

Priority-Based Switching Model Predictive Control for Sequential Manipulation of Space Robot

Ziran Liu

Harbin Institute of Technology, Harbin, China

Zijian Dai

Harbin Institute of Technology Shenzhen, Shenzhen, China

Chengfei Yue

Harbin Institute of Technology Shenzhen, Shenzhen, China

Tao Lin

Harbin Institute of Technology, Harbin, China

Antonella Ferrara, Fellow, IEEE

University of Pavia, Pavia, Italy

Xibin Cao

Harbin Institute of Technology, Harbin, China

Abstract— In this paper, a priority-based switching model predictive control (SMPC) method is proposed for space robots to execute sequential operation tasks efficiently. For a predefined series of subtasks, we develop a state-dependent SMPC approach that simultaneously determines control inputs and switching times. By incorporating switching within the prediction horizon, the method improves overall task efficiency and smooth transitions. However, it may impact individual subtask performance due to potential conflicts between objectives. To address this, a soft-priority concept modulates the importance of each stage within the SMPC framework, offering a balanced and practical solution. Numerical simulations verify the method's effectiveness and analyze how preference-based priorities influence overall task performance.

Manuscript received XXXXX 00, 0000; revised XXXXX 00, 0000; accepted XXXXX 00, 0000.

This work was supported by National Natural Science Foundation of China(12372045), Guangdong Basic and Applied Basic Research Foundation(2023B1515120018), Shenzhen Science and Technology Program (JCYJ20220818102207015) and China Scholarship Council. (*Corresponding author:* Chengfei Yue).

Author's addresses: Ziran Liu, Tao Lin and Xibin Cao are with the Research Center of Satellite Technology, Harbin Institute of Technology, Harbin 150001, China, E-mail:(e-mail: lzr1102@gmail.com);(e-mail: lintao6324@gmail.com);(e-mail: xbaoc@hit.edu.cn); Chengfei Yue and Zijian Dai are with the Institute of Space Science and Applied Technology, Harbin Institute of Technology, Shenzhen 518055, China, E-mail:(e-mail: yuechengfei@hit.edu.cn);(e-mail: 22s061008@stu.hit.edu.cn); Antonella Ferrara is with The Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Pavia 27100, Italy, E-mail:(e-mail: antonella.ferrara@unipv.it)

I. INTRODUCTION

In recent years, space robots have emerged as a safe and efficient way to assist or replace astronauts in various tasks [1], [2]. These robots can automate routine operations, such as capturing or transporting objects, thereby reducing the workload of human operators and enhancing efficiency [3]. On-orbit manipulation typically involves multiple stages, each with distinct control requirements. Traditional approaches, which rely on task-specific controller designs, are not only complex but also lack versatility and reusability. An economical approach is to cluster these stages into a limited set of fundamental task elements, known as skills [4]. This hierarchical task model simplifies the design of complex controllers for on-orbit operations by shifting the focus to planning skill sequences and implementing sequential control for space robots. Recent advancements in task planning, particularly with large language models, have significantly enhanced planning results and efficiency [5]. However, further research and development are needed to improve motion planning and control modules for sequential manipulations.

Sequential control for space robots involves designing controllers for individual skills and managing transitions between these skills. While the design of controllers for specific skills is well-established, using advanced control theories such as sliding mode control [6], composite control based on fully actuated systems [7], and time-optimal control [8], model predictive control (MPC) stands out for its flexibility and efficiency [9], [10].

However, when integrating sub-controllers to execute the sequences, many existing approaches neglect the potential impact of parameter changes at switching points on system performance, often relying on hard switching strategies for connections [11]. To address the jerky movements caused by abrupt cost function changes during transitions, the switching process is incorporated within the prediction horizon to develop a time-dependent SMPC [12]. Building on this, state-dependent switching provides a more comprehensive approach to accommodate the diverse conditions encountered during on-orbit operations. However, this method needs additional optimization of the switching instants. In [13], a transformation method is proposed that converts the original optimal control problem (OCP) into an equivalent problem parameterized by a extended state defined from switching instants. The corresponding derivatives of the cost function are then obtained by solving a two-point boundary value problem. In [14], an efficient second-order method base on derived gradient and Hessian is proposed; however, when applied to nonlinear systems, the linearization over an underlying time grid is required. Another approach is proposed to directly calculate this derivative. Initially presented for autonomous systems in [15], it is later expanded

to non-autonomous nonlinear switched systems by [16]. However, this method has been applied exclusively to switching optimal control problems (OCP). For switching model predictive control (SMPC), it is essential to account for changes in the sequence within the prediction horizon.

Additionally, while the design of SMPC can enhance overall task performance, it also influences the local behavior of each stage. Incorporating switches within the prediction horizon introduces cost functions from multiple stages, resulting in a multi-objective optimization problem. This added complexity may create conflicts between objectives or constraints, potentially preventing each subtask from achieving optimal performance. In designs with *strict or hard priorities*, the performance of higher-priority tasks is always guaranteed, as lower-priority tasks are executed only within the null space determined by nonlinear lexicographic optimization [17] or the Jacobian method [18]. However, isolating stages in this manner not only fails to enhance transition performance but may also limit potential synergies that could improve overall task efficiency. For example, during the "Approach-Capture" procedure, a space robot can begin executing the Capture operation concurrently with the Approach process. Other methods adopt *soft priorities*, where solutions are derived by a combination of weighted tasks [19]. All of the tasks can be executed, even if low-priority tasks may be compromised to ensure the high performance of high-priority tasks. Similarly, taking the on-orbit capture process of a space robot as an example. In the on-orbit capture process of space robots, the target capture task has a higher priority than the base attitude control task. The soft priority method combines and solves the weighted cost functions for both tasks, enabling high stable capture while maintaining attitude control. In contrast, the hard priority method solves the capture task first and conduct base attitude control when the capture is finished. In the capture phase, the base attitude control is ignored without additional degrees of freedom. While the use of weights cannot entirely eliminate conflicts between stages, it enables leveraging their synergistic effects, improving transition smoothness and enhancing the overall performance of space robot operations.

Motivated by the aforementioned challenges, the priority-based switching model predictive control method is proposed for the sequential manipulation of space robot. Inspired by [12], this method extends SMPC for a predefined skills and manipulation sequence, to incorporate state-dependent conditions. As shown in Fig.1, when switches occur within the prediction horizon, the control input for the current time instant is derived as a combination of inputs from all stages. To address the conflicts between subtasks arising in the prediction horizon, a task formulation incorporating preference-based priorities is developed. Additionally, the proposed method addresses and utilizes the premature initiation of subsequent stages by incorporating priority information into the SMPC framework, enhancing task efficiency. The impact of these priority weights on system performance is then evaluated through Monte Carlo simulations and further analyzed using Pareto optimality theory.

Compared with the existing works, the contributions of this paper lie in the following aspects

- 1) The soft priority strategy is introduced within a hierarchical task formulation to mitigate potential conflicts between task objectives of different stages for space robots.
- 2) For the sequential manipulation control of space robots, the state-dependent switching model predictive control method is designed to improve the transition performance. To address the challenge of on-orbit operations with multiple stages and long durations, an online switching detection method is introduced.
- 3) The causes of premature initiation of subsequent stages in traditional SMPC are analyzed. By introducing weights that represent soft priorities, this phenomenon is effectively utilized to enhance task execution efficiency. Finally, a numerical analysis is performed to provide design guidance.

NOTATION 1 In this paper, some standard notations are used. \mathbb{R}^n and $\mathbb{R}^{n \times m}$ denote the n -dimensional Euclidean space and the set of all $n \times m$ real matrices. $\text{diag}(x_1, x_2, \dots, x_n)$ represents a matrix with diagonal elements being x_1, x_2, \dots, x_n and 0 elsewhere. $\|x\| = \sqrt{x^T x}$ is the Euclidean vector norm.

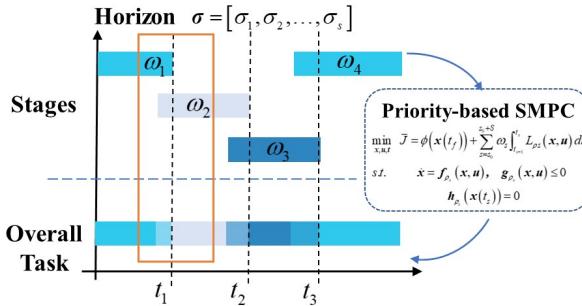


Fig. 1. Diagram for the main result. Around the switching points, the intensity of colors represents the combinations of stages with different weights.

II. SYSTEM MODELING

The space robot model used in this paper consists of a rigid-body **Base** and n rigid **Arms**, as shown in Fig.2. Before introducing the model, some basic assumptions are made as follows [6].

ASSUMPTION 1 The space robot operates in an aerospace environment where tasks are performed within a close range. Consequently, environmental forces such as gravity gradients and solar radiation pressure, as well as relative orbital dynamic effects, are considered negligible.

ASSUMPTION 2 *The space manipulator's motion states and dynamic properties are observable or known a priori.*

The derivation of the dynamic and kinematic models for space robots with multiple robotic arms follows methods well-documented in classic literature. Therefore, this paper focuses on providing only the key elements relevant to the subsequent controller design. To describe the position and orientation of the space robot effectively, the following coordinate systems are defined: \sum_I is the inertial frame; \sum_B is the body frame of the base, usually defined at the CoM (Centre of Mass) of the base; $\sum_{ei}(i = 1, 2, \dots, n)$ is the end-effector frames of the arm i .

