2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Extensive, Long-term Task and Motion Planning with Signal Temporal Logic Specification for Autonomous Construction

Mineto Satoh[1*], Rin Takano[1*] and Hiroyuki Oyama[1]

***Abstract*— We propose a hierarchical task and motion planning (TAMP) for autonomous construction that manipulates deformable objects, such as terrain excavation. The TAMP is required to generate an efficient task plan to meet high-level construction goals at sites with environmental diversity while ensuring motion feasibility. The difficulty, however, is to manipulate deformable objects containing nonlinear dynamics with a target given by a continuous value as the task specification. Optimization-based TAMP with signal temporal logic specifications in robotics is promising because of its continuous task specification and formulation as a nonlinear programming problem. The key to its application to extensive, long-term planning at real construction sites is a computationally efficient and stable formulation. We introduce a new expression for deformable objects with a simple differentiable function and a system model that can represent mode transitions based on machine action on the objects. This allows TAMP to be formulated as simultaneously selecting an action for objects and planning the motion to execute it. Furthermore, a hierarchical method that gives appropriate initial values is combined to improve optimality for large-scale nonlinear problems. From the verification by numerical experiments, the proposed method can generate a plan that minimizes the time to meet the task goal, even when the area is expanded.**

I. INTRODUCTION

Automated construction machinery, such as autonomous excavator systems [1], [2], [3], [4] is expected to contribute to work efficiency, construction accident prevention, and labor shortages through the application of robotics and automated driving technologies. Those autonomous machines generally need to be controlled to achieve the desired goals based on a higher-level construction plan, such as area or volume per unit time. While research [2],[5] is underway to make construction planning more efficient and precise with 3D models such as BIM (Building Information Modeling) and actual terrain data, there are still difficulties in autonomous control to satisfy the plans.

In the example of autonomous excavators, planning methods for meeting goals can be categorized into three approaches: fixed algorithms in the motion planning layer [3], [4], hierarchical rule-based task planning using abstracted excavation motion [2], [6], and data-driven, learning-based methods [7], [8]. The first approach involves repeated excavation and loading into a dump truck until a predetermined weight is reached [3], or repeated excavation using a state machine until the desired shape is achieved [4]. These methods cannot selectively perform multiple actions, such as moving to other locations, because the excavator body is stationary or limited to performing predetermined actions. Second, in the hierarchical approach [2], the 3D model of the terrain is partitioned and divided into layers based on excavator specifications to generate excavation paths to achieve changes in the terrain. The other [6] plans an excavation path that efficiently excavates within a certain area based on a complete coverage path planning algorithm and a cost function that considers the accessibility of dump trucks. Both methods assume that excavators operate according to an exact motion, making it difficult to reflect differences between planned and actual results or modifications to the motion plan in higher-level plans. Third, learning-based approaches show promise in addressing the above issues. A TaskNet planner [7] that decomposes high-level drilling objectives into subtasks can achieve the desired depth as target height. The two-stage method that integrates data-driven imitation learning and model-based trajectory optimization was proposed to generate trajectories to achieve desired excavation volumes [8]. However, these data-driven methods, whose performance depends on demonstration data, are concerned with application to construction sites with environmental diversity.

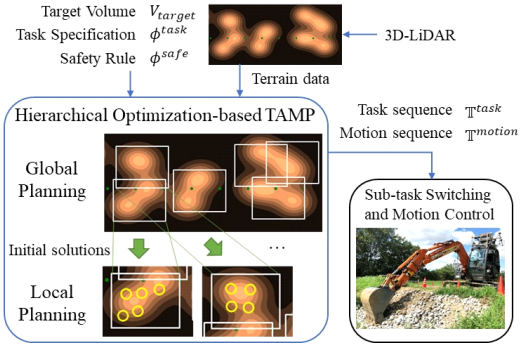

Fig. 1. System architecture for autonomous construction using hierarchical optimization-based TAMP.

Due to the limitations of the approaches described above, this paper focuses on the following two points. The first is a simultaneous task and motion planning (TAMP) method for deformable objects such as terrain. The key is modeling that allows for high-level planning (*i.e.,* task selection), considering the action of construction machinery on deformable objects. The second is considering the computational cost for application to extensive, long-term planning of real-world construction sites. The critical aspects related to compu-

[1]Data Science Laboratories, NEC Corporation, 1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, Japan, 211-8666 [mineto@nec.com]
*These authors contributed equally to this work

tational cost are the representation of deformable objects and the description of task specifications and nonlinear constraints on continuous variables.

In this paper, TAMP for robotic sequential manipulation [9] is extended for construction machinery to address the above challenges of handling deformable objects. Formal methods using temporal logic have been used to achieve desired states while satisfying various constraints [10]. In particular, methods based on signal temporal logic (STL) are suitable predicate logic forms for specifying properties of continuous signals. The key idea is to define a mode of system dynamics that focuses on the action on deformable objects and formulate it as a system model that simultaneously deals with the dynamics of machines and deformable objects. Then, task selection by switching this mode and motion planning are solved simultaneously as a nonlinear optimization problem to achieve an optimized TAMP. Note that while various approaches exist [11] for representing deformable objects, we propose a simple and computationally less demanding method suitable for optimization-based TAMP. Furthermore, a hierarchical method that gives appropriate initial values is combined to improve optimality for large-scale nonlinear problems. Fig. 1 shows an overview of the proposed system architecture for autonomous construction with hierarchical optimization-based TAMP. Given task specification and terrain data, global optimization first generates a plan for each local area that satisfies the objectives. Then, in each local optimization, the global optimization results are used as initial values to generate the final task and motion for autonomous operation. The paper has three main contributions:

- We propose a new expression for deformable objects with a simple differentiable function and a system model that can represent mode transitions based on machine actions on the objects.
- For earth excavation and loading tasks, we formulate the optimization-based TAMP with STL specifications and combine it with a hierarchical method.
- Numerical experimental results show that the proposed method minimizes the time needed to meet the task goal compared to methods that solved the task and motion separately, even when the domain is expanded.

