

Priority-Driven Task Allocation Framework for Robotic On-Orbit Servicing

Peter Kötting*

On-Orbit Servicing (OOS) missions performed by orbital space robots require an optimal and safe system behavior in order to achieve a full stack of multiple tasks. However, OOS tasks may conflict with each other at some particular operational points, especially in case of uncooperative targets with prior unknown motion states and geometry. To address these challenges, we propose a novel task-priority framework for optimal (local) allocation and prioritization of tasks according to user-defined metrics by exploiting the system redundancy within an online algorithm. Our approach uses *equality tasks* to track state trajectories for the end-effector and optionally also for the servicer-base, and *set-based tasks* to ensure safety and system constraints such as collision avoidance, the field of view, and manipulator workspace limits. We prove the asymptotic convergence of the equality task errors together with the satisfaction of all set-based tasks. Software-in-the-loop results with an end-to-end simulation test facility show the reliability of the proposed approach aiming to enable more efficient and reliable on-orbit servicing missions.

I. Nomenclature

$\{S\}$	=	Servicer-base frame (Figure 1)
$\{E\}$	=	End-effector frame (Figure 1)
$\{T\}$	=	Target object frame (Figure 1)
$\{R\}$	=	Orbital reference frame placed at the target origin
p_{XY}	=	Position of the origin of frame $\{Y\}$ relative to frame $\{X\}$
q_{XY}	=	Unit quaternion $q = [\epsilon^T \ \eta]^T \in \mathbb{S}^3$, with $\mathbb{S}^3 := \{x \in \mathbb{R}^4 : x^T x = 1\}$, orientation of frame $\{Y\}$ relative to frame $\{X\}$
R_{XY}	=	Rotation matrix $R_{XY} \in \mathfrak{so}(3)$ representing the orientation of frame $\{Y\}$ relative to frame $\{X\}$
v_{XY}	=	Translational velocity $v_{XY} \in \mathbb{R}^3$ of frame $\{Y\}$ relative to frame $\{X\}$, expressed in frame $\{X\}$
ω_{XY}	=	Angular velocity $\omega_{XY} \in \mathbb{R}^3$ of frame $\{Y\}$ relative to frame $\{X\}$, expressed in frame $\{X\}$
$S(\cdot)$	=	Skew-symmetric operator $S(\cdot) : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$, defined by $S(x)y = x \times y$ for any $x, y \in \mathbb{R}^3$
I_n	=	Identity matrix of dimension n
$0_{n \times m}$	=	Zero matrix with n rows and m columns

II. INTRODUCTION

Robotic systems play a pivotal role in advancing space exploration and utilization by enabling On-Orbit Servicing (OOS). This technology is indispensable for accomplishing a diverse set of mission tasks, such as satellite inspection, refueling, upgrades, repairs, debris removal, maintenance, and orbital structure construction. These tasks often involve high costs and risks, or are deemed infeasible when conducted by astronauts through extravehicular activities. The typical actuator configuration for an orbital robot includes joint drivetrains and spacecraft actuators such as thrusters (reaction control system) and momentum wheels.

Early control strategies for orbital robotics primarily focused on commanding the manipulator alone while disabling the spacecraft's actuators - a method known as the free-floating approach [1, 2]. This strategy offers the clear advantage of reduced fuel consumption. OOS missions typically require precise attitude control of the spacecraft due to constraints such as the limited field of view of onboard sensors used for relative navigation - an aspect not

*Research Associate, Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Peter.Koetting@dlr.de.

addressed by free-floating approaches. Moreover, free-floating schemes are inherently limited by dynamic singularities and a restricted operational workspace, as the spacecraft's pose remains uncontrolled. To overcome these limitations, free-flying control strategies have been developed. In this paradigm, all spacecraft-base actuators and the manipulator's joint drivers - are coordinated simultaneously [3]–[6]. In [3], an adaptive control scheme is presented, whereas [4] employs a feedback linearization-based approach to control both the spacecraft's attitude and the end-effector's pose. This method does not stabilize translational motion, which may lead to drift during contact and limit workspace coverage. The coordinate control approach in [5] utilizes a transposed Jacobian method to govern all actuators effectively. An alternative strategy is proposed in [6], which focuses on regulating the translation of the system center of mass (CoM). This enables free translation of the geometric center and decouples control tasks, thereby reducing reliance on thruster actuation. Another critical aspect in OOS missions is the multi-rate nature of the control system, an issue not addressed by the aforementioned approaches. Due to the limitations of base actuation and Guidance, Navigation, and Control (GNC) systems, the servicer-base typically operates at a low control frequency (1–10 Hz), while the manipulator operates at a significantly higher frequency (around 1000 Hz). The multi-rate tracking control scheme introduced in [7] addresses this disparity using a time-domain passivity framework. All of these coordinated control approaches rely on a suitable trajectory planning of both the end-effector and the spacecraft base

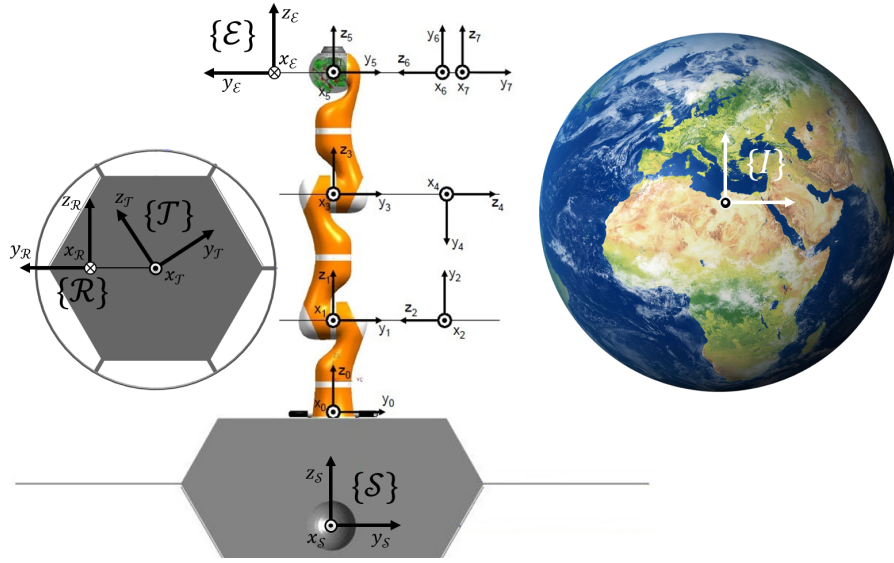


Fig. 1 OOS coordinate frames: inertial frame $\{I\}$, end-effector frame $\{E\}$, servicer-base frame $\{S\}$, target object frame $\{T\}$, orbital reference frame $\{R\}$, and manipulator link frames $\{0\}, \dots, \{7\}$.