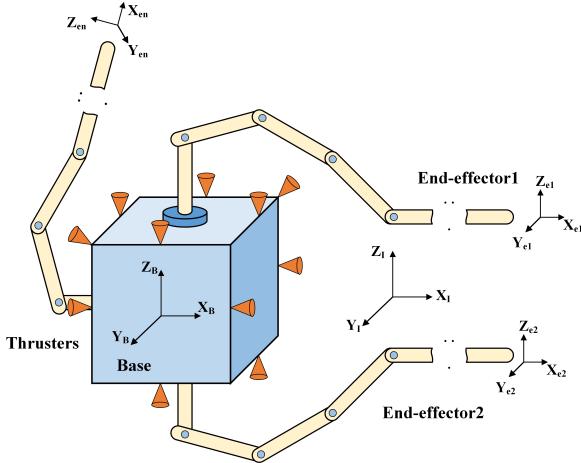


Fig. 2. The general model of a space robot system

Thus, the pose of base and end-effectors can be described as

$$\xi = \begin{bmatrix} p_b \\ p_{e1} \\ \vdots \\ p_{en} \end{bmatrix} = g(\mathbf{Q}) \quad (1)$$

where, $\mathbf{Q} = [p_b^T, q^T]^T$ is the state of space robot in joint space, and p_b is the pose of base, $q = [q_1^T, \dots, q_n^T]^T$ is the joint angles; p_{en} is the pose of n th end-effector.

By differentiating (1), the relationship between the linear and angular velocities of each end-effector and those of the base and joint velocities can be derived

$$\begin{aligned} \dot{\xi} &= \mathbf{J}_B \dot{p}_b + \mathbf{J}_M \dot{q} \\ &= \mathbf{J}_C \dot{\mathbf{Q}} \end{aligned} \quad (2)$$

where, \mathbf{J}_B and \mathbf{J}_M are the partitioned Jacobian matrix for the base and the arms, respectively.

As a result of Lagrangian mechanics, the system dynamics is obtained

$$\mathbf{H}(\mathbf{Q})\ddot{\mathbf{Q}} + \mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}})\dot{\mathbf{Q}} = \mathbf{F}_{total} \quad (3)$$

where, \mathbf{F}_{total} represents the generalized forces corresponding to the generalized coordinates, consisting of $\mathbf{F} = [\mathbf{F}_B^T \ \tau^T]^T$, where, \mathbf{F}_B is the base control force, τ is the joint control torque, and the external force applied on the end effector \mathbf{F}_h . $\mathbf{H}(\mathbf{Q})\ddot{\mathbf{Q}}$ is the inertial

force term proportional to the generalized acceleration; $\mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}})\dot{\mathbf{Q}} = \dot{\mathbf{H}}(\mathbf{Q})\dot{\mathbf{Q}} - \frac{1}{2}\dot{\mathbf{Q}}^T \frac{\partial \mathbf{H}(\mathbf{Q})}{\partial \mathbf{Q}} \dot{\mathbf{Q}}$ is the nonlinear term related to position and velocity that can be divided as $\mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}})\dot{\mathbf{Q}} = [\mathbf{c}_B^T \ \mathbf{c}_M^T]^T$, \mathbf{c}_B^T and \mathbf{c}_M^T are the nonlinear term related to the base and joints, respectively.

Since this paper does not investigate the interactions between the space robot and external environments, external forces are assumed to be zero. Besides, for the ease of subsequent controller design based on the dynamics equations, (3) can be rewritten as

$$\begin{bmatrix} \mathbf{H}_B & \mathbf{H}_{BM} \\ \mathbf{H}_{BM}^T & \mathbf{H}_M \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_B \\ \mathbf{C}_M \end{bmatrix} = \begin{bmatrix} \mathbf{F}_B \\ \tau \end{bmatrix} \quad (4)$$

The details of the coefficient matrices $\mathbf{H}(\mathbf{Q})$ and the velocity-dependent terms $\mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}})$ can be found in [20].

Furthermore, as space robot operations are generally described in task or Cartesian coordinates, to facilitate the direct controller design based on task objectives, a task-space model of the space robot is established:

$$\mathbf{H}_x(\mathbf{Q})\ddot{\xi} + \mathbf{C}_x(\mathbf{Q}, \dot{\mathbf{Q}}) = \mathbf{F}_m \quad (5)$$

where,

$$\begin{aligned} \mathbf{H}_x &= \mathbf{J}_C^{-T} \mathbf{H} \mathbf{J}_C^{-1} \\ \mathbf{C}_x &= \mathbf{J}_C^{-T} \mathbf{C} - \mathbf{H}_x \mathbf{J}_C \dot{\mathbf{Q}} \\ \mathbf{F}_m &= \mathbf{J}_C^{-T} \mathbf{F} \end{aligned} \quad (6)$$

However, in model predictive control (MPC) design, systems with strong nonlinearity and high degrees of freedom, such as those described by (4), often impose a significant computational burden. To address this, practical robotics and spacecraft control commonly employ a hierarchical control approach to simplify the optimization process. This involves utilizing low-level controllers for inverse dynamics inner-loop feedback linearization, ultimately reducing the system to a kinematic model that assumes direct control of joint accelerations [21]. More specifically, when the matrix $\mathbf{H}(\mathbf{Q})$ is full rank of all possible \mathbf{Q} , by designing a lower-level controller [22]

$$\mathbf{F} = \mathbf{H}(\mathbf{Q}) \mathbf{u} + \mathbf{C}(\mathbf{Q}, \dot{\mathbf{Q}}) \dot{\mathbf{Q}} \quad (7)$$

where \mathbf{u} is the applied input signal.

Then, choosing the state as $\mathbf{x} = [\mathbf{Q}^T, \dot{\mathbf{Q}}^T]^T$, the resulting nonlinear system which can be used in the MPC is given by

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}_1(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{\mathbf{Q}} \\ \mathbf{u} \end{bmatrix} \\ \xi = \mathbf{g}(\mathbf{Q}) \end{cases} \quad (8)$$

Similarly, for task space dynamics, the state variables are selected as $\dot{\mathbf{x}} = [\xi^T, \dot{\xi}^T]^T$ and organized into a format suitable for use in MPC, expressed as $\dot{\mathbf{x}} = \mathbf{f}_2(\mathbf{x}, \mathbf{u})$.

III. PRIORITY-BASED SWITCHING MPC

For the sequential control of space robots with a given task sequence, it is an efficient method to construct it as an SMPC with switching time optimization. However, applying state-dependent OCP methods to address this

problem presents several challenges: (i) In the receding horizon manner, although the overall task sequence is predefined, the sequence within the prediction horizon at each instant is still unknown. (ii) When switches occur within the prediction horizon, the control input at the current time is a combination of inputs for all stages within that horizon. This can lead to the premature initiation of subsequent subtasks. Particularly when conflicts arise, it creates a multi-objective optimization problem, preventing each subtask from achieving its optimal performance.

To achieve this, we intend to enhance the switching optimal control method in this section by adapting its traditionally fixed sequence to a receding horizon style. More importantly, for challenge (ii), instead of simply suppressing interactions using hard-priority methods, we extend the concept of soft priorities in task formulation and introduce them as the weights in the SMPC. This approach allows us to adjust interactions between stages, retaining and fully leveraging their beneficial aspects.

A. TASK FORMULATION

This section defines the task formulation required for space robots. As introduced above, there are various task representation models that decompose the overall task into sequences composed of several reusable basic elements, namely **Skill** [23], [24]. Skills are designed based on the task requirements, operational scenarios, and capabilities of the space robot to cover a typical on-orbit operation process. Based on these skills, elements such as state trajectories and switching conditions are developed, enabling their connection and reuse to form a complete task sequence.

The core of these skills is to construct a set of essential parameters based on the specific controller corresponding to each skill. Here, MPC is utilized as a generic control framework, which requires only parameter adjustments according to different skill definitions. This framework effectively simplifies the design of controllers for sequences involving multiple subtasks with varying requirements.

Therefore, assuming that the space robot requires at least K skills to perform all the on-orbit operation tasks. The skill properties of space robots are extracted and parameterized, represented as a tuple in the form

$$\sigma := (\mathcal{T}, \mathcal{W}, \mathcal{B}, \mathcal{C}, \mathcal{P}) \quad (9)$$

where, $\mathcal{T}, \mathcal{W}, \mathcal{B}, \mathcal{C}$ are commonly used elements for hierarchical control [25] or task planning framework [26].

Considering the distinct advantages of MPC in handling complex systems with strong non-linearity and multiple constraints, the elements of k th skill is incorporated into the MPC framework to construct the cost function

$$\bar{L}_k(x_k(t), u_k(t)) = \|x_k - x_k^d\|_{W_k}^2 + \|u\|_{W_r}^2 \quad (10)$$

where, $k \in \mathcal{K}$ and $\mathcal{K} = \{1, \dots, K\}$ ($K \geq 1$), $\mathcal{T} \subset \{\mathbf{x}_k^d\}$ represents the reference trajectory of the generalized coordinates \mathbf{x} ; \mathcal{W} represents the controller strategies, and due to the homogeneous design of the skill controllers, it

essentially constitutes a set of controller parameters $\mathcal{W} \subset \{\mathbf{W}_k, \mathbf{W}_r\}$. $\mathcal{B} \subset \{\mathbf{f}_k, \mathbf{g}_k\}$ represents the equality and inequality constraints during operation; $\mathcal{C} \subset \{\mathbf{h}_k(\mathbf{x}(t_{kf}))\}$ represents the boundary conditions for each stage, used to determine the switching instants.

Given a predefined sequence of skills, these four elements are enough for designing an SMPC controller that meets the requirements of overall on-orbit operation task. However, as mentioned before, when optimizing around switching points, conflicts between consecutive skills will prevent all of them from achieving global optimal performance. To address this, an additional element, \mathcal{P} , is introduced in this paper to represent the relative priority between skills.

Since the priority information \mathcal{P} effectively reflects user preferences for specific skills, integrating it into the existing MPC framework can be challenging. In the field of multi-objective optimization, a common approach is to incorporate it as weights into the cost function of each skill (10)

$$\mathcal{P} \subset \{\omega_k \mid k \in \mathcal{K}\} \quad (11)$$

Determining the value of \mathcal{P} is crucial for priority design. When the task execution process is known, the activation functions for weights can be learned using techniques such as Learning from Demonstration (LfD) or other data-driven methods [27]. However, for space robots, conducting multiple pre-experiments to collect training data is impractical. Therefore, weight values must be designed based on expert's preferences, and the relative priorities between different stages can be expressed through the differentiation in the magnitudes of these weights with paired comparison methods [28].

