modes according to the regulation and energy consumption (Fig. 3 (d)-(f)). This result demonstrates that the proposed method provides resilience under environmental restrictions.

VI. CONCLUSION

This work proposed a novel multi-robot task allocation framework designed for multi-mode robots. First, the multi-modality was encoded as a linear mapping between tasks and modes, where robot modes are represented by virtual robot nodes. Then, a constrained optimization problem was formulated to allocate tasks considering robot capabilities, energy consumption, and task execution. The proposed framework is able to encompass high relative degree task models, and guarantees of task allocation convergence are provided. Simulations demonstrated how the proposed framework allows robots to execute tasks by switching their mode according to energy consumption.

APPENDIX

The matrices used in Proposition 1 and 2 are defined as symmetric sparse matrices whose upper triangular parts are defined as follows, where $B_*(i, j)$ indicates (i, j) block of the matrix. $B_0(1, 1) = cI$, $B_0(1, 3) = \frac{d\gamma}{dh} \frac{dh}{dx} g(x)$, $B_0(1, 5) = \frac{d\gamma}{dh} \frac{dh}{dx} f(x)$. $c \in \mathbb{R}_{>0}$. $B_1(1, 3) = -\frac{1}{2}I$, $B_1(2, 3) = -\frac{1}{2}L_g h(x)^\top$, $B_1(3, 3) = -I$, $B_1(3, 5) = -\frac{1}{2}L_f h(x)$. $B_2(3, 4) = \frac{1}{2}\bar{\Theta}^\top \bar{\Phi}$, $B_2(4, 4) = \bar{\Phi}^\top \bar{\Phi}$, $B_2(4, 5) = -\frac{1}{2}\bar{\Phi}^\top \bar{\Psi}$. $B_3(3, 3) = A_\delta^\top A_\delta$, $B_3(3, 5) = -\frac{1}{2}A_\delta^\top b_\delta$, $B_3(4, 4) = A_\alpha^\top A_\alpha$, $B_3(4, 5) = -\frac{1}{2}A_\alpha^\top b_\alpha$. $B'_0(1, 1) = c_1 c_2 I$, $B'_0(1, 2) = \frac{\partial}{\partial \eta} \left(\frac{\partial h}{\partial x} f(\eta) \right) g$, $B'_0(1, 5) = \frac{\partial}{\partial x} \left(\frac{\partial h}{\partial x} f(\eta) \right) f(\eta) + \frac{\partial}{\partial \eta} \left(\frac{\partial h}{\partial x} f(\eta) \right) \phi(\eta) + (c_1 + c_2) \frac{\partial h}{\partial x} f(\eta)$, $B'_0(5, 5) = -2 \left\| \frac{\partial h}{\partial x} f(\eta) \right\|^2$. $c_1, c_2 \in \mathbb{R}_{>0}$.

REFERENCES

- [1] S. Kim, R. Lin, S. Coogan, and M. Egerstedt, "Area coverage using multiple aerial robots with coverage redundancy and collision avoidance," *IEEE Control Syst. Lett.*, vol. 8, pp. 610–615, 2024.
- [2] S. Saeedvand, H. S. Aghdasi, and J. Baltes, "Robust multi-objective multi-humanoid robots task allocation based on novel hybrid meta-heuristic algorithm," *Appl. Intell.*, vol. 49, no. 12, pp. 4097–4127, 2019.
- [3] K. S. Yakovlev, A. Andreychuk, T. Rybecký, and M. Kulich, "On the application of safe-interval path planning to a variant of the pickup and delivery problem," in *Proc. Int. Conf. Inform. Control, Autom. Robot.*, 2020, pp. 521–528.
- [4] G. J. Ducard and M. Allenspach, "Review of designs and flight control techniques of hybrid and convertible VTOL UAVs," *Aerosp. Sci. Technol.*, vol. 118, 2021, Art. no. 107035.
- [5] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Sci. Robot.*, vol. 9, no. 89, 2024, Art. no. eadi9641.
- [6] E. Sihite, A. Kalantari, R. Nemovi, A. Ramezani, and M. Gharib, "Multi-Modal Mobility Morphobot (M4) with appendage repurposing for locomotion plasticity enhancement," *Nature Commun.*, vol. 14, no. 1, 2023, Art. no. 3323.
- [7] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The Int. J. Robot. Research*, vol. 23, no. 9, pp. 939–954, 2004.
- [8] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The Int. J. Robot. Research*, vol. 32, no. 12, pp. 1495–1512, 2013.
- [9] D. Panagou, M. Turpin, and V. Kumar, "Decentralized goal assignment and safe trajectory generation in multirobot networks via multiple lyapunov functions," *IEEE Trans. Autom. Control*, vol. 65, no. 8, pp. 3365–3380, 2020.
- [10] G. Notomista, S. Mayya, Y. Emam, C. Kroninger, A. Bohannon, S. Hutchinson, and M. Egerstedt, "A resilient and energy-aware task allocation framework for heterogeneous multirobot systems," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 159–179, 2022.
- [11] D. Wu, G. Zeng, L. Meng, W. Zhou, and L. Li, "Gini coefficient-based task allocation for multi-robot systems with limited energy resources," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 155–168, 2018.
- [12] H. Ravichandar, K. Shaw, and S. Chernova, "STRATA: unified framework for task assignments in large teams of heterogeneous agents," *Auton. Agents and Multi-Agent Syst.*, vol. 34, no. 2, 2020, Art. no. 38.
- [13] A. Prorok, M. A. Hsieh, and V. Kumar, "The impact of diversity on optimal control policies for heterogeneous robot swarms," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 346–358, 2017.
- [14] F. Quinton, C. Grand, and C. Lesire, "Market approaches to the multi-robot task allocation problem: a survey," *J. Intell. Robotic Syst.*, vol. 107, no. 2, 2023, Art. no. 29.
- [15] L. E. Parker, "ALLIANCE: An architecture for fault tolerant multi-robot cooperation," *IEEE Trans. Robot. Autom.*, vol. 14, no. 2, pp. 220–240, 1998.
- [16] G. Notomista, S. Mayya, S. Hutchinson, and M. Egerstedt, "An optimal task allocation strategy for heterogeneous multi-robot systems," in *Proc. Eur. Control Conf.*, 2019, pp. 2071–2076.
- [17] G. Notomista, S. Mayya, M. Selvaggio, M. Santos, and C. Secchi, "A set-theoretic approach to multi-task execution and prioritization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 9873–9879.
- [18] M. Egerstedt, *Robot Ecology: Constraint-Based Design for Long-Duration Autonomy*. Princeton University Press, 2021.
- [19] A. D. Ames, G. Notomista, Y. Wardi, and M. Egerstedt, "Integral control barrier functions for dynamically defined control laws," *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 887–892, 2021.
- [20] G. Notomista and M. Egerstedt, "Constraint-driven coordinated control of multi-robot systems," in *Proc. Amer. Control Conf.*, 2019, pp. 1990–1996.