II. RELATED WORK

A. Task and Motion Planning in Robotics

We focus on three aspects from recent TAMP research on robotics: how to plan order-dependent tasks, handle object-robot interactions, and handle deformable objects. The hierarchical method [12] integrates continuous geometric planning with enumerative task planning to plan long-horizon order-dependent tasks. Although the method uses partial symbolic descriptions, the enumeration of possible operations and the symbolic description of states make it difficult to apply to deformable objects. In the multi-layered approach with linear temporal logic (LTL) [13], an abstract task plan and a motion plan are repeated until a motion plan is successfully computed. Since the action of the robot is discretized to suppress state explosion in LTL planning, it is challenging to consider continuous and order-dependent actions on deformable objects. The optimization-based TAMP can be formulated as an optimal control problem for a hybrid system that focuses on the interaction with the object. The logic-geometric program (LGP) method [14] combines continuous motion planning and discrete task specification and is formulated as a mixed integer nonlinear programming problem (MINLP). The approach [15] is to approximate the robot dynamics with a low-dimensional mixed-logic dynamical system (MLD) and perform a TAMP considering both the system and the LTL constraints. However, MLD and linear constraints are unsuitable for dealing with constraints involving continuous variables, such as satisfying a target volume. Recent advances in computer graphics and machine learning are expected to overcome the challenges of the integral complexity of modeling, perception, and control in deformable object manipulation [11]. However, computational cost can be an issue in optimization with complex learned models. Thus, applying the above methods to order-dependent long-term planning of deformable objects raises the issue of computational efficiency. We solve this by introducing a simple representation of deformable objects and a system model that can represent mode transitions based on machine actions on the objects.

B. Planning for Robotic Excavation

We focus on three approaches of excavation planning that achieve a given goal: hierarchical rule-based, optimization-based, and data-driven. The hierarchical approaches [2], [6], [16], using rules or strategies based on abstracted excavation motion, allow for extensive and long-term planning, including excavator movement. However, such top-down task plans are limited to use in environments where motion feasibility is always high because they cannot consider the feasibility of the actual motion plan. The method [17] formulates the excavation task as a constrained optimization that achieves the target state of the terrain in the action space. Although a 2D occupancy grid-based soil model is used, extending the model to a large 3D area is computationally difficult due to iterative calculations. Another optimization-based approach [18] is a hierarchical combination of coarse and refined planning to avoid problems that require a large state space and complex interactions with soil. The method includes coarse planning based on current and target geometry and local optimization to satisfy geometric constraints such as reachability but does not include wide-area travel planning. The data-driven approach [7] can be categorized as replacing the traditional rule-based hierarchical approach with a learning-based one. Instead of solving the TAMP problem, the higher-level task is decomposed into subtasks and task primitives, each processed independently. The imitation learning-based method [8] plans with the excavator stationary and cannot generate plans with extensive area and long-term travel. In addition, these data-driven methods suffer from low generalizability because their performance is dependent on the

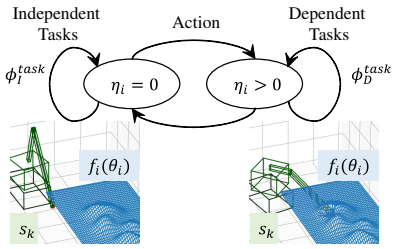

Fig. 2. Schematic of the relationship between the switch variable η and modes of the system described in Eq. (1) and Eq. (2). The state of the machine $s_{i,k}$ and the deformable object f_i in each mode are illustrated below.

training data. Since the methods described above do not solve the TAMP problem, it is difficult to optimize task selection and task specifications, considering motion. Furthermore, it is challenging to expand the area of application.

III. Modeling for Deformable Object Manipulation

A. Deformable Objects Representation

First, we describe the representation of the dynamics model for deformable objects. If the object to be manipulated is abstracted, the system model can be formulated to include the state of the object. However, for deformable objects such as terrain, complex dynamics models such as grid-based or learning-based models are not applicable to optimization-based TAMP in terms of computational complexity. Therefore, we propose a method to represent the surface shape of deformable objects with a simple differentiable function, focusing on the shape change caused by machine action. Further, we introduce switch variable $0 \leq \eta \leq 1$ to represent two modes: a state in which the machine does not act on the object (defined as an independent task) and a state in which it acts on the deformable object to change its shape (defined as a dependent task). In the following, we consider a discrete bounded time series $\tau^{task} := \{t_i \mid i = 0, \ldots, m\}$ representing the task phase. Let f_i be a function of the surface shape of a deformable object at time t_i, *i.e.,* the height $z = f_i(x, y)$. The parameter θ_i is then introduced to characterize the change in shape f_i due to machine action (*e.g.,* coordinates of the action point, etc.). Assuming that excavation is an action as an example of deformable object manipulation, the update of the function f can be expressed as follows:

$$f_{i+1} = f_i + \eta_i w(\theta_i)\left(h(\theta_i) - f_i\right), \tag{1}$$

where w is a window function representing the region of the machine action, h expresses the shape updated by the machine action, and η is the switch variable. Eq. (1) represents a geometric approximation of the terrain shape removed by excavation when the machine action is excavation.

B. System Model for Task Switching

Next, we describe the system model for machines that can represent mode transitions based on action on deformable objects, which is an extension of TAMP for robotic manipulation [9]. Let each task phase i consist of discrete bounded time series representing motion step $\tau_i^{motion} := \{t_{i,k} \mid k = 0, \ldots, n\}$. We introduce $g(s_{i,k}, \theta_i^{con}) \geq 0$ as a constraint that relates the state of the machine to the update of the shape function expressed in Eq. (1). Here, $\theta_i^{con} \in \theta_i$ represents the parameters involved in the constraints in θ_i. This is because each state is not independent of the other when the machine and the object are in contact. The details are described in Section IV. The system model can be written as follows:

$$s_{i,k+1} = s_{i,k} + u_{i,k}\left(t_{i,k+1} - t_{i,k}\right), \tag{2}$$

$$\eta_i\, g\left(s_{i,k}, \theta_i^{con}\right) \geq 0, \tag{3}$$

where $s_{i,k} \in \mathbb{R}^N$ is a state vector that consists of states of machines, and $u_{i,k} \in \mathbb{R}^N$ is a control input vector. Note that Eq. (3) is a constraint that relates the state of the machine to the state of the deformable object, and this constraint can only be active when the variable η_i, representing the modes of the system, satisfies the specific condition. Specifically, the constraint $g\left(s_{i,k}, \theta_i^{con}\right) \geq 0$ is not required when $\eta_i = 0$ holds (*i.e.,* independent task ϕ_I^{task}) and is required when $\eta_i > 0$ holds (*i.e.,* dependent task ϕ_D^{task}).