Moreover, the final solution may not accurately reflect the priorities expressed by the weights because the relative importance in the solving process is determined by the value of the whole cost function $\omega_k \bar{L}_k$. When there is a significant difference in the values of \bar{L}_k , the relative magnitude of the weights needs to be designed to compensate for this disparity. While the independent absolute value of the weights does not affect the relative priorities, excessively large or small magnitudes of the cost function can significantly effect the convergence speed and transient performance of the control system. Therefore, normalization of the skill cost functions is employed to facilitate the design of relative level for priorities [29]

$$L_k = \frac{\bar{L}_k - \bar{L}_k^*}{\bar{L}_k^{nad} - \bar{L}_k^* + \zeta} \quad (12)$$

where, \bar{L}_k^* and \bar{L}_k^{nad} are the minimum and maximum value of cost function in the previous prediction horizon, respectively; ζ is a threshold used to prevent issues such as division by zero.

Based on a predefined library of skills, **Task** σ can be described as a sequence composed of piecewise activation of skills, namely **Stage** σ_{ρ_s}

$$\sigma = [\sigma_{\rho_1}, \sigma_{\rho_2}, \dots, \sigma_{\rho_{S+1}}] \quad (13)$$

which means the system undergoes S switches in the time interval $[0, T]$; $\rho_s \in \mathcal{K}$.

B. MPC Formulation

Consistent with the previous section, it is assumed that the space robot needs the capability to perform at least K skills. Such that for each $k \in \mathcal{K}$, f_k is a vector field that describes the system dynamics for skill k . And the operation sequence σ are defined by some advanced task planning methods as *a priori*.

However, during the prediction horizon $[T_0, T_f]$, the switching instants are not entirely known. Assuming that S switches occur in the time horizon, $t_s \in \mathbb{R}$ for $s = s_0, s_0 + 1, \dots, s_0 + S - 1 \in \mathbb{R}^S$ is introduced to represent the time of the switch from task σ_{ρ_s} to $\sigma_{\rho_{s+1}}$, where $\sigma_{\rho_s} \in \sigma$ and

$$0 \leq T_0 \leq t_{s_0} \leq t_{s_0+1} \leq \dots \leq t_{s_0+S-1} \leq T_f \leq T \quad (14)$$

where, $t_0 = t_{s_0-1}$ and $t_f = t_{s_0+S}$.

In every stage, the corresponding evolution of the state \mathbf{x} is described by

$$\dot{\mathbf{x}} = \mathbf{f}_{\rho_s}(\mathbf{x}(t), \mathbf{u}(t)), t \in [t_{s-1}, t_s] \quad (15)$$

where \mathbf{f}_{ρ_s} is the system model established in II.

Considering the compliance and safety during the on-orbit operation, it is assumed that there are no state jumps during the switching between stages, meaning that $\lim_{t \uparrow t_s} \mathbf{x}(t) = \lim_{t \downarrow t_s} \mathbf{x}(t)$. The optimal control problem is to find the switch time vector $\mathbf{t} = [t_{s_0}, t_{s_0+1}, \dots, t_{s_0+S-1}]$ and the control input \mathbf{u} that minimize the cost function $J(\mathbf{x}, \mathbf{u}, \mathbf{t})$ given as

$$J = \sum_{s=s_0}^{s_0+S} \omega_s \int_{t_{s-1}}^{t_s} L_{\rho_s}(\mathbf{x}(t), \mathbf{u}(t)) dt + \phi(\mathbf{x}(t_f)) \quad (16)$$

where, ω_s , $\phi(\mathbf{x}(t_f))$ and L_{ρ_s} are the preference-based weight representing the priority information, terminal cost and task-dependent running costs defined in the task formulation, respectively. As the costs are defined as twice differentiable function with respect to the decision variables, ϕ , L_{ρ_s} , J and \mathbf{f}_{ρ_s} in this paper meet the assumptions that they are Lipschitz continuous and have Lipschitz derivatives in \mathbf{x} .

Compared to traditional SMPC, the weights adjust the relative contributions of the cost functions at each stage, ensuring that the system's behavior near switching points adheres to a predefined order of importance. This approach can not only prevent interference between stages but also enhance their synergy, thereby improving overall task performance and efficiency.

Various constraints must be considered in the optimization process. In addition to the dynamics equation constraints (15) and the switching time constraints (14), it is also necessary to consider equality or inequality constraints including state and control limits

$$\mathbf{g}_{\rho_s}(\mathbf{x}(t), \mathbf{u}(t)) \leq 0 \quad (17)$$

, as well as switching condition constraints

$$\mathbf{h}_{\rho_s}(\mathbf{x}(t_s)) = 0 \quad (18)$$

which is a general expression of state-dependent switching, and can also be reformulated into a time-dependent form when necessary.

To solve the designed switching model predictive controller, two main challenges must be addressed: first, at each instant, solve the optimal control problem defined in (16) based on the given switching sequences within the prediction horizon; second, determine the switching sequence in the prediction horizon at the next instant.

C. Solving OCP

At each time instant, the SMPC can be treated as an open-loop switching OCP. A commonly used approach for solving such OCPs is the direct multiple shooting (DMS) method, which discretizes the continuous problem to enable efficient and robust numerical computation [30], [31]. The prediction horizon is discretized into N time intervals. For switching OCP, to ensure that switching occurs only at sampling points, a predefined number of sampling points N_s is allocated to each stage rather than using a fixed-step sampling approach as in conventional NMPC. The local sampling times are calculated as

$$\Delta t_s = \frac{t_s - t_{s-1}}{N_s} \quad (19)$$

Then, based on the discretized state $\mathbf{X} = \{\mathbf{x}(i, s)\}$, discretized control input $\mathbf{U} = \{\mathbf{u}(i, s)\}$ and switching instants $\mathbf{t} = [t_{s_0}, t_{s_0+1}, \dots, t_{s_0+S-1}]$, the resultant nonlinear programming (NLP) is obtained

$$\begin{aligned} & \min_{\mathbf{X}, \mathbf{U}, \mathbf{t}} \bar{J} = \phi(\mathbf{x}_N) + \sum_{s=s_0}^{s_0+S} \sum_{i=1}^{N_s} \omega_s L_{\rho_s}(\mathbf{x}_{i,s}, \mathbf{u}_{i,s}) \Delta t_s \\ & \text{s.t. } \mathbf{x}_0 = \bar{\mathbf{x}}, \\ & \quad \mathbf{x}_{i,s} + \mathbf{f}_{\rho_s}(\mathbf{x}_{i,s}, \mathbf{u}_{i,s}) \Delta t_s - \mathbf{x}_{i+1,s} = 0, \\ & \quad \mathbf{g}_{\rho_s}(\mathbf{x}_{i,s}, \mathbf{u}_{i,s}) \leq 0, \\ & \quad \mathbf{h}_{\rho_s}(\mathbf{x}_{N_s,s}) = 0, \\ & \quad t_{s-1} - t_s \leq 0 \end{aligned} \quad (20)$$

where the MPC cost function (16) and constraints (14) (18) are discretized. Since the DMS sampling step size is calculated by the optimized switching instants and predefined number of sampling steps, there may be instances where a large step size compromises simulation accuracy, especially when switches begin to appear or are about to disappear within the horizon. Here N_s is designed to be a relatively large number of steps, sacrificing some computational efficiency for improved simulation accuracy. Additionally, a mesh-refinement method can be employed, which involves monitoring the step size and adding more steps when it becomes too large, ensuring better resolution during critical transitions.

To solve the NLP problem and optimize the switching time instants and control input, gradient information must be provided to the NLP solver. The gradient of cost function with respect to the control input $D_{\mathbf{U}} \bar{J}(\mathbf{X}, \mathbf{U}, \mathbf{t})$

can be easily obtained by solving sensitivity equations as traditional OCP and here generated by CasADi, a famous open-source software tool for algorithmic differentiation [32].

And to calculate that with respect to switching instants $D_t \bar{J}(\mathbf{X}, \mathbf{U}, t)$, in [16], a direct method is proposed. For any $\nu : \rightarrow \mathbb{R}^n$, $\nu(t^-) = \lim_{\tau \uparrow t} \nu(\tau)$ and $\nu(t^+) = \lim_{\tau \downarrow t} \nu(\tau)$ are defined as the left and right limits of the variable ν , respectively. Then, it is defined that

$$\begin{aligned}\Delta \mathbf{f}(t_s^-) &= \mathbf{f}_{\rho_s}(\mathbf{x}(t_s), \mathbf{u}(t_s^-)) - \mathbf{f}_{\rho_{s+1}}(\mathbf{x}(t_s), \mathbf{u}(t_s^-)) \\ \Delta \mathbf{f}(t_s^+) &= \mathbf{f}_{\rho_s}(\mathbf{x}(t_s), \mathbf{u}(t_s^+)) - \mathbf{f}_{\rho_{s+1}}(\mathbf{x}(t_s), \mathbf{u}(t_s^+)).\end{aligned}\quad (21)$$

The differentiation of adjacent cost functions $\Delta L(t_s^-)$ and $\Delta L(t_s^+)$ are also defined similarly.

LEMMA 1 [15]: *The directional derivative of the cost function in the direction e_s exists and*

$$\nabla_{e_s} J(\mathbf{x}, \mathbf{u}, t) = \psi^T(t_s) \Delta \mathbf{f}(t_s^+) + \Delta L(t_s^+) \quad (22)$$

where $\psi(t), t \in [t_0, t_f]$ is governed by the following piecewise ODEs:

$$\begin{aligned}\dot{\psi} &= -\frac{\partial \mathbf{f}_{\rho_{s+1}}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u})\psi - \frac{\partial L_{\rho_{s+1}}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u}) \\ \psi(t_f) &= D_{\mathbf{x}}\phi(\mathbf{x}(t_f))^T\end{aligned}\quad (23)$$

where, $t \in [t_s, t_{s+1}]$ and $\psi(t_s^-) = \psi(t_s^+)$.