However, trajectory planning either considers the end-effector only [4], where a reference for the servicer-base is not considered, or it considers the robotic system dynamics with system and safety constraints, resulting in a nonlinear optimization problem with very high computational demand [8], which needs to be partially solved offline comprising a limited set of tumbling target states and target geometry.

We address this issue by considering the system from a control allocation point of view. Whereby, the proposed allocation computes online optimal (local) actuator states for the whole system by tracking an end-effector trajectory, enabling an unlimited workspace while explicitly satisfying system and safety constraints. Hence, the trajectory planner is designed to compute an end-effector trajectory only, without considering system details, as done in [4]. Accordingly, the guidance and control algorithms can be solved fully online, enabling the full autonomous operation of the system. Whereas the model predictive control based coordinate control approach [12] solves the complete constrained control allocation problem as a nonlinear program, they are typically computationally intensive. In contrast, the proposed method offers a computational lightweight solution for real-time implementation on onboard computers, without imposing stringent requirements on processing capabilities.

The structure of the paper is as follows: The structure of the paper is as follows: Section III introduces our proposed computationally efficient closed-loop control allocation framework for free-flying orbital robots. Section IV provides a proof of asymptotic stability for the equality tasks, using an attitude representation based on unit quaternions. Section V presents software-in-the-loop simulation results, and Section VI concludes the paper.

III. Closed-Loop Control Allocation

A. Control Allocation Setup

We propose a control allocation framework, as shown in Fig. 2 (blue dashed), enabling a combination of guidance with low-level control approaches. The objective is to ensure compliance with constraints expressed in terms of set-based tasks by selecting a particular inverse kinematic (IK) in the control allocation that satisfies the prescribed tasks in Table 2 based on their respective priority order.

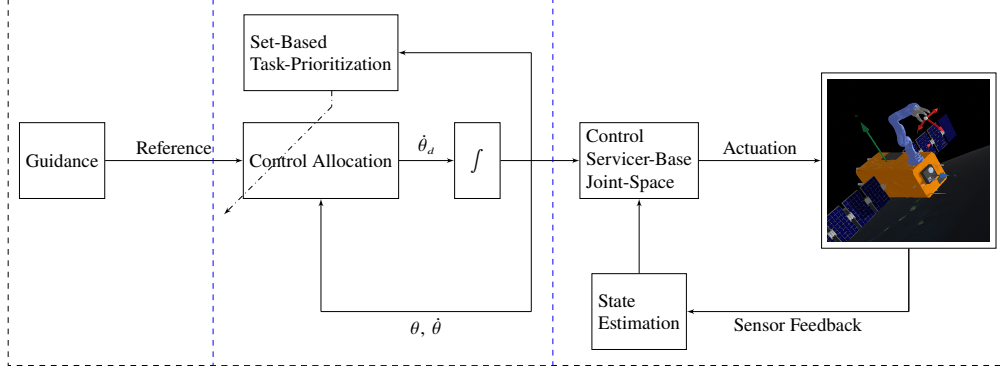


Fig. 2 GNC structure, with inner control loops and an outer control allocation loop.

Safety and system constraints must be primarily satisfied, so we consider these as high-priority set-based tasks, i.e., all set-based tasks have a higher priority than the highest equality task. We specify set-based tasks by their valid sets C in the following priority order: (i) Collision avoidance, involves ensuring that the robot does not collide with the target specified by a safety sphere with the minimal radius c_{a1} ; (ii) Field of view, involves ensuring that the robot's sensors (cameras, LiDAR), have an unobstructed view of the target important for object recognition, navigation, or surveillance, which is specified by the maximal vector error c_{b1} ; (iii) Joint limits, involves ensuring that robot joint angles remain in the valid set C_c ; (iv) Manipulator workspace limits, involves ensuring that the manipulator, remains within a specified workspace, defined as the set of points within the intersection of two spheres with the minimal and maximal radii of C_d and C_e , the end-effector can reach. Primarily, we integrate an end-effector trajectory as an equality task to be tracked

Table 2 Task overview

σ	Task description	Type	valid set C
σ_a	Collision Avoidance (CA)	Set-based	$C_a = [c_{a1}, \infty)$
σ_b	Field of View (FoV)	Set-based	$C_b = [0, c_{b2}]$
σ_c	Joint Limit (JL)	Set-based	$C_c = [c_{c1}, c_{c2}]$
σ_d	Workspace Limit I (WLI)	Set-based	$C_d = [c_{d1}, c_{d2}]$
σ_e	Workspace Limit II (WLII)	Set-based	$C_e = [c_{e1}, c_{e2}]$
σ_1	End-effector pose	Equality	-
σ_2	Servicer-base pose	Equality	-

since this is the major concern for the reach and grasping phase. As a secondary equality task, we include tracking of the servicer-base, since this is only considered optional, but can be necessary depending on the operation phase, e.g., for the final approach.

For the system in free-flying mode, we define an actuator space velocity vector by

$$\dot{\theta} = \begin{pmatrix} \dot{\theta}_b & \dot{\theta}_m \end{pmatrix}^T \in \mathbb{R}^{n_s}, \quad (1)$$

with $n_s = 6 + n$ DoF, which includes the manipulator joint velocities $\dot{\theta}_m \in \mathbb{R}^n$ and the servicer-base velocity $\dot{\theta}_b = (v_{\mathcal{RS}} \quad \omega_{\mathcal{RS}})^T \in \mathbb{R}^6$, with the linear and angular components respectively. By time integration and the quaternion propagation (2) we obtain all states θ . Based on integration of a feedback loop we avoid drifting of the allocated values.