Fig. 2 shows the relationship between the switch variable η and modes of the system described in Eq. (1) and Eq. (2). The mode here does not represent a switch in the hybrid system described in [9], but rather a switch between interacting with deformable objects or not. As an example, Fig. 2 illustrates the following two modes of tasks performed by an excavator. In mode $\eta = 0$, the excavator does not act on soil and performs independent tasks (*e.g.,* moving), and in mode $\eta > 0$, it performs dependent tasks that act on soil (*e.g.,* digging). This indicates that the task to be performed by the excavator can be selected by switching modes, *i.e.,* by changing the value of the switch variable. Thus, task planning can be done by generating time series values, *i.e.,* sequences of switch variables η_i.

C. STL for Task Specification

Finally, we describe an approach that uses STL to specify tasks based on the above system model. STL is a rich predicate logic formalism for specifying continuous signal properties and determining whether a signal at a given time satisfies certain conditions [19]. It is suitable for specifying continuous tasks on deformable objects because it allows for specifying constraints of time-dependent continuous quantities. The robustness degree determines whether the given signal σ at time t_k satisfies a formula ϕ, which is satisfied when $\rho^{\phi}(\sigma, t_k) > 0$ holds. In the example shown in Fig. 2, the constraints on task selection by switch variables can be described using STL as follows. First, the STL formula ϕ_0 and ϕ_1 that the switch variables satisfy the mode $\eta = 0$ or $\eta > 0$ can be expressed as follows, respectively:

$$\begin{aligned} \phi_0 &:= \{-\eta + \varepsilon > 0\}, \\ \phi_1 &:= \{\eta - \varepsilon > 0\}, \end{aligned} \tag{4}$$

where, $0 \leq \eta \leq 1$ and ε is a positive small constant. Let ϕ_I^{task} be the condition for an independent task in mode $\eta = 0$ and ϕ_D^{task} be the condition for a dependent task in mode $\eta > 0$.

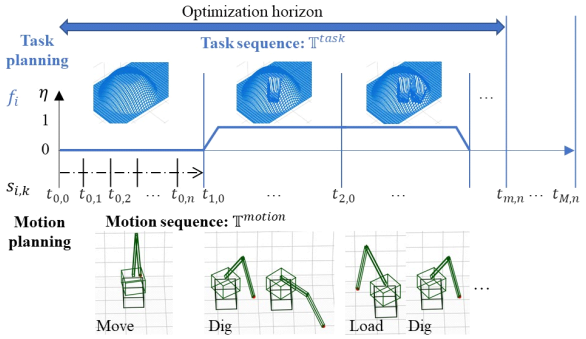

Fig. 3. Schematic diagram of the task and motion planning sequence and optimization horizon $t_{m,n}$. Time step $t_{i,k}$ represents task phase i, motion step k, and an example of the change in surface shape f_i and state $s_{i,k}$ when the switch variable η_i changes from 0 to 1 is illustrated. $t_{M,n}$ is the final time step after repeated optimization of the local area shown in Algorithm 1.

The condition for either of these to hold can be expressed as follows:

$$\phi^{task} = \left(\phi_0 \wedge \phi_I^{task}\right) \vee \left(\phi_1 \wedge \phi_D^{task}\right). \tag{5}$$

Then, the robustness degree is defined as follows:

$$\begin{aligned} \rho^{\phi_A} \equiv \rho^{\phi_0 \wedge \phi_I^{task}} &:= \min\left(\rho^{\phi_0}, \rho^{\phi_I^{task}}\right), \\ \rho^{\phi_B} \equiv \rho^{\phi_1 \wedge \phi_D^{task}} &:= \min\left(\rho^{\phi_1}, \rho^{\phi_D^{task}}\right), \\ \rho^{\phi^{task}} = \rho^{\phi_A \vee \phi_B} &:= \max\left(\rho^{\phi_A}, \rho^{\phi_B}\right). \end{aligned} \tag{6}$$

Additionally, the finally operator $\mathbf{F}_{[a,b]}$ and the globally operator $\mathbf{G}_{[a,b]}$ are defined as follows:

$$\rho^{\mathbf{F}_{[a,b]}\phi}(\sigma, t_k) := \max_{t' \in [t_k+a, t_k+b]} \rho^{\phi}(\sigma, t'), \tag{7}$$

$$\rho^{\mathbf{G}_{[a,b]}\phi}(\sigma, t_k) := \min_{t' \in [t_k+a, t_k+b]} \rho^{\phi}(\sigma, t'). \tag{8}$$

The smooth approximation of the robustness degree for STL, which has focused on the field of control theory [20], is then applied to eliminate the inclusion of min and max operations that make it non-differentiable. In addition to accomplishing the above tasks, restrictive conditions, such as safety rules, can be added according to the environment and task. With the above system model and task specification as constraints, we can solve the optimal control problems for TAMP in the form of a nonlinear programming problem (NLP).

IV. TAMP for Excavation and Loading Task

We formulate the optimization-based TAMP with STL specifications for earth excavation and loading tasks using the system model and task specification in Section III.