Based on Lemma 1, the directional derivative of the augmented cost function (16) in direction e_s is given as

$$\nabla_{e_s} J(\mathbf{x}, \mathbf{u}, t) = \psi^T(t_s) \Delta \mathbf{f}(t_s^+) + \omega_s \Delta L(t_s^+) \quad (24)$$

Then, assuming that the system input is continuous at switching point, i.e. $\mathbf{u}(t_s^+) = \mathbf{u}(t_s^-)$, which is easily satisfied for system without state jumps and changes in controller structure, the partial derivative of the cost function with respect to the switching instants t_s exists and is given as:

$$\frac{\partial \bar{J}}{\partial t_s} = \psi^T(t_s) \Delta \mathbf{f}(t_s) + \omega_s \Delta L(t_s) \quad (25)$$

When applied to the discretized form (20), the following gradient is obtained

$$\begin{aligned}\frac{\partial \bar{J}}{\partial t_s} &= \psi^T(t_s) (\mathbf{f}_{\rho_s}(\mathbf{x}_{N_s, s}, \mathbf{u}_{N_s, s}) - \mathbf{f}_{\rho_{s+1}}(\mathbf{x}_{N_s, s}, \mathbf{u}_{N_s, s})) \\ &\quad + \omega_s (L_{\rho_s}(\mathbf{x}_{N_s, s}, \mathbf{u}_{N_s, s}) - L_{\rho_{s+1}}(\mathbf{x}_{N_s, s}, \mathbf{u}_{N_s, s}))\end{aligned}\quad (26)$$

where the costate ψ is also discretized as

$$\begin{aligned}\delta \psi &= -\frac{\partial \mathbf{f}_{\rho_{s+1}}}{\partial \mathbf{x}}(\mathbf{x}_{N_s, s}, \mathbf{u}_{N_s, s})\psi - \frac{\partial L_{\rho_{s+1}}}{\partial \mathbf{x}}(\mathbf{x}_{N_s, s}, \mathbf{u}_{N_s, s}) \\ \psi(t_f) &= D_{\mathbf{x}}\phi(\mathbf{x}_N)^T\end{aligned}\quad (27)$$

Finally, $D_t \bar{J}(\mathbf{X}, \mathbf{U}, t)$ exists and is given as $D_t \bar{J}(\mathbf{X}, \mathbf{U}, t) = [(\partial \bar{J}/\partial t_{s_0}), \dots, (\partial \bar{J}/\partial t_{s_0+S-1})]$.

D. Moving Sequence Method

As stated in challenge(i), unlike full-horizon optimization in OCP, MPC is optimized within a finite horizon. When only the skills sequence is given and the instants need to be optimized, the switches that may occur within

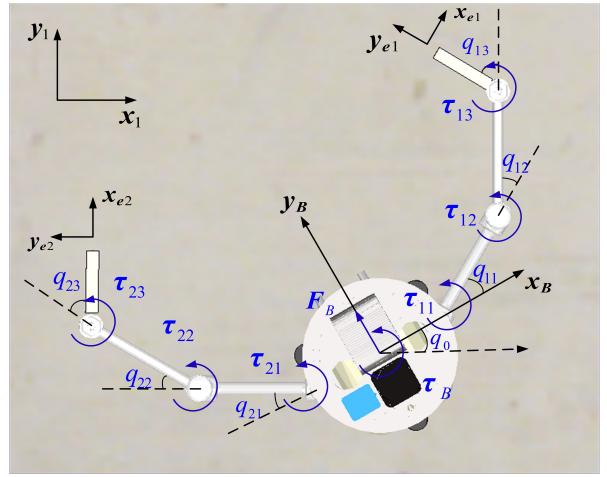


Fig. 3. Diagram of the space robot model

the prediction horizon remain unknown. Therefore, the moving switching sequence is introduced to determine the potential sequence in prediction horizon online before each computation [33].

Let $\sigma_s(t) = [\sigma_{s_0}, \sigma_{s_0+1}, \dots, \sigma_{s_0+S+1}]$ be the switching sequence known at the sampling instant t_{sample} . At the next instant $t_{sample} + \Delta t$, since only one switching occurs at a time and the interval between switches in space robot operation skills is generally longer than sampling step, the possible changes to the current sequence are whether the first element σ_{s_0} needs to be removed or a new element σ_{s_0+S+2} needs to be added at the end.

To achieve this, for the former one, if the first time t_{s_0} in the optimized switching instants is less than the sampling time $t_{sample} + \Delta t$, σ_{s_0} should be removed. For the latter one, the switching detection is conducted based on the prediction results of state $\mathbf{x}(t_{sample} + \Delta t)$. As long as one of these situations occurs, the task sequence should be updated, based on which (20) is solved to obtain the corresponding control inputs and switch times subsequently. Thus, with the overall sequence known, the moving sequence method ensures that the space robot tasks can be optimized online based on the SMPC.

E. Optimality Analysis

For the SMPC problem as shown in (16), its optimality can be ensured through the Karush-Kuhn-Tucker (KKT) conditions, as demonstrated in [33]. However, when conflicts arise between the requirements and corresponding cost functions of each stage, it becomes impossible to find an optimal solution that simultaneously optimizes all objectives. Therefore, the impact of priority on the optimality of each stage is analyzed here. Meanwhile, in SMPC, when no switches are detected within the prediction horizon, the optimality aligns with traditional MPC. Thus, the analysis will focus solely on the horizons where switches occur.

Algorithm 1 Priority-based SMPC

Require: Overall sequence σ , Current state $\mathbf{x}(t)$, Previous solution $\mathbf{u}(t)$, t , Previous sequence σ_s , Prediction horizon $[T_0, T_f]$

- 1: $\mathbf{X}(t) \leftarrow$ Predict state based on $\mathbf{x}(t)$, $\mathbf{u}(t)$, σ_s
- 2: **if** t_{s_0} from $t < t_0$ **then**
- 3: remove σ_{s_0} from σ_s
- 4: **end if**
- 5: **if** new switching σ_{s_0+S+2} is detected **then**
- 6: $\sigma_s \leftarrow [\sigma_s, \sigma_{s_0+S+2}]$
- 7: **end if**
- 8: **for** $k = s_0, \dots, s_0 + S + 2$ **do**
- 9: $L_k^*, L_k^{nad} \leftarrow \min L_k(\mathbf{X}, \mathbf{U}), \max L_k(\mathbf{X}, \mathbf{U})$
- 10: **end for**
- 11: $D_t J(\mathbf{X}, \mathbf{U}, t) \leftarrow (25)$
- 12: $D_U J(\mathbf{X}, \mathbf{U}, t) \leftarrow$ Calculated by Casadi
- 13: $\mathbf{X}^*, \mathbf{U}^*, t^* \leftarrow$ NLPSSolver((20), $\mathbf{X}, \mathbf{U}, D_t J, D_U J$)
- 14: **return** $\mathbf{X}^*, \mathbf{U}^*$

LEMMA 2 [34] A feasible point \mathbf{x}^* is Pareto optimal if and only if there exists no other feasible point such that $J(\mathbf{x}) \leq J(\mathbf{x}^*)$.

As mentioned in Challenge ii), traditional switched MPC exhibits a phenomenon of premature initiation, where the actual switching occurs before the expected switching time. We assume that the switching time instants t has been optimized at the current moment t_0 . Furthermore, in the absence of weight adjustments, the cost function of stage ρ_s is expressed as:

$$J_{\rho_s} = \int_{t_{s-1}}^{t_s} L_{\rho_s}(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (28)$$

The corresponding optimal solution is $(\mathbf{x}_{\rho_s}^*, \mathbf{u}_{\rho_s}^*)$.

For a multi-stage control, the cost function of switching MPC, defined in (16), is essentially a multi-objective optimization problem formed by the cost functions of each stage

$$\min J = \sum_{s=s_0}^{s_0+S} J_{\rho_s}(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (29)$$

The cost function of the subsequent stages J_{ρ_s} are also optimized at time t_0 . The solution $(\mathbf{x}^*, \mathbf{u}^*)$ is a Pareto optimal solution, composed of the optimal solutions for each stage $(\mathbf{x}_{\rho_s}^*, \mathbf{u}_{\rho_s}^*)$ [34]. It means that the action of stage ρ_s should start immediately, rather than waiting until after the optimized switching instant t_{s-1} .

However, when stage ρ_s first appears within the prediction horizon, J_{ρ_s} is influenced by its integration horizon from t_{s-1} to t_s and takes a relatively small value. As analyzed in [28], this results in a significant deviation of $(\mathbf{x}^*, \mathbf{u}^*)$ from $(\mathbf{x}_{\rho_s}^*, \mathbf{u}_{\rho_s}^*)$. In engineering practice, this can be approximated as no motion occurring. To intuitively describe the task process, the actual switching instant is defined as the moment when not only J_{ρ_s} is optimized, but also the degree of closeness between $(\mathbf{x}^*, \mathbf{u}^*)$ and $(\mathbf{x}_{\rho_s}^*, \mathbf{u}_{\rho_s}^*)$ exceeds an empirical threshold ε

. This indicates that the extent of premature initiation in each stage is related to the relative magnitude of its cost function value J_{ρ_s} and can be adjusted through weight correction.

Additionally, as described in [35], if a given stage is realizable exactly, it can be achieved using a weighted-sum objective with positive weights $\omega_s > 0$ at any given precision. These two points illustrate that, a well-designed weight strategy can achieve the desired accuracy and initiation instant for each stage, thereby improving overall task performance and efficiency.

Besides, in multi-objective optimization problems for linear systems, the influence of weights on the optimal solution can be clearly expressed. However, this paper focuses on nonlinear space robot systems, where the stages are not entirely independent but are interconnected through initial and final state constraints. This makes the traditional methods used in multi-objective optimization inapplicable. Therefore, a numerical approach will be adopted in the following sections to address this issue.