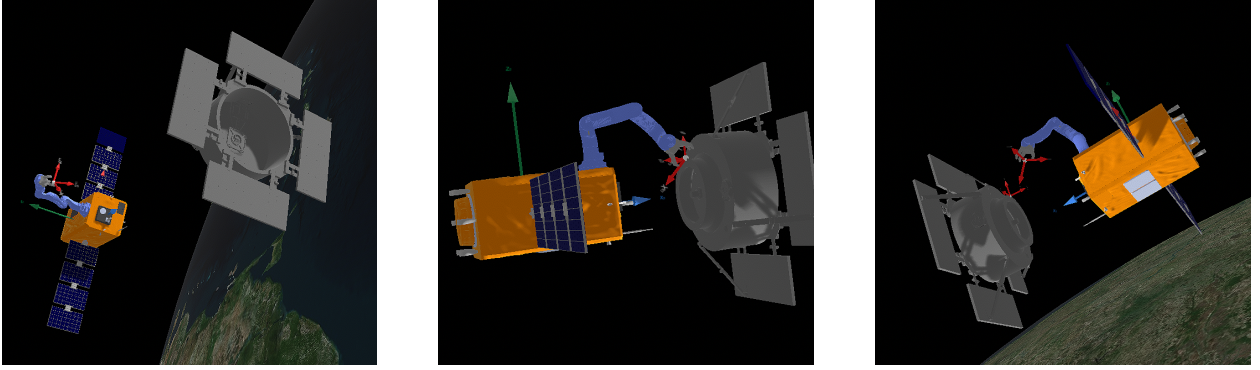


Fig. 3 Servicer and Target during OOS Phases: (I) Approach (left), (II) Reach (center), (III) Escape (right)

B. System Kinematics

This section establishes the fundamental kinematic relationships of the system, which form the basis for the proposed control allocation strategy. The orientation of bodies is represented using unit quaternions, defined as q , which provide a singularity-free and compact parametrization of rotations. Their propagation is given by

$$\dot{q} = \frac{1}{2} \Xi(q) \omega, \quad \Xi(q) = \begin{pmatrix} \eta I_3 + S(\epsilon) \\ -\epsilon^T \end{pmatrix} \in \mathbb{R}^{4 \times 3}, \quad (2)$$

with $S(\cdot) : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ the skew-symmetric operator. For a manipulator mounted on the servicer, the end-effector twist relative to the base is

$$\begin{pmatrix} v_{SE} \\ \omega_{SE} \end{pmatrix} = J_m(\theta_m) \dot{\theta}_m, \quad J_m(\theta_m) \in \mathbb{R}^{6 \times n}. \quad (3)$$

The end-effector twist relative to the orbital reference frame is obtained by rigid-body velocity composition

$$v_{RE} = v_{RS} - S(R_{RS} p_{SE}) \omega_{RS} + R_{RS} v_{SE}, \quad (4)$$

$$\omega_{RE} = \omega_{RS} + R_{RS} \omega_{SE}. \quad (5)$$

This yields the full system Jacobian $J_1 = (A_1 \ A_2 J_m) \in \mathbb{R}^{6 \times n_s}$ with

$$A_1 = \begin{pmatrix} I_3 & -S(R_{RS} p_{SE}) \\ 0_3 & I_3 \end{pmatrix}, \quad A_2 = \begin{pmatrix} R_{RS} & 0_3 \\ 0_3 & R_{RS} \end{pmatrix}. \quad (6)$$

Finally, the Jacobian relations for the primary (end-effector) and secondary (servicer-base) tasks are

$$\begin{pmatrix} v_{RE} \\ \omega_{RE} \end{pmatrix} = J_1(\theta) \dot{\theta}, \quad \begin{pmatrix} v_{RS} \\ \omega_{RS} \end{pmatrix} = J_2(\theta) \dot{\theta}, \quad \text{with } J_2 = (I_6 \ 0_{6 \times n}) \in \mathbb{R}^{6 \times n_s}. \quad (7)$$

C. Equality Tasks

Let us define the pose of the end-effector σ_1 and servicer-base σ_2 by the forward kinematics

$$\sigma_1 = \begin{pmatrix} p_{\mathcal{RE}} \\ q_{\mathcal{RE}} \end{pmatrix} = f_1(\theta), \quad \sigma_2 = \begin{pmatrix} p_{\mathcal{RS}} \\ q_{\mathcal{RS}} \end{pmatrix} = f_2(\theta), \quad (8)$$

as the task variables to be tracked. Since unit quaternions are used for the attitude representation, the respective control errors are given by

$$\tilde{p} = p - p_d, \quad \tilde{q} = \begin{pmatrix} \tilde{\epsilon} \\ \tilde{\eta} \end{pmatrix} = \begin{pmatrix} \Xi^T(q_d) \\ q_d^T \end{pmatrix} q, \quad (9)$$

where the subscript d indicates desired values. Accordingly, the end-effector tracking error $\tilde{\sigma}_1 \in \mathbb{R}^6$ and the servicer-base tracking error $\tilde{\sigma}_2 \in \mathbb{R}^6$ are defined by

$$\tilde{\sigma}_1 = \begin{pmatrix} \tilde{p}_{\mathcal{RE}} \\ \tilde{\epsilon}_{\mathcal{RE}} \end{pmatrix}, \quad \tilde{\sigma}_2 = \begin{pmatrix} \tilde{p}_{\mathcal{RS}} \\ \tilde{\epsilon}_{\mathcal{RS}} \end{pmatrix}. \quad (10)$$

We define the control inputs $v_1 \in \mathbb{R}^6$, $v_2 \in \mathbb{R}^6$ by

$$v_1 = \begin{pmatrix} v_{\mathcal{RE}} \\ \omega_{\mathcal{RE}} \end{pmatrix} - K_1 \tilde{\sigma}_1, \quad v_2 = \begin{pmatrix} v_{\mathcal{RS}} \\ \omega_{\mathcal{RS}} \end{pmatrix} - K_2 \tilde{\sigma}_2, \quad (11)$$

where the first term can be considered as a feedforward term and $K_1 = k_1 I_6 \in \mathbb{R}^{6 \times 6}$, $K_2 = k_2 I_6 \in \mathbb{R}^{6 \times 6}$ are the feedback matrices. For the regulation case, the time derivative of the pose tracking error is given by

$$\dot{\tilde{\sigma}}_1 = T(\tilde{q}_{\mathcal{RE}}) J_1 \dot{\theta}, \quad \dot{\tilde{\sigma}}_2 = T(\tilde{q}_{\mathcal{RS}}) J_2 \dot{\theta}, \quad (12)$$

$$\text{with } T(\tilde{q}) = \begin{pmatrix} I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & \frac{1}{2}(S(\tilde{\epsilon}) + \tilde{\eta} I_3) \end{pmatrix}. \quad (13)$$

D. Set-based Tasks

Each set-based task is based on a scalar error σ_i and its corresponding Jacobian relationship $\dot{\sigma}_i = J_i \dot{\theta}$ with $i \in \{a, b, c, d, e\}$. Firstly, we consider collision avoidance of the servicer with a target by the euclidean distance σ_a between the servicer position $p_{\mathcal{RS}}$ and target position $p_{\mathcal{RT}}$, i.e.,

$$\sigma_a = \|p_{\mathcal{RT}} - p_{\mathcal{RS}}\|, \quad J_a := -\frac{(p_{\mathcal{RT}} - p_{\mathcal{RS}})^T}{\sigma_a} (I_{3 \times 3} \quad 0_{3 \times (3+n)}). \quad (14)$$