A. Definition of Optimization Problem

We focus on excavator tasks that alter terrain shape, such as trenching, excavation, and material relocation, and treat the shape changed by the action of the excavator as a deformable object. In the following, let f in Eq. (1) represent the surface profile of the terrain and be a differentiable function. The switch variable η introduced in Section III selects the excavator task as either moving as an independent task ϕ_I^{task} or loading as a dependent task ϕ_D^{task}. Fig. 3 shows the schematic diagram of the task and motion planning sequence and optimization horizon. The important point is to separate the time series representing the state of the machines $s_{i,k}$ in Eq. (2) from the time series representing the state of the terrain f_i in Eq. (1). Thus, the time vector of the horizon length m phases can be defined as:

$$\mathbb{T}_{0:m}^{time} = [t_{0,0}, \ldots, t_{0,n}, \ldots, t_{m,0}, \ldots, t_{m,n}]. \tag{9}$$

Therefore, Fig. 3 denotes the terrain f that changes with task sequence τ^{task} as f_i and the machine states s that changes with the motion sequence τ_i^{motion} of each task phase i as $s_{i,k}$. From Eq. (1), the elements that change the terrain are the switch variable η and the parameter θ that characterizes the terrain update, so we define task planning as determining the following sequences $\mathbb{T}^{task}$:

$$\mathbb{T}^{task} = \left[\left(\eta_1, \theta_1^T\right), \ldots, \left(\eta_m, \theta_m^T\right)\right]. \tag{10}$$

Fig. 3 illustrates the case where the switch variable is $\eta_i = 0$ or $\eta_i > 0$ in each phase i, and the terrain shape changes when $\eta_i > 0$ is satisfied. Let V_i be the excavation volume in the task phase i and define the volume sequence as follows:

$$\mathbb{V} = [V_1, \ldots, V_m]. \tag{11}$$

From Eq. (2), we define motion planning at task phase i as determining the following sequences $\mathbb{T}^{motion}$:

$$\mathbb{T}^{motion} = \left[\left({s_1}^T, {u_1}^T\right), \ldots, \left({s_n}^T, {u_n}^T\right)\right]. \tag{12}$$

Finally, from Eq. (9) – (12), the optimization vector is defined as $Z = [\mathbb{T}^{motion}, \mathbb{T}^{task}, \mathbb{T}_{0:m}^{time}, \mathbb{V}]^T$. The definition of the optimization problem is to obtain an optimal solution of vector Z under a given task specification ϕ^{task} and terrain data. The constraints and evaluation functions of the optimization problem are described in detail below.

B. Constraints for Excavation and Loading

1) Kinematics and System Model: Fig. 4 schematically illustrates the excavator kinematics and model of excavating the terrain and loading the excavated material onto a dump truck. For simplicity, the movement of the bucket is omitted. The position and angle states of the excavator $q^{BH} \in s$ and the control inputs $u^{BH} \in u$ are defined as follows:

$$q^{BH} = \begin{pmatrix} p_x^B \\ p_y^B \\ \theta_z^B \\ \theta_{turn} \\ \theta_{boom} \\ \theta_{arm} \end{pmatrix}, \quad u^{BH} = \begin{pmatrix} \cos(\theta_z^B) & 0 \\ \sin(\theta_z^B) & \\ 0 & \mathbf{I}_4 \end{pmatrix} \begin{pmatrix} v \\ \omega_z \\ \omega_{turn} \\ \omega_{boom} \\ \omega_{arm} \end{pmatrix}. \tag{13}$$

We define the position of the excavator as $\left(p_x^B, p_y^B\right)$ in the global coordinate system and the yaw angle on the xy-plane as θ_z^B. The joint angles of the excavator $\theta_{turn}, \theta_{boom}$, and θ_{arm} are represented by the excavator's coordinate system. Note that the control input in Eq. (13) is $u^{BH} \in \mathbb{R}^6$, but the actual control input is defined as $u_{input}^{BH} =$

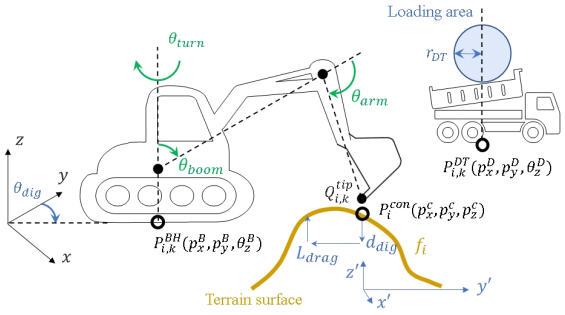

Fig. 4. Schematic diagram of the excavator kinematics and model of excavating terrain and loading dump trucks. The global coordinates are noted in black, the excavator's relative coordinates are green, and the excavation coordinates with respect to the contact point are blue symbols.

$[v, \omega_z, \omega_{turn}, \omega_{boom}, \omega_{arm}]^T \in \mathbb{R}^5$. Similarly, the state of the dump truck and control inputs are defined as $q^{DT} = \left[p^D_x, p^D_y, \theta^D_z\right]^T \in \mathbb{R}^3$ and $u^{DT}_{input} = [v, \omega_z]^T \in \mathbb{R}^2$, respectively. Then, s and u in Eq. (2) are expressed as follows:

$$s = \left[q^{BH\,T}, q^{DT\,T}\right]^T, \quad u = \left[u^{BH\,T}, u^{DT\,T}\right]^T. \tag{14}$$

For simplicity, the loading area on the dump is defined as within a radius r_{DT} with respect to the position $\left(p^D_x, p^D_y\right)$.