IV. SIMULATION RESULTS

To validate the effectiveness of the proposed algorithm, simulations are conducted using a planar dual-arm space robot model established in CoppeliaSim Edu 4.0.0, as shown in Fig.3. Please note that, as the focus of this paper is not on improving computational efficiency when applying SMPC to high-degree-of-freedom systems, nor on some specific tasks requiring coordinated operation of multiple robotic arms. Therefore, simplifying the simulation with planar motion is both reasonable and widely adopted [7], [36]. The detailed parameters are listed in Table.I, where 0 represents the base, and 1 to 3 represent the links extending outward from the base. Given the symmetrical configuration of the space robot's arms, it is sufficient to provide the link parameters for only one arm.

Furthermore, in the simulations, the prediction horizon of MPC is set to be $T_h = T_f - T_0 = 4s$. For each stage, the same value of discrete steps $N_s = 100$ is assigned.

TABLE I
Parameters of the space robot model

Body No.	$m_i(kg)$	$I_i(kg \cdot m^2)$	$b_i(m)$	$a_i(m)$
0	30	1.35	0.3	–
1	4.2	0.35	0.25	0.25
2	4.2	0.35	0.25	0.25
3	2	0.0267	0.1	0.1

A. Simulation Scenario

During the numerical simulation, a classic on-orbit transportation scenario using space robot is designed, consisting of four stages: Approach, Capture, Transport, and Release, as shown in Fig.4 [37],

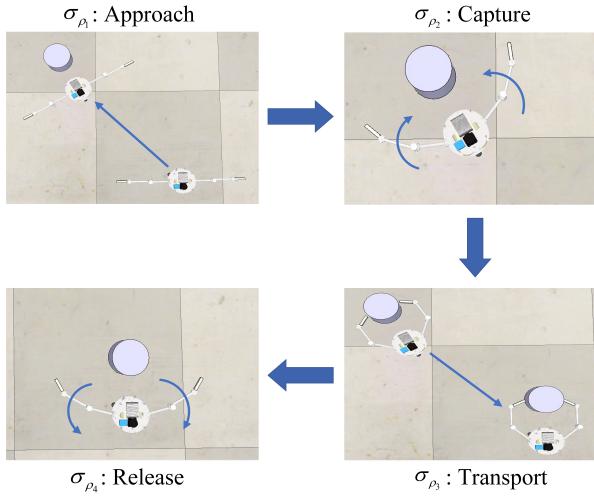


Fig. 4. The on-orbit transportation scenario used for simulation

[38]. Without considering contact dynamics, the space robot must operate in at least two working modes, or skills to cover this on-orbit operation scenario:

(1) The first skill involves the simultaneous control of the base pose and joint angles within the joint space during maneuvering operations. Therefore, the system state is described in the joint space as $\mathbf{x} = [\mathbf{Q}^T, \dot{\mathbf{Q}}^T]^T$ and governed by the system dynamics $\dot{\mathbf{x}}(t) = \mathbf{f}_1(\mathbf{x}(t), \mathbf{u}(t))$. The aim of this skill is to stabilize the robot at a set-point or along a desired trajectory \mathbf{Q}^d . Accordingly, the cost function should be formulated as

$$\bar{L}_1(\mathbf{x}(t), \mathbf{u}(t)) = \|\mathbf{Q} - \mathbf{Q}^d\|_{W_q}^2 + \|\mathbf{u}\|_{W_r}^2 \quad (30)$$

(2) The second skill focuses only on tracking the end-effector pose in task space while keeping the base pose fixed during operations. This approach facilitates intuitive task design and ensures the stability of the manipulation process. Therefore, the system state and dynamics are chosen as $\dot{\mathbf{x}} = [\xi^T, \dot{\xi}^T]^T$ and $\dot{\mathbf{x}} = \mathbf{f}_2(\mathbf{x}(t), \mathbf{u}(t))$, respectively. Given the desired state $\xi^d = [\mathbf{p}_{b0}^T, \mathbf{p}_{e1d}^T, \mathbf{p}_{e2d}^T]^T$, the following cost function should be optimized

$$\bar{L}_2(\mathbf{x}(t), \mathbf{u}(t)) = \|\xi - \xi^d\|_{W_p}^2 + \|\mathbf{u}\|_{W_r}^2 \quad (31)$$

Furthermore, for both modes, the space robot is subject to some common constraints, including angular position constraints $\mathbf{Q} \in [\mathbf{Q}_{\min}^T, \mathbf{Q}_{\max}^T]^T$ and velocity constraints $\dot{\mathbf{Q}} \in [\dot{\mathbf{Q}}_{\min}^T, \dot{\mathbf{Q}}_{\max}^T]^T$.

REMARK 1 *The skill design presented here is merely a feasible, though not necessarily optimal, approach aimed at integrating with advanced task planning methods and offering a convenient framework for describing overall task sequences. In practice, even with only a single skill, the switching of reference trajectories, state constraints, or control parameters during multi-stage on-orbit operations can effectively validate the primary contribution of this paper: the priority-based switching MPC.*

The sequence of overall tasks is designed as shown in Table II. The detailed introduction and parameters of the sequence σ are provided as follows:

(1) Approach σ_{ρ_1} : The space robot maneuvers from its initial position $\mathbf{Q}_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ to a designated location to prepare for capturing the target object. As the first subtask, it is assigned baseline importance in the overall operation, serving as the reference point for priority design, i.e. $\omega_1 = 1$. This means that the weight values for subsequent subtasks will be chosen relative to this baseline priority. Then, based on the formulation of the first skill, the reference and control parameters are

$$\mathbf{Q}_{\rho_1}^d = \begin{bmatrix} -2m & -2m & \pi/6\text{rad} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{W}_q = \text{diag}(100, 100, 10, 100, 100, 10, 100, 100, 10)$$

Besides, in addition to skill-related equality constraint of dynamics $\dot{\mathbf{x}} = \mathbf{f}_1(\mathbf{x}, \mathbf{u})$, the following constraints are applied: the limits on angular position

$$\mathbf{Q}_{\min} = \begin{bmatrix} -5m & -5m & -\pi\text{rad} \\ -\frac{\pi}{2}\text{rad} & -\pi\text{rad} & -\pi\text{rad} \\ -\frac{\pi}{2}\text{rad} & -\pi\text{rad} & -\pi\text{rad} \end{bmatrix}$$

$$\mathbf{Q}_{\max} = \begin{bmatrix} 5m & 5m & \pi\text{rad} \\ \frac{\pi}{2}\text{rad} & \pi\text{rad} & \pi\text{rad} \\ \frac{\pi}{2}\text{rad} & \pi\text{rad} & \pi\text{rad} \end{bmatrix}$$

and velocity

$$\dot{\mathbf{Q}}_{\min} = \begin{bmatrix} -0.2\text{m/s} & -0.2\text{m/s} & -0.2\text{rad/s} \\ -0.15\text{rad/s} & -0.15\text{rad/s} & -0.15\text{rad/s} \\ -0.15\text{rad/s} & -0.15\text{rad/s} & -0.15\text{rad/s} \\ 0.2\text{m/s} & 0.2\text{m/s} & 0.2\text{rad/s} \\ 0.15\text{rad/s} & 0.15\text{rad/s} & 0.15\text{rad/s} \\ 0.15\text{rad/s} & 0.15\text{rad/s} & 0.15\text{rad/s} \end{bmatrix}$$

are consistent across all stages, thus they will not be reintroduced below.

TABLE II
The predefined task sequence

σ	\mathcal{T}	\mathcal{W}	\mathcal{B}	\mathcal{P}
σ_{ρ_1}	$\mathbf{Q}_{\rho_1}^d$	$\mathbf{W}_q, \mathbf{W}_r$	$\mathbf{x}(t_1) = \mathbf{Q}_{\rho_1}^d$	$\omega_1 = 1$
σ_{ρ_2}	$\xi_{\rho_2}^d$	$\mathbf{W}_p, \mathbf{W}_r$	$\mathbf{x}(t_2) = \xi_{\rho_2}^d$	$\omega_2 = 10^2$
σ_{ρ_3}	$\mathbf{Q}_{\rho_3}^d$	$\mathbf{W}_q, \mathbf{W}_r$	$\mathbf{x}(t_3) = \mathbf{Q}_{\rho_3}^d$	$\omega_3 = 1$
σ_{ρ_4}	$\xi_{\rho_4}^d$	$\mathbf{W}_p, \mathbf{W}_r$	-	$\omega_4 = 10^{-2}$

(2) Capture σ_{ρ_2} : The most critical part of the operation process is capturing the target object using the space robot. As such, this stage is assigned the highest priority, with a preference weight of $\omega_2 = 10^2$. Upon reaching the designated position $\mathbf{x}(t_1) = \mathbf{Q}_{\rho_1}^d$, the controller transitions to the second skill. At this stage, the controller tracks the desired end-effector state while keeping the base fixed.