Secondly, the field-of-view is integrated by the error between a sensor direction vector $a = R_{\mathcal{RS}} e_1 \in \mathbb{R}^3$ with $e_1 = [1 \ 0 \ 0]^T$ and its desired counterpart $a_d \in \mathbb{R}^3$. Accordingly, we derive the task

$$\sigma_b = \|a_d - a\|, \quad J_b := -\frac{(a_d - a)^T}{\sigma_b + \nu} (0_{3 \times 3} \quad -R_{\mathcal{RS}} [e_1 \times] e_1 \quad 0_{3 \times n}), \quad (15)$$

with small value $\nu > 0$ to avoid a singularity at $\sigma_b = 0$. Thirdly, joint-space limits are trivially implemented. For instance, the first robot joint is limited by

$$\sigma_c = (0_{1 \times 6} \quad 1 \quad 0_{1 \times 6}) \theta, \quad J_c := (0_{1 \times 6} \quad 1 \quad 0_{1 \times 6}) \in \mathbb{R}^{1 \times n_s}. \quad (16)$$

Finally, we consider workspace limits. Two tasks specified by the euclidean distances approximate the workspace by

$$\sigma_d = \|p_{s2} - p_{sE}\|, \quad \sigma_e = \|p_{se} - p_{sE}\|, \quad (17)$$

with some user defined position vector p_{se} and the respective Jacobians by

$$J_d := -\frac{((p_{s2} - p_{sE})^T \quad 0_{1 \times 3})}{\sigma_d} (0_{6 \times 6} \quad J_m), \quad J_e := -\frac{((p_{se} - p_{sE})^T \quad 0_{1 \times 3})}{\sigma_e} (0_{6 \times 6} \quad J_m). \quad (18)$$

E. Inverse Mapping

Considering the problem of following a desired end-effector pose trajectory $\sigma_{1d} \in \mathbb{R}^m$, the joint references $\theta_d \in \mathbb{R}^{n_s}$ needs to be computed. This can be done for redundant systems $n_s > m$ by solving a weighted least-squares problem and imposing minimum-norm velocity relative to an arbitrary null space velocity $\dot{\theta}_{0d} \in \mathbb{R}^{n_s}$, leading to

$$\dot{\theta}_d = J_1^\dagger \dot{\sigma}_{1d} + (I_{n_s} - J_1^\dagger J_1) \dot{\theta}_{0d}, \quad (19)$$

where $J_1^\dagger := W^{-1} J_1^T (J_1 W^{-1} J_1^T)^{-1} \in \mathbb{R}^{n_s \times m}$ defines the right pseudoinverse of J_1 with the weighting matrix $W \in \mathbb{R}^{n_s \times n_s}$. It includes the null space projector

$$N = (I_{n_s} - J_1^\dagger J_1) \in \mathbb{R}^{n_s \times n_s}, \quad (20)$$

where N projects $\dot{\theta}_{0d}$ in the null space of the Jacobian matrix. This can be interpreted in such a way that a generated movement $\dot{\theta}_{0d}$ of the robotic actuator DoF does not affect the given task of the end-effector. The concept of null space projection is exploited for multiple tasks in the following.

F. Task Prioritization

The computation of multiple prioritized equality tasks can be performed by the augmented inverse-based projection [9]

$$\dot{\theta}_d = J_1^\dagger v_1 + N_1 J_2^\dagger v_2 + \dots + N_{12\dots(k-1)}^\dagger J_k^\dagger v_k, \quad (21)$$

with the number of equality tasks k adapted by our control inputs (11) where

$$N_{12\dots(k-1)}^\dagger = (I_{n_s} - J_{12\dots(k-1)}^\dagger J_{12\dots(k-1)}) \quad (22)$$

defines the common null-space projector. The combined tasks are described by the augmented Jacobian

$$J_{12\dots(k-1)}^A = \begin{pmatrix} J_1 \\ J_2 \\ \dots \\ J_{k-1} \end{pmatrix}. \quad (23)$$

In the following, we expand this method for set-based tasks. A set-based task with index $i \in \{a, b, c, d, e\}$ is activated, if σ_i is outside of its desired valid set C_i and its time derivative $\dot{\sigma}_i$ moves even further. Every mode is defined, as a representation of a specific combination of set-based tasks, which are activated. For compactness, we formulate the CLIK and its system dynamics for the first $j = 2$ set-based tasks σ_a and σ_b resulting in $2^j = 4$ modes. Accordingly, this can be extended following the same systematic for all $j = 5$ set-based tasks, which results in $2^j = 32$ modes as shown in the simulation results. Based on the augmented inverse-based projection (21) and its null-space projections (22) including the set-based tasks in k we derive our IK with Table 2, which are summarized in Table 3. For further methodical details see [10]. Mode 1 represents the case when all states stay within the valid set C and the IK results by

Table 3 IK-Equations

Mode i	Set-based tasks	Equations $\dot{\theta}_d = g_i$
1	no activation	$g_1 = J_1^\dagger v_1 + N_1 J_2^\dagger v_2$
2	σ_a active	$g_2 = N_a J_1^\dagger v_1 + N_{a1} J_2^\dagger v_2$
3	σ_b active	$g_3 = N_b J_1^\dagger v_1 + N_{b1} J_2^\dagger v_2$
4	σ_a and σ_b active	$g_4 = N_{ab} J_1^\dagger v_1 + N_{ab1} J_2^\dagger v_2$

just using the equality tasks. The collision avoidance mode 2 results by adding the equality task $\dot{\sigma}_a = J_a \dot{\theta} = 0$ as the highest priority to keep at least distance to the target. We derive the field of view mode 3 by adding an equality task $\dot{\sigma}_b = J_b \dot{\theta} = 0$ as the highest priority to avoid increasing the attitude error above the limits. Finally, mode 4 combines the set-based tasks σ_a and σ_b , which is achieved by adding both before mentioned equations in the priority order according to Table 2.