Fig. 4 also shows the contact point $P^{con}_i\left(p^c_x, p^c_y, p^c_z\right)$ with the terrain surface f_i of phase i and a cross-sectional view of the excavation trajectory in the excavation coordinate system $x'y'z'$, represented by the rotation angle θ_{dig} from the global coordinate system. Based on the definition of the function f that represents the terrain surface, the z-coordinate of the contact point can be calculated as $p^c_z = f(p^c_x, p^c_y)$. Here, d_{dig} and L_{drag} represent the insertion depth and the bucket pull length from the contact point P^{con}_i, respectively, and are included in the digging parameter θ_i in Eq. (1). On the other hand, θ^{con}_i, which represents the parameters involved in the constraints, can be specifically defined as:

$$\theta^{con}_i = [p^c_x, p^c_y, \theta_{dig}]^T. \tag{15}$$

Then, the specific expression of the constraint $g\left(s_{i,k}, \theta^{con}_i\right)$ that relates the excavator to the terrain in Eq. (3) can be expressed as follows:

$$g\left(q^{BH}_{i,k}, \theta^{con}_i\right) = -\left\| \begin{pmatrix} Q^{tip}_{i,k} \\ \theta_{tip} \end{pmatrix} - \begin{pmatrix} P^{con}_i \\ \theta_{dig} \end{pmatrix} \right\|^2 + \varepsilon_d, \tag{16}$$

where ε_d is a positive small constant. $Q^{tip}_{i,k}$ and θ_{tip} represent the coordinates and angle of the excavator tip (*i.e.,* bucket) in contact with the deformable object, computed from the machine state $q^{BH}_{i,k}$. Also, P^{con}_i and θ_{dig} are obtained as a parameter of Eq. (15).

2) Specification for task: As a dependent task ϕ^{task}_D, define STL formula for excavating terrain and loading it into a dump truck as follows. The goal condition of a task phase i can be defined as the excavation volume of phase V_{i-1} being positive and the tip of the excavator $Q^{tip}_{i,k}$ reaching the loading area (*i.e.,* within radius r_{DT} of the dump location $P^{DT}_{i,k}$) of the dump during phase i, each of which can be written using the STL formula as follows:

$$\phi_v := \{V_{i-1} - \varepsilon_V > 0\}, \tag{17}$$

$$\begin{aligned} \phi_{load} &= \mathbf{F}_{[t_{i,0}, t_{i,n}]}\left(\mu^{load}_{i,k}\right), \\ \mu^{load}_{i,k} &:= \{-\eta_i \left\| Q^{tip}_{i,k} - P^{DT}_{i,k} \right\|^2 + r^2_{DT} > 0\}, \end{aligned} \tag{18}$$

where ε_V is a positive small constant and the finally operator $\mathbf{F}_{[a,b]}$ can be calculated by Eq. (7). Then, by applying smooth approximations of the robustness degree, the dependent task ϕ^{task}_D in Eq. (6) can be expressed as:

$$\tilde{\rho}^{\phi^{task}_D} = \tilde{\rho}^{\phi_v \wedge \phi_{load}}, \tag{19}$$

where smooth STL robustness degree is noted as $\tilde{\rho}^{\phi}$. In the following, we assume that the independent task ϕ^{task}_I has no specification, *i.e.,* the overall task $\tilde{\rho}^{\phi^{task}} = \tilde{\rho}^{\phi^{task}_D}$.

3) Other Constraints: First, other constraints require a relationship between the volume vector $\mathbb{V}$ and the shape change of the terrain. The excavation volume V_i for each task phase i must be equal to the integral of the change in terrain state $f_i - f_{i-1}$ over the excavation area. Therefore, the following equality constraint must hold:

$$V_i - \int_{-w_{buc}/2}^{w_{buc}/2} \int_{-L_{drag}}^{0} (f_i - f_{i-1})\, dxdy = 0, \quad (i = 1, \ldots, m) \tag{20}$$

where w_{buc} is the width of the excavator bucket. The actual integral calculations were performed by numerical integration using the Gaussian quadrature method [21]. Further, from the condition that the excavation volume V_i (*i.e.,* one excavation volume) is less than or equal to the maximum volume V_{max} based on the excavator bucket size, the following inequality constraint is required:

$$V_i \leq V_{max}, \quad (i = 1, \ldots, m). \tag{21}$$

Next, safety conditions other than task completion are defined for practicality. Let $P^{DT}_{1,0}$ be the position of the dump truck at the time of loading and $t_{m,n}$ be the time it takes for one dump truck to finish loading. The condition that the dump trucks do not move during that time is expressed in the STL formula as follows:

$$\begin{aligned} \phi_{dump} &= \mathbf{G}_{[t_{1,0}, t_{m,n}]}\left(\mu^{dump}_{i,k}\right), \\ \mu^{dump}_{i,k} &:= \{-\| P^{DT}_{i,k} - P^{DT}_{1,0} \|^2 + \varepsilon^2_r > 0\}, \end{aligned} \tag{22}$$

where ε_r is a positive small constant and the globally operator $\mathbf{G}_{[a,b]}$ can be calculated by Eq. (8).

The conditions for avoiding collisions between the excavator and the dump truck can also be described as follows:

$$\begin{aligned} \phi_{col} &= \mathbf{G}_{[t_{0,0}, t_{m,n}]}\left(\mu^{col}_{i,k}\right), \\ \mu^{col}_{i,k} &:= \{-\| P^{BH}_{i,k} - P^{DT}_{i,k} \|^2 + L^2_{col} > 0\}, \end{aligned} \tag{23}$$

where L_{col} represents the distance to avoid collision, considering the size of both machines. Other constraints, such as no-entry areas, can be set. Finally, the STL formula ϕ^{safe}

that satisfies all safety rules is defined by combining all conditions using the operator $\wedge$. Then, the smooth robustness degree can be expressed as:

$$\tilde{\rho}^{\phi^{safe}} = \tilde{\rho}^{\phi_{dump} \wedge \phi_{col}}. \tag{24}$$