$$\mathbf{p}_e^d = \begin{bmatrix} -1.9334\text{m} & 3.1677\text{m} & 2.8798\text{rad} \\ -3.0446\text{m} & 2.5261\text{m} & -1.8326\text{rad} \end{bmatrix}$$

$$\mathbf{W}_p = \text{diag}(100, 100, 100, 10, 10, 1, 10, 10, 1)$$

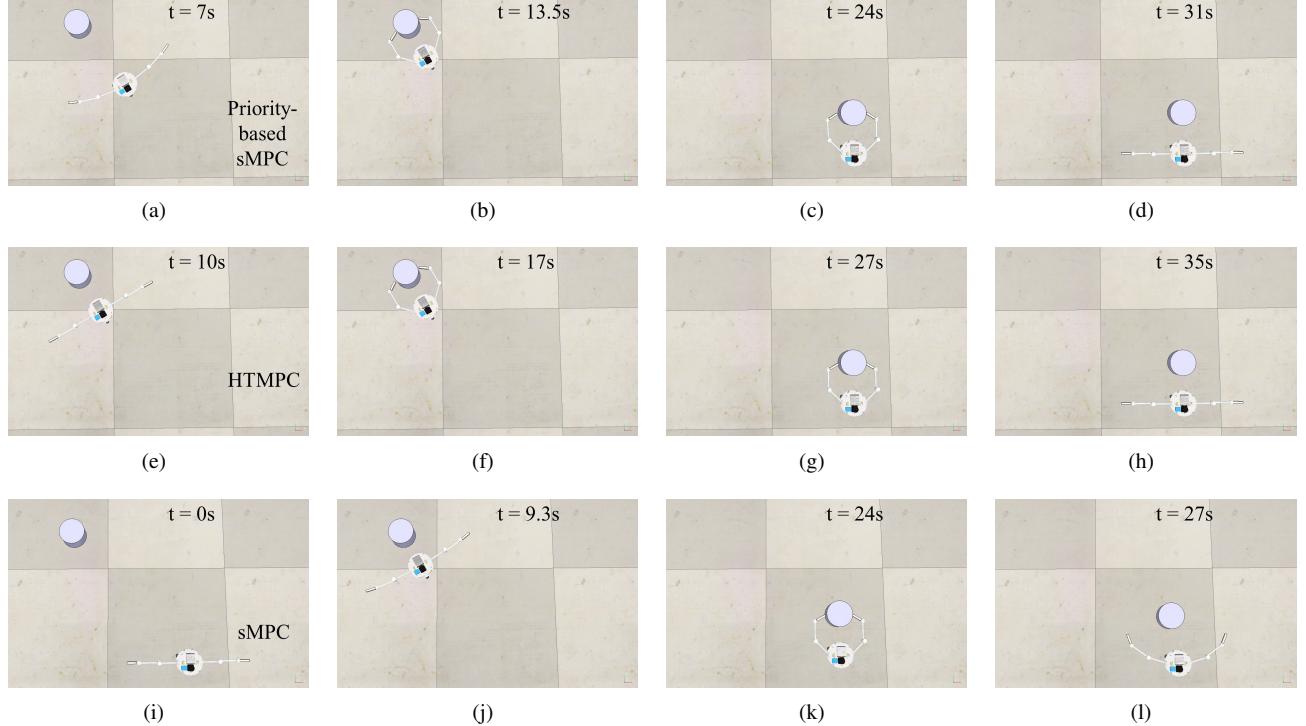


Fig. 5. Snapshots of the on-orbit transportation simulation results around the optimized switching instants t_s and the actual switching instants t_{s_real} (if exists) are compared for the priority-based SMPC (proposed), HTMPC and SMPC. Each row represents snapshots from the same set of simulations at several key moments. In SMPC and priority-based SMPC, switching times are optimized online, whereas in HTMPC, the switching times remain fixed.

(3) Transport σ_{ρ_3} : After successfully capturing the target $\mathbf{x}(t_2) = \xi_{\rho_2}^d$, the space robot moves the object to a desired location

$$\mathbf{Q}^d = \begin{bmatrix} 0 & 0 & 0 \\ \frac{\pi}{4}\text{rad} & \frac{\pi}{4}\text{rad} & \frac{\pi}{3}\text{rad} \\ -\frac{\pi}{4}\text{rad} & -\frac{\pi}{4}\text{rad} & -\frac{\pi}{3}\text{rad} \end{bmatrix}$$

This process is also controlled using the first skill, meaning the controller parameters and constraints are identical to those in σ_{ρ_1} . Since the priority of this stage is lower than that of σ_2 , the weight is designed to be $\omega_3 = 1$.

(4) Release σ_{ρ_4} : Finally, upon reaching the final destination $\mathbf{x}(t_3) = \mathbf{Q}_{\rho_3}^d$, the space robot releases the target object, completing the operation and preparing for new tasks. This process is also controlled using the second skill, similar to σ_{ρ_2} with the following setup:

$$\mathbf{P}_e^d = \begin{bmatrix} 1.5\text{m} & 0\text{m} & 0\text{rad} \\ -1.5\text{m} & 0\text{m} & 0\text{rad} \end{bmatrix}$$

However, in the priority design, this stage is assigned the lowest priority to ensure the successful execution of the more critical tasks preceding it. Therefore, the weight is designed to be $\omega_4 = 10^{-2}$.

B. Priority Analysis

Since the user preferences are explicitly contained in the priority information, traditional methods used for

weight design and analysis in multi-objective optimization are not applicable here. To investigate the impact of introducing priority information on switching MPC, the switching between σ_{ρ_1} and σ_{ρ_2} is used as an example. And the Monte Carlo method is utilized to analyze the influence of different weight combinations on the system performance and optimization solution, providing guidance for a precise design of weight values.

As described in challenge (ii), interactions between different stages can result in the premature initiation of subsequent stages. To quantitatively evaluate the improvement provided by the priority information and prove the optimality analysis, the indicator of the actual start time t_{s_real} is introduced as the time when the control variable \mathbf{u}_{ρ_s} corresponding to stage σ_{ρ_s} exceeds a certain threshold, i.e., $\|\mathbf{u}_{\rho_s}\| \geq \varepsilon$. In this study, the threshold is empirically selected as a small value, 10^{-4} .

The analysis is executed in the space $\omega_1 \times \omega_2 \in [10^{-3}, 10^3] \times [10^{-3}, 10^3]$ with 25×25 points. The results are shown in Fig. 6. It shows a clear trend in the behavior of the actual start times t_{s_real} as the ratio ω_2/ω_1 increases. Specifically, as the ratio increases, the actual start times progressively advance from the optimized switching times t_s , indicating that when a switch is detected within the prediction horizon, the control inputs at the current time are influenced by subsequent stages. This influence is effectively modifiable by adjusting the combination of

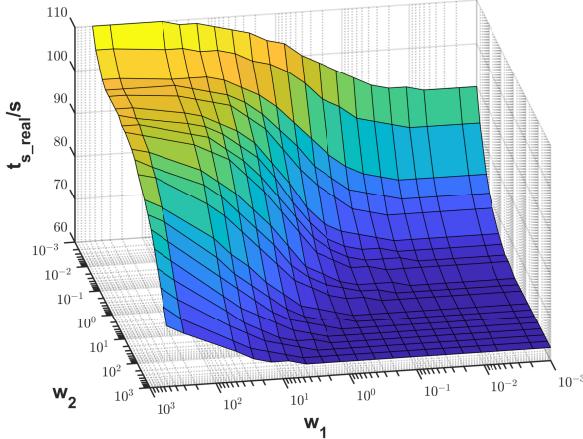


Fig. 6. The impact of weights on the actual start time

weights, demonstrating that the priority design can control the timing of stage transitions.

However, as the values of the weights continue to evolve, t_{s_real} stabilizes around 6.5 seconds. This stabilization occurs because the earliest start time cannot precede the first detection of a switch within the prediction horizon, i.e. $t_{s_real} \geq t_s - T_h$. It also demonstrates that the weights chosen for priority design in this paper are reasonable. The weight combination $[\omega_1, \omega_2] = [1, 10^2]$ can effectively enhance work efficiency while avoiding the decline in control performance that could result from excessively large weights.

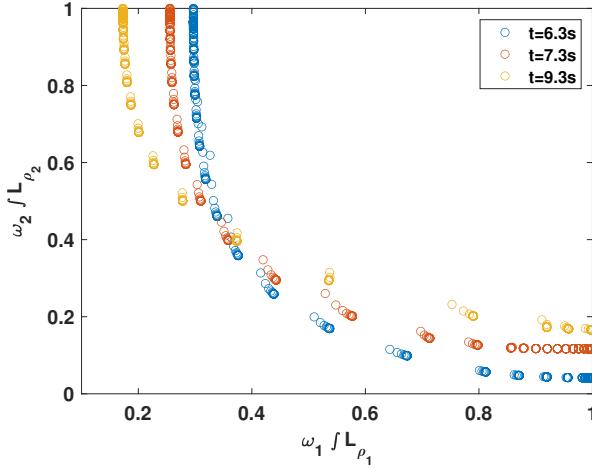


Fig. 7. Pareto optimal set in the criterion space at different sampling time instants

Then, at different sampling time, when switches occur within the prediction horizon, the same set of weights is used to solve the multi-objective optimization problem that simultaneously involves two stages. By calculating and normalizing the cost function values for each stage, the Pareto front as shown in Fig. 7 is obtained. At the same moment, also influenced by the weights, as the ratio ω_2/ω_1 increases, the optimal solution moves towards the upper-left region. Additionally, as the sampling time progresses, the curve also moves towards the upper-

left region, reflecting the impact of the integral horizon $\int_{t_{s-1}}^{t_s}$ or the number of discrete grids N_s on the Pareto optimal solution set, as previously discussed in optimality analysis.