G. System Dynamics

A common construction in CLIK is to feed back the desired states, thereby keeping the kinematic loop separate from the low-level dynamic loop, i.e., $\dot{\theta}_d = \dot{\theta}$, as shown in Fig. 2. The dynamics depends on the particular mode, which is determined based on the state vector

$$z = \begin{pmatrix} \theta^T & \sigma_a^T & \sigma_b^T & \tilde{\sigma}^T \end{pmatrix}^T, \quad (24)$$

$\tilde{\sigma} = [\tilde{\sigma}_1^T \quad \tilde{\sigma}_2^T]^T$ with respect to the valid set

$$C := \mathbb{R}^{n_s} \times C_a \times C_b \times \mathbb{R}^{12}. \quad (25)$$

We consider the regulation case in the system dynamics analysis. For mode 1 it results in the system dynamics

$$\dot{\theta} = J_1^\dagger v_1 + N_1 J_2^\dagger v_2, \quad (26)$$

$$\dot{\sigma}_a = J_a \dot{\theta} \stackrel{(26)}{=} J_a (J_1^\dagger v_1 + N_1 J_2^\dagger v_2), \quad (27)$$

$$\dot{\sigma}_b = J_b \dot{\theta} \stackrel{(26)}{=} J_b (J_1^\dagger v_1 + N_1 J_2^\dagger v_2), \quad (28)$$

$$\dot{\tilde{\sigma}} = \begin{pmatrix} \dot{\tilde{\sigma}}_1 \\ \dot{\tilde{\sigma}}_2 \end{pmatrix} \stackrel{(12)}{=} \begin{pmatrix} T J_1 \\ T J_2 \end{pmatrix} \dot{\theta} \stackrel{(26)(11)}{=} - \begin{pmatrix} T K_1 & T J_1 N_1 J_2^\dagger K_2 \\ T J_2 J_1^\dagger K_1 & T J_2 N_1 J_2^\dagger K_2 \end{pmatrix} \begin{pmatrix} \tilde{\sigma}_1 \\ \tilde{\sigma}_2 \end{pmatrix} =: -M_1 \tilde{\sigma}, \quad (29)$$

$$\dot{z} = \begin{pmatrix} \dot{\theta} \\ \dot{\sigma}_a \\ \dot{\sigma}_b \\ \dot{\tilde{\sigma}} \end{pmatrix} =: \begin{pmatrix} f_{11}(z) \\ f_{12}(z) \\ f_{13}(z) \\ f_{14}(z) \end{pmatrix} =: F_1(z), \quad (30)$$

with $J_1 J_1^\dagger = I_6$. In mode 1 no set-based task is active, thus, no equality tasks is affected. Hence, the state trajectory of z follows $F_1(z)$ as long as z stays in C . If z leaves C , the mode is switched in dependence of which set-based task is violated. In case it leaves the set C_a , i.e., $\sigma_a \leq c_{a_1}$ and $\dot{\sigma}_a = f_{12}(z) < 0$, mode 2 gets activated and with the IK $\dot{\theta} = g_2$ it results in the system dynamics

$$\dot{\theta} = N_a J_1^\dagger v_1 + N_{a1} J_2^\dagger v_2 \quad (31)$$

$$\dot{\sigma}_a = J_a \dot{\theta} \stackrel{(31)}{=} J_a (N_a J_1^\dagger v_1 + N_{a1} J_2^\dagger v_2) = 0 \quad (32)$$

$$\dot{\sigma}_b = J_b \dot{\theta} \stackrel{(31)}{=} J_b (N_a J_1^\dagger v_1 + N_{a1} J_2^\dagger v_2) \quad (33)$$

$$\dot{\tilde{\sigma}} \stackrel{(12)(31)(11)}{=} - \begin{pmatrix} T J_1 N_a J_1^\dagger K_1 & T J_1 N_{a1} J_2^\dagger K_2 \\ T J_2 N_a J_1^\dagger K_1 & T J_2 N_{a1} J_2^\dagger K_2 \end{pmatrix} \begin{pmatrix} \tilde{\sigma}_1 \\ \tilde{\sigma}_2 \end{pmatrix} =: -M_2 \tilde{\sigma} \quad (34)$$

$$\dot{z} = \begin{pmatrix} \dot{\theta} \\ \dot{\sigma}_a \\ \dot{\sigma}_b \\ \dot{\tilde{\sigma}} \end{pmatrix} =: \begin{pmatrix} f_{21}(z) \\ 0 \\ f_{23}(z) \\ f_{24}(z) \end{pmatrix} =: F_2(z). \quad (35)$$

If z leaves C_b , i.e., $\sigma_b \geq c_{b_2}$ and $\dot{\sigma}_b = f_{i3}(z) > 0$, mode 3 gets activated and based on the IK $\dot{\theta} = g_3$ it results in the system dynamics

$$\dot{\theta} = N_b J_1^\dagger v_1 + N_{b1} J_2^\dagger v_2 \quad (36)$$

$$\dot{\sigma}_a = J_a \dot{\theta} \stackrel{(36)}{=} J_a (N_b J_1^\dagger v_1 + N_{b1} J_2^\dagger v_2) \quad (37)$$

$$\dot{\sigma}_b = J_b \dot{\theta} \stackrel{(36)}{=} J_b (N_b J_1^\dagger v_1 + N_{b1} J_2^\dagger v_2) = 0 \quad (38)$$

$$\dot{\tilde{\sigma}} \stackrel{(12)(36)(11)}{=} - \begin{pmatrix} T J_1 N_b J_1^\dagger K_1 & T J_1 N_{b1} J_2^\dagger K_2 \\ T J_2 N_b J_1^\dagger K_1 & T J_2 N_{b1} J_2^\dagger K_2 \end{pmatrix} \begin{pmatrix} \tilde{\sigma}_1 \\ \tilde{\sigma}_2 \end{pmatrix} =: -M_3 \tilde{\sigma} \quad (39)$$

$$\dot{z} = \begin{pmatrix} \dot{\theta} \\ \dot{\sigma}_a \\ \dot{\sigma}_b \\ \dot{\tilde{\sigma}} \end{pmatrix} =: \begin{pmatrix} f_{31}(z) \\ f_{33}(z) \\ 0 \\ f_{34}(z) \end{pmatrix} = F_3(z). \quad (40)$$

In case z is leaving C_a and C_b , i.e., if $\sigma_a \leq c_{a1}$, $\dot{\sigma}_a = f_{i2}(z) < 0$, $\sigma_b \geq c_{b2}$, and $\dot{\sigma}_b = f_{i3}(z) > 0$, then mode 4 gets activated and with the IK $\dot{\theta} = g_4$ it results in the system dynamics

$$\dot{\theta} = N_{ab} J_1^\dagger v_1 + N_{ab1} J_2^\dagger v_2 \quad (41)$$

$$\dot{\sigma}_a = J_a \dot{\theta} \stackrel{(41)}{=} J_a (N_{ab} J_1^\dagger v_1 + N_{ab1} J_2^\dagger v_2) = 0 \quad (42)$$

$$\dot{\sigma}_b = J_b \dot{\theta} \stackrel{(41)}{=} J_b (N_{ab} J_1^\dagger v_1 + N_{ab1} J_2^\dagger v_2) = 0 \quad (43)$$

$$\dot{\tilde{\sigma}} \stackrel{(12)(41)(11)}{=} - \begin{pmatrix} T J_1 N_{ab} J_1^\dagger K_1 & T J_1 N_{ab1} J_2^\dagger K_2 \\ T J_2 N_{ab} J_1^\dagger K_1 & T J_2 N_{ab1} J_2^\dagger K_2 \end{pmatrix} \begin{pmatrix} \tilde{\sigma}_1 \\ \tilde{\sigma}_2 \end{pmatrix} =: -M_4 \tilde{\sigma} \quad (44)$$

$$\dot{z} = \begin{pmatrix} \dot{\theta} \\ \dot{\sigma}_a \\ \dot{\sigma}_b \\ \dot{\tilde{\sigma}} \end{pmatrix} =: \begin{pmatrix} f_{41}(z) \\ 0 \\ 0 \\ f_{44}(z) \end{pmatrix} = F_4(z). \quad (45)$$