C. Formulation as an Optimal Control Problem

The TAMP problem can be modeled as a finite horizon optimal control problem, satisfying defined robustness degrees under the proposed dynamics in Eq. (1) (2). This optimal control problem with phase $i = 0, \ldots, m$ as a horizon can be formulated as the following NLP:

$$\begin{aligned} OP_{0:m}: \quad & \min_{Z} \quad J(Z) \\ & s.t. \quad Eq.(1),(2),(3),(13),(14),(15),(16),(20),(21), \\ & \mathbb{T}^{time}_{i,k} \in \mathscr{T}^{time},\ \mathbb{T}^{motion}_{k} \in \mathscr{T}^{motion},\ \forall k = 1, \ldots, n, \\ & \mathbb{T}^{task}_{i} \in \mathscr{T}^{task},\ \mathbb{V}_i \in \mathbb{R}_{\geq 0},\ \forall i = 1, \ldots, m, \\ & \tilde{\rho}^{\phi^{task}} \geq s_{\rho_1},\ \tilde{\rho}^{\phi^{safe}} \geq s_{\rho_2},\ s_{\rho} \in \mathbb{R}^2_{\geq 0}, \end{aligned} \tag{25}$$

where Z is an optimization vector defined as $Z = [\mathbb{T}^{motion}, \mathbb{T}^{task}, \mathbb{T}^{time}_{0:m}, \mathbb{V}]$, and additional slack variables $s_{\rho} = [s_{\rho_1}, s_{\rho_2}]^T \in \mathbb{R}^2_{\geq 0}$ for determining the lower bounds of the robustness degrees. The objective function J is as follows:

$$\begin{aligned} & J\left(\mathbb{T}^{motion}, \mathbb{T}^{task}, \mathbb{T}^{time}_{0:m}, \mathbb{V}\right) := \\ & \alpha_u \sum_{i=0}^{m} \sum_{k=0}^{n} \left[{u^{BH}_{input}}^T Q^{BH}_u u^{BH}_{input} + {u^{DT}_{input}}^T Q^{DT}_u u^{DT}_{input} \right] \\ & + \alpha_t t_{m,n} + \alpha_V \left(\sum_{i=0}^{m} V_i - V_{target} \right)^2 + \alpha_\eta \sum_{i=0}^{m} \eta_i, \end{aligned} \tag{26}$$

where $\alpha_u, \alpha_t, \alpha_V, \alpha_\eta$ are the weight coefficient, V_{target} is the target volume, and $t_{m,n}$ is the last time step of the optimization horizon. $Q^{BH}_u \in \mathbb{R}^{5\times5}$ and $Q^{DT}_u \in \mathbb{R}^{2\times2}$ are positive definite weighting matrices for u^{BH}_{input} and u^{DT}_{input} ,respectively. The first term of this evaluation function J represents the cost of the control input, the second term represents the cost of the time required, the third term represents the cost of meeting the target volume, and the fourth term represents the penalty of the switch variable. The switch variable penalty is imposed so that the minimum number of dependent tasks ϕ^{task}_D is selected to achieve the task goal. The optimization of Eq. (25) yields a sequence of tasks and motions with a horizon length of 1 to m phases described in Eq. (9), *i.e.,* the period during which the motion of n steps in Eq. (12) is repeated m times. Note that the switch variables are continuous rather than binary because solving MINLP under nonlinear and continuous constraints is difficult.

D. Hierarchical Optimization

We propose a hierarchical optimization method to solve the initial solution dependence problem [22] and achieve extensive and long-term planning. In general, for nonlinear optimal control problems, the optimal solution can be obtained with a small number of iterations by providing initial values close to the steady-state values. In practice, the initial value of the motion vector $\mathbb{T}^{motion}$ in Eq. (12) can adapt the information from the actual machine. However, the contact points P^{con}_i in the task vector $\mathbb{T}^{task}$ in Eq. (10) become more challenging to determine as the target area becomes larger. Therefore, the optimization process is divided into two layers: global optimization to achieve the target volume V_{target} for the entire map using the local area as the one excavation unit, and local optimization to achieve the target volume of the local area using the bucket size as the one excavation unit. The hierarchical planning idea is similar to the methods in [2], [6], [16]. However, the key is that each hierarchy optimizes task and motion at a different granularity, and the solution obtained in the upper layer is used as the initial solution for the optimization in the lower layer.

Algorithm 1 Hierarchical Optimization Algorithm

Input: An initial global vector Z_0, terrain data, target volume V_{target}, local area size S_l and maximum divisions m_g, bucket size S_b and maximum excavation m_l.

Output: Optimal local vectors Z^{local}_j for each local area j

```
Generate surface function f from terrain data.
Global: Z = OP_{0:m}(Z_0, V_target, S_l, m_g)
Determine local area (1,...,N_local) satisfing η > 0.
Set initial local solution for P^con from global Z.
for j = 1,...,N_local do
    Local: Z^local_j = OP_{m(j-1):mj}(Z_j, V_j, S_b, m_l)
```

In more detail, let Z be the global optimization vector and Z^{local}_j be the local optimization vector for each local region j. In global optimization, the unit excavation area is set to the local area size S_l larger than the excavator bucket size S_b. Then, find the optimal solution of Z according to Eq. (25) for the entire global map with the given maximum number of partitions m_g as the length of the optimization horizon $t_{m_g,n}$. Note that the switch variable η, included in the optimal solution of Z as the task sequence $\mathbb{T}^{task}$ in Eq. (10), represents whether or not to excavate for each local area. As a result, the index of the local area satisfying $\eta > 0$ can be set to $j = 1 \ldots N_{local}$ as the target area for local optimization. Here, N_{local} is the number of the local area and satisfies $N_{local} \leq m_g$. In local optimization, the unit excavation area is set to the bucket size S_b, and each $P^{con} \in \theta$, included in the global optimal solution of Z as the task sequence $\mathbb{T}^{task}$, is set to the initial solution for each local contact point P^{con}_j in Z^{local}_j. Then, for each local area $j\ (= 1 \ldots N_{local})$, find the optimal solution for Z^{local}_j according to Eq. (25), with the maximum number of excavations m_l as the optimization horizon $t_{m_l,n}$. This procedure is summarized in Algorithm 1.

V. Results and Validation

Numerical experiments were performed using randomly generated surface functions as global maps. Our proposed optimization problem was modeled with CasADi [23] and solved iteratively by IPOPT [24].