C. Full-procedure Results

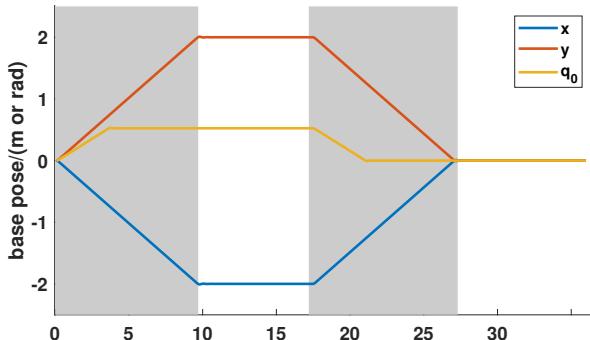
Finally, the on-orbit transportation process of space robot with four stages is utilized to illustrate in detail the main issues investigated in this paper. To further validate the effectiveness and superiority of the proposed method, this study not only compares it with traditional SMPC but also with the strict-priority-based hierarchical task MPC (HTMPC) method recently proposed in [17] for sequential manipulation tasks. Using the task sequence σ also as the priority sequence, the multi-task control problem is formulated as a lexicographic MPC

$$\begin{aligned} J_{\rho_1}^*(\mathbf{x}) &= \min_{\mathbf{u}} \{J_{\rho_1}(\mathbf{x}, \mathbf{u}) \mid \mathbf{u} \in C_p(\mathbf{x})\} \\ J_{\rho_s}^*(\mathbf{x}) &= \min_{\mathbf{u}} \{J_{\rho_s}(\mathbf{x}, \mathbf{u}) \mid J_{\rho_i}(\mathbf{x}, \mathbf{u}) \leq J_{\rho_s}^*(\mathbf{x}), \\ &\quad \mathbf{u} \in C_S(\mathbf{x}), \forall i \in [1, s-1]\} \end{aligned} \quad (32)$$

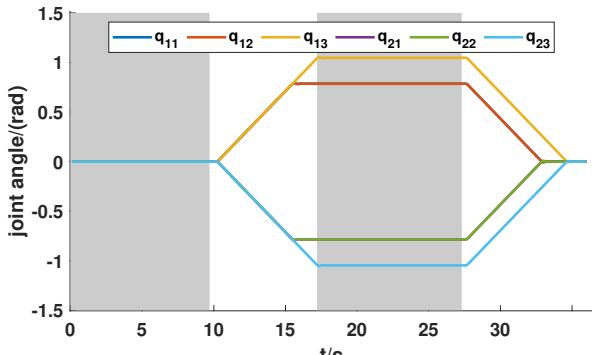
where, $C_S(\mathbf{x})$ is the constraint set as defined before. At each time instant, the lexicographic optimization is performed on the cost functions of all remaining skills in the sequence, and first action in the optimal results is applied as the control input.

REMARK 2 In HTMPC, the priority sequence is treated as a task sequence, where high-priority stages must be executed first and stages must be executed "one by one." Also, this strategy requires solving a lexicographical optimization problem that includes all stages at each moment. In contrast, the method proposed in this paper introduces a soft priority strategy within the switched MPC framework, enabling concurrency between stages. It allows the time and priority sequences to be designed independently. This approach offers greater flexibility in task design and simplifies the optimization process at each step. However, it requires the additional design of a switching optimization method.

Firstly, the HTMPC simulation results around the switching points are shown in Fig. 5(e)-5(h). Based on strict priority, it functions similarly to traditional hard-switching methods under the simulation scenario presented in this paper, where subtasks are executed "one after another" according to a predefined timeline. This behavior arises because the temporal sequence is treated as the priority order, ensuring the performance of the current stage while completely avoiding the influence of subsequent stages. As a result, the whole task is executed successfully in a sequential manner. Therefore, the results of this method are used as a benchmark for further comparison. These characteristics are further illustrated in Fig. 8, where the transitions between the gray and white sections, representing different stages, align perfectly with the inflection points of the curves. Moreover, some important instants in this scenario are:



(a) Base pose



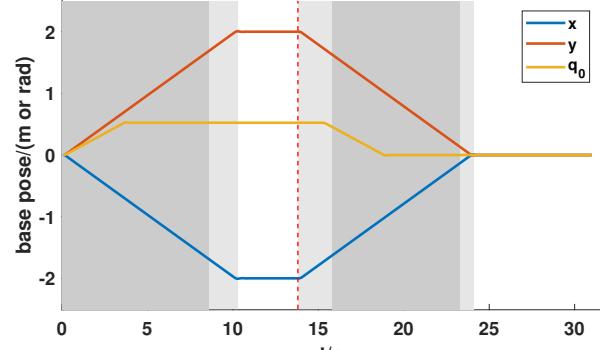
(b) Joint angles

Fig. 8. Control results for HTMPC. The dark gray and white blocks represent the alternate activation of two kind of skills, respectively.

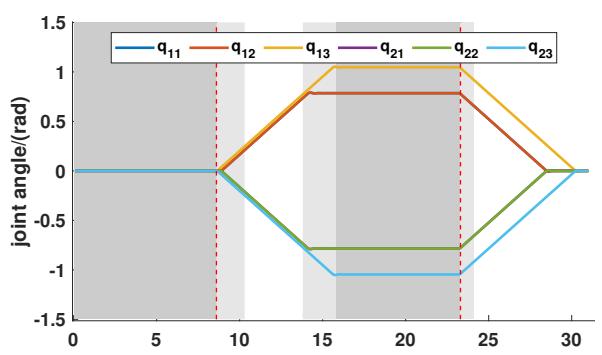
the three switching occur at $t_1 = 9.7s$, $t_2 = 17.2s$, $t_3 = 27.3s$, respectively, which also correspond to the completion times of each stage, and the overall task end at $t_{end} = 35.4s$.

Conversely, in SMPC, this influence exists at every switch. As shown in Fig.5(i)-5(l), the actual start time t_{s_real} always occur earlier than the optimized switching time, resulting in multiple concurrent processes. This is illustrated by the light gray blocks that appear multiple times in Fig.9. This indicates that due to the consideration of subsequent stages within the prediction horizon, skills of the next stages are inevitably initiated before their optimized instants, resulting in overlaps or concurrency between stages. In some cases, such as the Approach-Capture process shown in Fig.5(i) and 5(j), the earlier start of Capture ($t_{real1} = 8.8s$) does not significantly affect the execution of Approach, only slightly delaying its completion time ($t_1 = 10.3s$). This, in turn, advances the completion time of Capture ($t_2 = 15.1s$), improving the overall task efficiency. However, in other cases, like the Transport-Release process shown in Fig.5(k) and 5(l), Release occurs too early, resulting in the object being released before reaching the desired location, which directly leads to mission failure. Thus, it demonstrates the necessity of employing methods to regulate the interactions and impacts between adjacent stages.

Therefore, in the proposed priority-based SMPC, this effect is regulated by the incorporation of priority in-

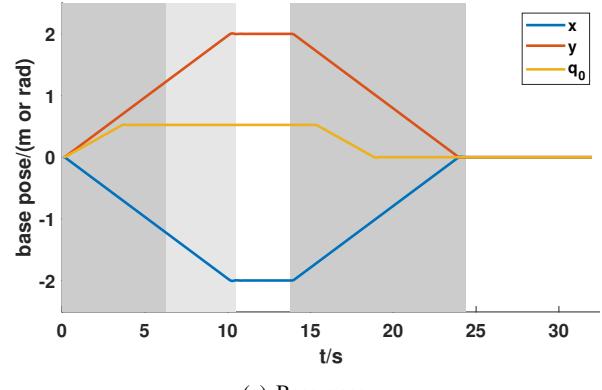


(a) Base pose

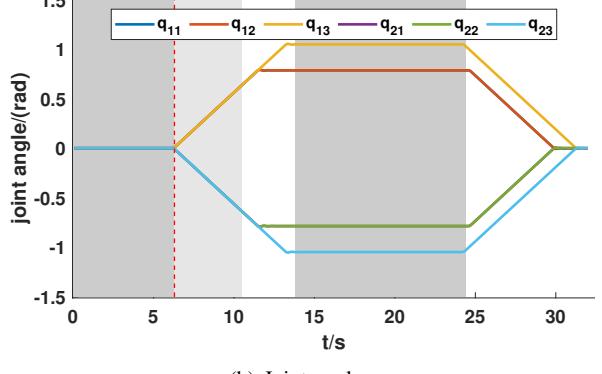


(b) Joint angles

Fig. 9. Control results for SMPC. The additional light gray block represents the concurrent activation of the two skills.



(a) Base pose



(b) Joint angles

Fig. 10. Control results for priority-based SMPC

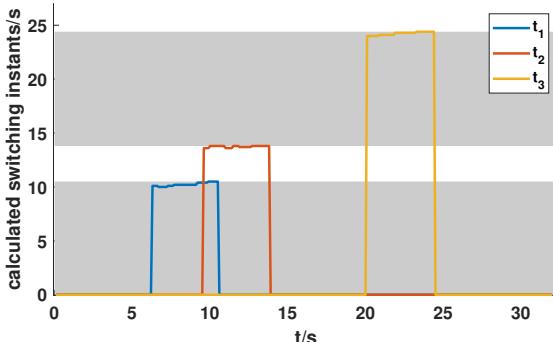


Fig. 11. Optimized switching instants

formation, as shown in Fig.5(a)-5(d) and 10. It ensures that only the Capture is significantly advanced, with its actual start time ($t_{1_real} = 6.3s$) and completion time ($t_2 = 13.8s$) occurring earlier than in traditional SMPC. Meanwhile, the other stages operate as per the optimized results ($t_{2_real} = t_2$ and $t_{3_real} = t_3 = 24.4s$). This regulation is achieved because, near the switching instants, the weights adjust the relative contributions of the cost functions for each stage based on the given priorities. This allows the influence to be amplified when beneficial and avoided when detrimental, thereby improving the smoothness and efficiency of the overall task ($t_{end} = 31.3s$). Finally, the moving switching sequence problem is addressed, and the optimized switching instants in the prediction horizon are shown in Fig.11.

V. CONCLUSION

In this paper, the priority-based switching model predictive control method is proposed to execute the sequential operation tasks of space robots. Initially, the state-dependent SMPC is developed based on a predefined manipulation sequence. The preference-based priority information is then introduced into the task formulation and considered as weights within the SMPC, enabling the regulation rather than mere suppression of interactions between adjacent stages. Additionally, as priority contains user preferences which are challenging to analyze theoretically, Monte Carlo method is employed to discuss the impact on the performance and optimality of the control system. Simulation results have demonstrated the effectiveness of the proposed method.