IV. Stability Analysis

Let us begin the stability analysis by defining three assumptions: Since the quaternion coordinates have two equilibria, we restrict the initial error to the “positive equilibrium” in *Assumption 1*; *Assumption 2* formulates a criterion on task independence according to [10]; *Assumption 3* specifies the task gains according to [9] with some minor changes subject to the unit quaternion representation.

Assumption 1 (Initial States) *The initial states are defined to be in the set*

$$\Gamma_+ := \{\mathbb{R}^3 \times \{\tilde{\epsilon}_{RE} \in \mathbb{S}^3, \tilde{\eta}_{RE} \in (0, 1]\} \\ \times \mathbb{R}^3 \times \{\tilde{\epsilon}_{RS} \in \mathbb{S}^3, \tilde{\eta}_{RS} \in (0, 1]\}\}, \quad (46)$$

for the state vector (24).

Assumption 2 (Task Independence) *When an additional task is considered, the task Jacobian is independent with respect to the combination of the higher-priority Jacobians, i.e.,*

$$\rho(J_{1,\dots,(l-1)}^\dagger) + \rho(J_l^\dagger) = \rho([J_{1,\dots,(l-1)}^\dagger, J_l^\dagger]), \quad (47)$$

for $l \in \{2, \dots, (j+k)\}$ where $\rho(\dots)$ is the rank of the matrix, k is the number of equality tasks and j the total number of set-based tasks.

Assumption 3 (Control Gains) For the case of $k = 2$ the task gains are chosen as $K_1 = k_1 I_6 \in \mathbb{R}^{6 \times 6}$, $K_2 = k_2 I_6 \in \mathbb{R}^{6 \times 6}$ for the first, and second task, respectively, with

$$k_1 > 0, \quad k_2 > \frac{\bar{\lambda}_{21}^{-2}}{4\underline{\lambda}_{11}\underline{\lambda}_{22}} k_1, \quad (48)$$

where $\bar{\lambda}_{kl}$ and $\underline{\lambda}_{kl}$ denote the largest and smallest singular values for all modes i of the matrix $M_{i,kl}K_x^{-1}$, i.e., the submatrix $M_{i,kl}$ without the respective matrix gain contribution K_x^{-1} .

Theorem 1 (Asymptotic stability in the large) Let Assumptions 1, 2 and 3 hold. Then the origin of the task error dynamics (29) is asymptotically stable in the large on Γ_+ for the regulation case. The assumptions on the gains and the independence conditions are sufficient. In the special case in which only Assumption 2 is violated, the error dynamics (29) is (Lyapunov) stable but does not necessarily converge to zero.

proof 1 The proof is an extension of [9] and [10], including the attitude problem with unit quaternion representation. It uses the Lyapunov direct method. We define a positive definite Lyapunov function candidate $V : \Gamma_+ \rightarrow \mathbb{R}$ whose time derivative $\dot{V}$ is negative definite in Γ_+ along the system's trajectories under Assumptions 1, 2, and 3. Therefore, the error converges asymptotically to zero.

We begin in Part I by showing asymptotic stability for mode 1. Afterwards we generalize in Part II the result for all modes and consider the case when Assumption 2 does not hold.

Part I. We define the continuously differentiable Lyapunov function candidate $V_1 : \Gamma_+ \rightarrow \mathbb{R}$ as

$$V_1(\tilde{\sigma}) := \frac{1}{2} \tilde{\sigma}^T \tilde{\sigma}. \quad (49)$$

Thus, for the time derivative of the Lyapunov function candidate with $\dot{\theta} = g_1$ we get

$$\begin{aligned} \dot{V}_1 &= \tilde{\sigma}^T \dot{\tilde{\sigma}} \stackrel{(29)}{=} -\tilde{\sigma}^T \begin{pmatrix} TK_1 & TJ_1 N_1 J_2^\dagger K_2 \\ TJ_2 J_1^\dagger K_1 & TJ_2 N_1 J_2^\dagger K_2 \end{pmatrix} \tilde{\sigma} = -\tilde{\sigma}^T \underbrace{\begin{pmatrix} M_{1,11} & M_{1,12} \\ M_{1,21} & M_{1,22} \end{pmatrix}}_{=: M_1} \tilde{\sigma} \\ &= -\tilde{\sigma}_1^T M_{1,11} \tilde{\sigma}_1 - \tilde{\sigma}_2^T M_{1,22} \tilde{\sigma}_2 - \tilde{\sigma}_1^T M_{1,12} \tilde{\sigma}_2 - \tilde{\sigma}_2^T M_{1,21} \tilde{\sigma}_1. \end{aligned} \quad (50)$$

Below we discuss the negative definiteness of $\dot{V}_1$ by its subcomponents:

1. If Assumption 1 holds, the first component is negative definite since

$$\begin{aligned} -\tilde{\sigma}_1^T M_{1,11} \tilde{\sigma}_1 &= -\tilde{\sigma}_1^T (T(\tilde{q})K_1) \tilde{\sigma}_1 \\ &\stackrel{(13)}{=} -\tilde{\sigma}_1^T \begin{pmatrix} k_1 I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & \frac{k_1}{2} (S(\tilde{\epsilon}) + \tilde{\eta} I_3) \end{pmatrix} \tilde{\sigma}_1 < 0 \quad \forall \tilde{\eta} > 0. \end{aligned} \quad (51)$$

2. If the tasks are independent according to Assumption 2, then $J_2 N_1 J_2^\dagger = I_6$. Consequently, with $K_2 = k_2 I_6$ and the same properties of $T(\tilde{q})$ as in (51), the second component satisfies $-\tilde{\sigma}_2^T M_{1,22} \tilde{\sigma}_2 < 0$.
3. By definition of the null-space projector, $J_1 N_1 = J_1 (I - J_1^\dagger J_1) = 0$, which implies $M_{1,12} = TJ_1 N_1 J_2^\dagger K_2 = 0$ and therefore the third component $-\tilde{\sigma}_1^T M_{1,12} \tilde{\sigma}_2 = 0$.
4. As presented in [9], the control gains are chosen according to Assumption 3 such that the remaining cross term $-\tilde{\sigma}_2^T M_{1,21} \tilde{\sigma}_1$ can be handled and the sum $\dot{V}_1$ is negative definite.