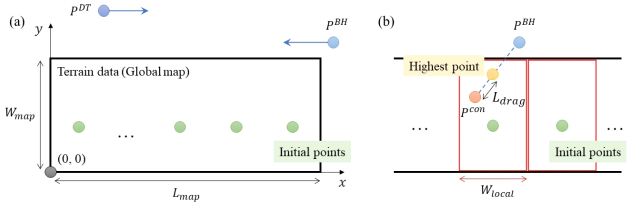

Fig. 5. (a) Schematic diagram of the arrangement in the xy-plane in the numerical experiment. (b) Contact point determination method in the comparative method where tasks and motions are planned separately.

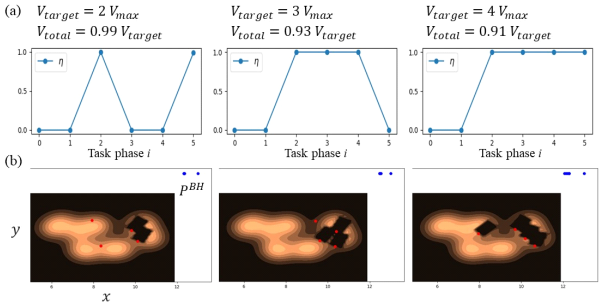

Fig. 6. Example results of (a) the switch variable values for each task phase and (b) final (*i.e.,* cumulative) contour maps of the terrain surface.

A. Verification of Task Switching

First, task selection by switch variables was verified by numerical simulation. The global map of the terrain was generated virtually by the superposition of radial basis functions (RBF), and the following equation was used to update the surface function in Eq. (1):

$$f_{i+1} = f_i + \eta_j w_x(\theta_i) w_y(\theta_i) \left(h(\theta_i) - f_i\right), \tag{27}$$

where w_x, w_y, and h are window functions that approximate the excavation area and the shape function that is updated by excavation, respectively, as follows:

$$\begin{aligned} w_x &= \left(\sigma_{w_k}(x - p_x^c + w_{buc}/2) - \sigma_{w_k}(x - p_x^c - w_{buc}/2)\right), \\ w_y &= \left(\sigma_{w_k}(y - p_y^c + L_{drag}) - \sigma_{w_k}(y - p_y^c)\right), \\ h &= p_z^c - d_{dig}. \end{aligned} \tag{28}$$

Note that Eq. (28) is intended to represent the geometric change in terrain shape due to excavation by a simple differentiable function. In the following, L_{drag} and d_{dig} are set to fixed values 1.0 m and 0.4 m for simplicity, although they can be included in the digging parameter θ and determined by optimization. $\sigma_{w_k}(x)$ express sigmoid function defined as $\sigma_{w_k}(x) = \frac{1}{1+e^{-w_k(x)}}$, and (p_x^c, p_y^c) are the x and y coordinates of the contact point P^{con} obtained as digging parameter θ in optimal vector Z. Fig. 5 (a) shows the arrangement in the xy-plane of the numerical experiment. A 7-tonne excavator (boom length 3.6 m, arm length 2.6 m) was assumed, and the initial position was set to (13.0, 7.2). The initial position of the dump truck is (2.0, 13.0). For simplicity, the control inputs were set to values common to excavator and dump, translation $v = 0.86\ m/s$, and rotation $\omega_z = 0.11\ rad/s$ as the upper limit. The excavator swivel and the angular velocity of each joint were also commonly set to an upper limit of 1.1 rad/s. The global map size was set to $L_{map} = 12\ m$, $W_{map} = 6\ m$, and the terrain data was generated using the RBF base number of 14. For local optimization, the excavator bucket width was set to 0.7 m, and the bucket size to $S_b = wL_{drag} = 0.7\ m^2$, and the maximum volume to $V_{max} = 0.28\ m^3$. For global optimization, the local area size was set to $S_l = 5 \times wL_{drag} = 3.5\ m^2$, and the maximum volume to $V_{max} = 1.0\ m^3$. The initial contact points during global optimization were equally spaced according to the maximum number of local areas m_g.

Fig. 6 shows example results of (a) the switch variable values for each task phase ($i = 1, \ldots, 5$) and (b) final (*i.e.,* cumulative) contour maps of the terrain surface in one optimization horizon. The target volume V_{target} is an integer multiple (2 to 4) of the bucket maximum volume V_{max}. Each graph in Fig. 6 (a) shows that the task phase i satisfying $\eta > 1$ is discretely determined depending on the given target volume. In Fig. 6 (b), the blue dot represents the position of the excavator P^{BH}. The first two phases of each task phase i are assigned to the excavator and dump movements (*i.e.,* independent task ϕ_I^{task}), thus imposing the constraint $\eta = 0$. In the subsequent phases, $\eta > 0$ is satisfied so that the minimum number of excavations (*i.e.,* dependent tasks ϕ_D^{task}) is required to meet the target volume. The total excavation volume V_{total} is expressed as a rate of the target volume. These results confirm that it is possible to discretely select dependent or independent tasks with the switch variable η depending on the continuous target volume.

B. Verification of Extensive and Long-term Planning

A comparison is shown between the proposed method (TAMP) and the method of planning tasks and motions separately, such as [16] (denoted as "Separate"). The RBF basis number was set to 30 in the same environment as in Fig. 5 (a) to cover a wider area. The method of "Separate" was based on the same system model as the proposed method, with the contact point P^{con} set at L_{drag} offset from the highest point on the terrain as shown in Fig. 5 (b). The width of the local area was set to $W_{local} = 2\ m$. Only the local optimization in Eq. (25) was performed except for the task vector $\mathbb{T}^{task}$. The number of excavations in each local area was fixed at $m_l = 5$, the same value as the upper limit of the TAMP.