REFERENCES

- [1] C. YUE, L. LU, Y. WU, Q. CAO, and X. CAO, "Research progress and prospect of the key technologies for on-orbit spacecraft swarm manipulation," *Journal of Astronautics*, vol. 44, no. 6, p. 817, 2023.
- [2] L. Lu, C. Yue, Q. Shen, and X. Cao, "Hierarchical Passivity-Based Force-Position-Configuration Coordinated Control of Multi-Branched Spacecraft," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–16, 2024.
- [3] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.
- [4] M. Bjelonic, R. Grandia, M. Geilinger, O. Harley, V. S. Medeiros, V. Pajovic, E. Jelavic, S. Coros, and M. Hutter, "Offline motion libraries and online MPC for advanced mobility skills," *The International Journal of Robotics Research*, vol. 41, no. 9-10, pp. 903–924, 2022.
- [5] C. Yue, T. Lin, X. Zhang, X. Chen, and X. Cao, "Hierarchical path planning for multi-arm spacecraft with general translational and rotational locomotion mode," *Science China Technological Sciences*, vol. 66, no. 4, pp. 1180–1191, 2023.
- [6] M. Wang, L. Zong, and J. Yuan, "Position/Force Control for a Dual-Arm Space Manipulator Gripping and Transporting an Unknown Target Without Grapple Fixtures," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 2, pp. 1770–1783, 2024.
- [7] G. Tian, B. Li, Q. Zhao, and G. Duan, "High-Precision Trajectory Tracking Control for Free-Flying Space Manipulators With Multiple Constraints and System Uncertainties," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 1, pp. 789–801, 2024.
- [8] Q. An, Y. Zhang, Q. Hu, M. Li, J. Li, and A. Mao, "Time-Optimal Path Tracking for Dual-Arm Free-Floating Space Manipulator System Using Convex Programming," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 5, pp. 6670–6682, 2023.
- [9] G. P. Incremona, A. Ferrara, and L. Magni, "Asynchronous Networked MPC With ISM for Uncertain Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4305–4317, 2017.
- [10] L. Shi, S. Kayastha, and J. Katupitiya, "Robust coordinated control of a dual-arm space robot," *Acta Astronautica*, vol. 138, pp. 475–489, 2017.
- [11] A. Ferrara, G. P. Incremona, and B. Sangiovanni, "Tracking control via switched Integral Sliding Mode with application to robot manipulators," *Control Engineering Practice*, vol. 90, pp. 257–266, 2019.
- [12] T. Gold, A. Volz, and K. Graichen, "Model Predictive Interaction Control for Robotic Manipulation Tasks," *IEEE Transactions on Robotics*, pp. 1–14, 2022.
- [13] X. Xu and P. Antsaklis, "Optimal Control of Switched Systems Based on Parameterization of the Switching Instants," *IEEE Transactions on Automatic Control*, vol. 49, no. 1, pp. 2–16, 2004.
- [14] B. Stellato, S. Ober-Blöbaum, and P. J. Goulart, "Second-Order Switching Time Optimization for Switched Dynamical Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5407–5414, 2017.
- [15] M. Egerstedt, Y. Wardi, and H. Axelsson, "Transition-Time Optimization for Switched-Mode Dynamical Systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 110–115, 2006.
- [16] M. Kamgarpour and C. Tomlin, "On optimal control of non-autonomous switched systems with a fixed mode sequence," *Automatica*, vol. 48, no. 6, pp. 1177–1181, 2012.
- [17] X. Du, S. Zhou, and A. P. Schoellig, "Hierarchical Task Model Predictive Control for Sequential Mobile Manipulation Tasks," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1270–1277, 2024.
- [18] H. Gamper, L. R. Pérez, A. Mueller, A. D. Rosales, and M. Di Castro, "An Inverse Kinematics Algorithm With Smooth Task Switching for Redundant Robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4527–4534, 2024.
- [19] V. Modugno, G. Neumann, E. Rueckert, G. Oriolo, J. Peters, and S. Ivaldi, "Learning soft task priorities for control of redundant robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden: IEEE, 2016, pp. 221–226.
- [20] B. Cai, C. Wei, C. Yue, Y. Yu, X. Chen, and Y. Geng, "Artificial potential incorporated adaptive fixed-time sliding mode control for target capture," *Advances in Space Research*, vol. 72, no. 4, pp. 982–996, 2023.

- [21] I. Dadiotis, A. Laurenzi, and N. Tsagarakis, "Whole-body MPC for highly redundant legged manipulators: Experimental evaluation with a 37 DoF dual-arm quadruped," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8.
- [22] A. Caron, E. Arcari, M. Wermelinger, L. Hewing, M. Hutter, and M. N. Zeilinger, "Data-Driven Model Predictive Control for Trajectory Tracking With a Robotic Arm," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3758–3765, 2019.
- [23] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282–291, 2016.
- [24] H. Mosemann and F. Wahl, "Automatic decomposition of planned assembly sequences into skill primitives," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 709–718, 2001.
- [25] T. Gold, A. Lomakin, T. Goller, A. Volz, and K. Graichen, "Towards a Generic Manipulation Framework for Robots based on Model Predictive Interaction Control," in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*. Beijing, China: IEEE, 2020, pp. 401–406.
- [26] T. Lin, C. Yue, Z. Liu, and X. Cao, "Modular Multi-Level Replanning TAMP Framework for Dynamic Environment," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4234–4241, 2024.
- [27] N. Jaquier, Y. Zhou, J. Starke, and T. Asfour, "Learning to Sequence and Blend Robot Skills via Differentiable Optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8431–8438, 2022.
- [28] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: New insights," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010.
- [29] L. He, H. Ishibuchi, A. Trivedi, H. Wang, Y. Nan, and D. Srinivasan, "A Survey of Normalization Methods in Multiobjective Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 6, pp. 1028–1048, 2021.
- [30] S. Katayama and T. Ohtsuka, "Whole-body model predictive control with rigid contacts via online switching time optimization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Kyoto, Japan: IEEE, 2022, pp. 8858–8865.
- [31] M. Keller, D. Abel, and T. Albin, "Time-Optimal Multi-Stage NMPC for In-Cycle Control of Free-Piston Linear Generators," in *2021 American Control Conference (ACC)*. New Orleans, LA, USA: IEEE, 2021, pp. 4783–4790.
- [32] J. A. E. 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, no. 1, pp. 1–36, 2019.
- [33] S. Katayama, M. Doi, and T. Ohtsuka, "A moving switching sequence approach for nonlinear model predictive control of switched systems with state-dependent switches and state jumps," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 2, pp. 719–740, 2020.
- [34] A. Bemporad and D. Muñoz De La Peña, "Multiobjective model predictive control," *Automatica*, vol. 45, no. 12, pp. 2823–2830, 2009.
- [35] K. Bouyarmane and A. Kheddar, "On Weight-Prioritized Multitask Control of Humanoid Robots," *IEEE Transactions on Automatic Control*, vol. 63, no. 6, pp. 1632–1647, 2018.
- [36] X. Wang, L. Shi, and J. Katupitiya, "A strategy to decelerate and capture a spinning object by a dual-arm space robot," *Aerospace Science and Technology*, vol. 113, p. 106682, 2021.
- [37] Z. Liu, T. Lin, H. Wang, C. Yue, and X. Cao, "Design and Demonstration for an Air-bearing-based Space Robot Testbed," in *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Jinghong, China: IEEE, 2022, pp. 321–326.
- [38] L. Shi, X. Xiao, M. Shan, and X. Wang, "Force control of a space robot in on-orbit servicing operations," *Acta Astronautica*, vol. 193, pp. 469–482, 2022.



Ziran Liu received the B.S. degree in space-craft design and engineering from Harbin Institute of Technology, Harbin, China in 2020. He is currently pursuing his Ph.D. degree in the School of Astronautics at Harbin Institute of Technology, Harbin, China.

His research interests include spacecraft attitude control system design, model predictive control and space robotics.



Zijian Dai received the B.S. degree in automation from Harbin Engineering University, Harbin, China, in 2022, and the M.S. degree in aeronautical and astronautical science and technology from Harbin Institute of Technology, Shenzhen, China, in 2025. He is currently working toward the Ph.D. degree in robotics at the Department of Computer, Control and Management Engineering, Sapienza University of Rome, Rome, Italy.

His current research interests include whole-body motion control and multi-contact optimization for robotic systems.



Chengfei Yue (Member, IEEE) received the B.E. degree from Harbin Institute of Technology, China, in 2013, and Ph.D. degree from the National University of Singapore in 2019. He is a professor of Harbin Institute of Technology Shenzhen (HITSZ), and the Lab Director of Advanced Space Application and Intelligent Control (ASAIC Lab).

He is also a committee member of the Committee of Space Intelligence, Chinese Society of Space Research, and steering committee member of IFAC TC 7.3. He is also the NOC Chair of upcoming IFAC ACA 2025.

His current research interests including: spacecraft dynamics and control, intelligent planning and control. He has published over 60 technical papers and articles in international refereed journals and conferences.



Tao Lin received the B.S. degree in aircraft design and engineering from the Harbin Institute of Technology, Harbin, China, in 2021. He is currently working toward the Ph.D. degree in aeronautical and astronautical science and technology with the School of Aerospace, Harbin Institute of Technology, Harbin, China.

His research interests include task and motion planning, automated manufacturing systems, and on-orbit assembly.



Antonella Ferrara (Fellow, IEEE) received the M.Sc. degree in electronic engineering and the Ph.D. degree in computer science and electronics from the University of Genoa, Italy, in 1987 and 1992, respectively. Since 2005, she has been a Full Professor of automatic control with the University of Pavia, Italy. She is the author and co-author of more than 450 publications, including more than 170 journal papers, two monographs, and one edited book.

She is a member of the IEEE Control Systems Society and of the IEEE

Intelligent Transportation Systems Society. She is an IEEE Fellow, IFAC Fellow and AAIA Fellow.



Xibin Cao received the B.S., M.S., and Ph.D. degrees in forging and stamping from Harbin Institute of Technology, Harbin, China, in 1985, 1988, and 1991, respectively.

Since 1991, he has been with the School of Astronautics, Harbin Institute of Technology, where he is currently a Full Professor. Since 2009, he has been the Dean of the Astronautics School, Harbin Institute of Technology, and since 2015, an Assistant President with Harbin Institute of Technology.

Dr. Cao was the Distinguished Professor of Yangtze River Scholar, Ministry of Education of China, in 2005. He was an Academician of the Chinese Academy of Engineering.