Therefore, V_1 is positive definite and $\dot{V}_1$ is negative definite, and thus $\sigma = 0$ is an asymptotically stable equilibrium point for mode 1. Stability of the other modes follows by the same reasoning.

Part II. We choose a Lyapunov function candidate for all modes as

$$V(\sigma) := \frac{1}{2} \sigma^T \sigma, \quad (52)$$

which is positive definite on Γ_+ . For mode i , the corresponding time derivative satisfies

$$\dot{V}_i = -\sigma^T M_i \sigma, \quad (53)$$

where M_i is defined by the task error dynamics (29), (34), (39), and (44). Under Assumptions 1, 2 and 3, each M_i is positive definite, and thus $\dot{V}_i$ is negative definite for every mode i .

We also consider the convex combination

$$\begin{aligned} \dot{V} &= -\sigma^T \underbrace{(\lambda_1 M_1 + \lambda_2 M_2 + \dots + \lambda_{2j} M_{2j})}_{=: Q} \sigma \\ \text{for } \sum_{i=1}^{2j} \lambda_i &= 1, \lambda_i \geq 0. \end{aligned} \quad (54)$$

Since the cone of positive definite matrices is convex, the matrix Q is also positive definite. Consequently, $\dot{V}$ is negative definite and $\sigma = 0$ is an asymptotically stable equilibrium point for all modes.

However, in the case of activating a mode $i \neq 1$ such that the independence condition in Assumption 2 does not hold, some components become only negative semidefinite and include matrices of the form $J_i N_x J_i^T \geq 0$. Consequently, the corresponding contribution to $\dot{V}$ is negative-semidefinite and the total $\dot{V}$ is also negative-semidefinite (rather than negative definite). Accordingly, based on the positive definite Lyapunov function V and the negative-semidefinite $\dot{V}$, the system is Lyapunov stable, but it is not necessarily converging to the equilibrium point $\sigma = 0$.

Remark 1 If a mode $i \neq 1$ is active, $\dot{V}$ is only negative-semidefinite, since a direct trajectory that reaches the tracking goal $\sigma = 0$ would violate the respective set-based task. Either an indirect trajectory within the admissible set C leads to $\sigma = 0$, or the minimal tracking error $\sigma_{\min}$ attainable on C (i.e., subject to the set-based task) defines a new equilibrium point.

Remark 2 The task allocation is guided by a continuous trajectory, such that the current orientation remains in the vicinity of the desired one. Accordingly, the restriction on the quaternion error imposed by Assumption 1 is not restrictive in practice.

Remark 3 The stability proof can be extended to consider more than $k = 2$ equality tasks, and thus to follow more than two pose trajectories. This requires showing positive definiteness of the matrices $M_i \in \mathbb{R}^{6k \times 6k}$. For example, for an orbital robot with two manipulators we obtain $k = 3$ pose tracking tasks.

V. Simulation Results

A. Parametrical and Simulation Setup

We use a DLR lightweight robot [13] with 7 joints and a scaled size of 2 as an exemplary configuration of a space robot arm. The set-based tasks are specified by the minimum target distance $c_{a_1} = 3.13$ m, the maximum field vector error $c_{b_2} = 0.0875$ ($\approx 5^\circ$), the joint limits $c_{c_1} = -\frac{\pi}{8}$ rad, $c_{c_2} = \frac{\pi}{8}$ rad, and the workspace limits $c_{d_1} = 0.4$ m, $c_{d_2} = 1.5$ m, $c_{e_1} = 0$ m, $c_{e_2} = 1.6$ m. The control parameters are chosen according to Assumption 3 by $k_1 = 3$ and $k_2 = \frac{3}{5}$.

The control allocation is implemented on our real-time satellite simulator (SASI) within the attitude and orbital control unit, running at a cycle rate of 10 [Hz]. SASI interfaces the Dynamik Simulator (DYSI) and two hardware-in-the-loop (HiL) facilities: The European Proximity Operations Simulator (EPOS 2.0) (Figure 5), a facility for inspection and rendezvous simulation, and the On-Orbit Servicing Simulator (OOS-Sim) (Figure 4), a test facility for capturing/berthing simulation [11]. The following software-in-the-loop simulation results are based on the DYSI representing orbital and multi-body dynamics of the servicer and target, including actuator models, a sloshing model, and environment models for solar radiation pressure, atmospheric drag, and gravity gradient torque.

B. Simulation Results

Firstly, we verify our stability results from Theorem 1 within simulations which are shown in Figure 6. The analysis considers the following scenarios: (i) End-effector pose regulation while the servicer-base stays stationary and (ii)



Fig. 4 On-Orbit Servicing Simulator (OOS-Sim)



Fig. 5 European Proximity Operations Simulator (EPOS 2.0) [11]

servicer-base pose regulation commanded through the null space where the manipulator needs to perform a relative movement to keep and regulate its pose in the reference. Both show asymptotic convergence of the equality task errors expressed by its Lyapunov candidate function.

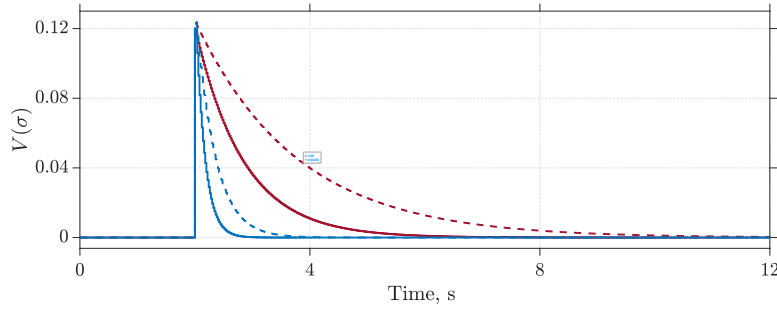


Fig. 6 Asymptotic convergence of the task error $\tilde{\sigma}$ based on Lyapunov function $V(\tilde{\sigma})$ for the (i) end-effector (blue) and the (ii) servicer-base (red). The solid line illustrate the case after a position step $-[0.1, 0.1, 0.1]^T$ and the dashed line shows the case after an orientation step about $\frac{\pi}{8}$ around all axis.