Fig. 7 (a) shows a scatter plot of the total excavation volume V_{total} and the task duration T_{total} for each method. Note that the task duration T_{total} is expressed as the final time step $t_{M,n}$ after repeated optimization of the local area divided by the time required to excavate 1 m^3 (defined as T_{1m^3}), *i.e.,* the time required per unit volume (1 m^3). Four target volumes ($V_{target} = 4, 5, 6, 7\ m^3$) were given in the TAMP, and terrain data were generated 10 times, each with a different random seed. Since the "Separate" method is not a planning method that gives a target volume, the total number of excavations (*i.e.,* total volume) was varied by varying the number of local area divisions from 5 to 12. Results show that our TAMP method has a linear relationship between total

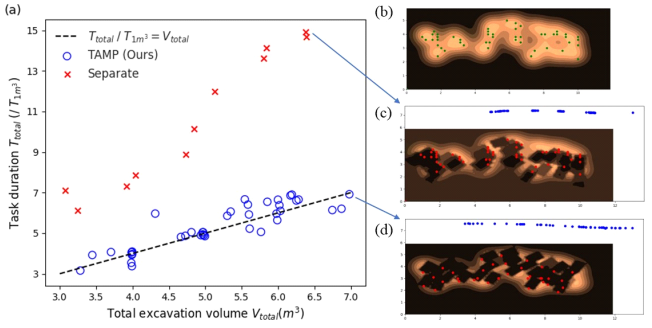

Fig. 7. (a) Results of comparison of total excavation volume V_{total} and task duration T_{total} for the proposed (TAMP) and the "Separate" methods, (b) map before excavation, (c) and (d) show the excavation results of the "Separate" and TAMP methods, respectively.

TABLE I

RESULT OF THE MEAN AND STANDARD DEVIATION (SD) OF THE TOTAL EXCAVATION VOLUME RELATIVE TO THE TARGET VOLUME FOR TAMP.

$V_{target}(m^3)$	4.0	5.0	6.0	7.0
$Mean(m^3)$	3.84	4.83	5.66	6.35
$Mean/V_{target}$ (%)	95.9	96.6	94.3	90.7
$SD(m^3)$	0.267	0.217	0.238	0.370

volume and task duration, *i.e.,* the increase in task duration concerning the expansion of the excavation area is minimal. This is because the minimum movement, excavation location, and number of excavations were planned to meet the target volume. Fig. 7 (b) shows the map before excavation, and (c) and (d) show the example results of excavation by "Separate" and TAMP methods, respectively. Table I shows the results of the mean and standard deviation (SD) of the total excavation volume relative to the target volume for TAMP method. The results show that more than 90% of the target volume is achieved despite expanding the area.

VI. CONCLUSIONS

We proposed a hierarchical optimization-based TAMP for autonomous construction. The key was an expression for deformable objects with a simple differentiable function and a system model that can represent mode transitions based on machine action. The hierarchical method that gives appropriate initial values was combined to improve optimality for large-scale nonlinear problems. Numerical experiments showed that the proposed method minimized the time needed to meet the task goal compared to the method that solved the task and motion separately, even when the area was expanded. The system shown in Fig. 1 is being validated with a real excavator, and its application to various environments and comparison of work efficiency is future work.

REFERENCES

[1] L. Zhang, J. Zhao, P. Long, L. Wang, L. Qian, F. Lu, X. Song, and D. Manocha, "An autonomous excavator system for material loading tasks," *Science Robotics*, vol. 6, no. 55, p. eabc3164, 2021.

[2] J. Seo, S. Lee, J. Kim, and S.-K. Kim, "Task planner design for an automated excavation system," *Automation in Construction*, vol. 20, no. 7, pp. 954–966, 2011.

[3] H. Yoshida, T. Yoshimoto, D. Umino, and N. Mori, "Practical full automation of excavation and loading for hydraulic excavators in indoor environments," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 2153–2160.

[4] D. Jud, G. Hottiger, P. Leemann, and M. Hutter, "Planning and control for autonomous excavation," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2151–2158, 2017.

[5] J. Kim, S. Lee, J. Seo, D.-E. Lee, and H. S. Choi, "The integration of earthwork design review and planning using uav-based point cloud and bim," *Applied Sciences*, vol. 11, no. 8, p. 3435, 2021.

[6] J. Kim, D.-e. Lee, and J. Seo, "Task planning strategy and path similarity analysis for an autonomous excavator," *Automation in Construction*, vol. 112, p. 103108, 2020.

[7] J. Zhao and L. Zhang, "Tasknet: A neural task planner for autonomous excavator," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 2220–2226.

[8] Q. Guo, Z. Ye, L. Wang, and L. Zhang, "Imitation learning and model integrated excavator trajectory planning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 5737–5743.

[9] R. Takano, H. Oyama, and M. Yamakita, "Continuous optimization-based task and motion planning with signal temporal logic specifications for sequential manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8409–8415.

[10] C. Belta, B. Yordanov, and E. A. Gol, *Formal methods for discrete-time dynamical systems*. Springer, 2017, vol. 89.

[11] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.

[12] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1470–1477.

[13] K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Towards manipulation planning with temporal logic specifications," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 346–352.

[14] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning." in *IJCAI*, 2015, pp. 1930–1936.

[15] M. Katayama, S. Tokuda, M. Yamakita, and H. Oyama, "Fast ltl-based flexible planning for dual-arm manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6605–6612.

[16] L. Wang, Z. Ye, and L. Zhang, "Hierarchical planning for autonomous excavator on material loading tasks," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 38. IAARC Publications, 2021, pp. 827–834.

[17] S. Singh and R. G. Simmons, "Task planning for robotic excavation." in *IROS*, vol. 92, 1992, pp. 1284–1291.

[18] A. Stentz, J. Bares, S. Singh, and P. Rowe, "A robotic excavator for autonomous truck loading," *Autonomous Robots*, vol. 7, pp. 175–186, 1999.

[19] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*. Springer, 2004, pp. 152–166.

[20] Y. Gilpin, V. Kurtz, and H. Lin, "A smooth robustness measure of signal temporal logic for symbolic control," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 241–246, 2020.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN 77, 2nd Ed.* Cambridge University Press, 2003.

[22] L. T. Biegler, "Efficient solution of dynamic optimization and nmpc problems," in *Nonlinear model predictive control*. Springer, 2000, pp. 219–243.

[23] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[24] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.