For analyzing purposes we consider a reference scenario as shown in Figure 3 with the following sequence of (discrete event triggered) guided phases: (I) Final Approach: Servicer approaches a target from the Far Approach Alignment Point (FAA) to the Mating Point; (II) Reach Phase: Servicer moves the end-effector to a time-varying grasping pose on a target, which spins around the approach axis; (III) Escape Phase: Servicer escapes from the target to a safe pose; (IV) Retreat Phase: Servicer moves back to the FAA ready for a retreat. Figure 7 shows the tracking of the end-effector pose σ_1 to a reference trajectory by smooth allocation of the actuator states for the prior described reference scenario.

For the motion phase (II) (Figure 8) and (III) (Figure 9), we compare the allocation based on the full pose tracking σ_1 , σ_2 versus the end-effector tracking σ_1 . The setting with full pose tracking is chosen such that servicer-base synchronizes with the target, resulting in a manipulator movement, where the target is seen as stationary. To provide a concrete example of the optimality metric, we parameterize the weighting matrix W as a diagonal matrix that minimizes the quadratic velocities θ for each degree of freedom via the weighing element w_i . Accordingly, based on the relative weights of the matrix elements w_1 to w_6 and w_7 to w_{13} , the end-effector maneuver is carried out primarily by the base or by the manipulator. If the system moves out of the valid set C , as shown in Figure 8 and 9, the respective switching between modes ensures system and safety constraints, such that all set-based tasks are satisfied during our reference scenario.

By examining the reach phase depicted in Figure 8, it has been noticed that we can minimize the movement of the base during the reach phase to the time-varying grasping pose of a spinning target by utilizing an appropriate weighting of W . This results in a reduction in fuel consumption. Figure 9 demonstrates how the full use of system dexterity

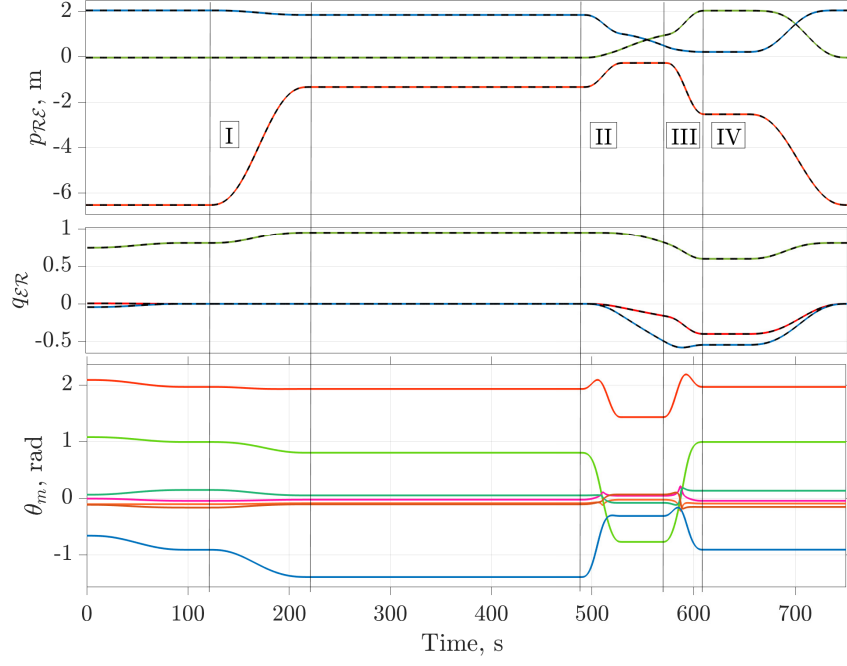


Fig. 7 Top: End-effector position tracking p_{RE} , decomposed in p_x (red), p_y (green), p_z (blue), its guided reference (black, dashed). Middle: End-effector orientation tracking, decomposed in ϵ_1 (red), ϵ_2 (green), ϵ_3 (blue), its guided reference (black, dashed). Bottom: Allocated manipulator joint angles θ_m .

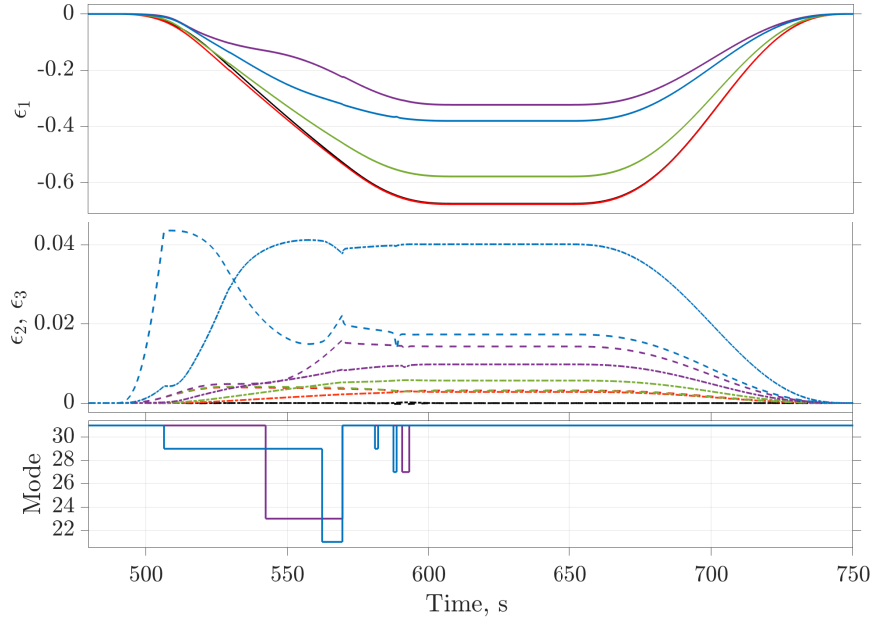


Fig. 8 Reach Phase: Allocated trajectories q_{SR} with ϵ_1 (solid), ϵ_2 (dashed), ϵ_3 (dot dashed) based on $W = \text{diag}\{1000, 1000, 1000, w_3, 1000, 1000, 1, \dots, 1\}$ with (i) $w_4 = 1$ (red), (ii) $w_4 = 10$ (green), (iii) $w_4 = 50$ (purple) and (iv) $w_4 = w_5 = w_6 = 20$ (blue) and (v) tracking of guided base σ_2 (black) and its allocated modes with prior specified colors.

by the allocation can improve the performance in mobility, e.g., a pitch movement of the base leads to an additional end-effector escape motion of the end-effector. The resulting increased mobility is essential for executing successful

escape maneuvers in high-risk situations or unexpected events, allowing the robot to avoid collisions and navigate challenging environments more effectively. Figures 8 and 9 also show a discontinuous behavior caused by switching between modes, which can reduce the performance and fuel efficiency. This drawback can be addressed by employing Model Predictive Control (MPC), as demonstrated in [12], although MPC is significantly more computationally demanding. Consequently, in addition to mission- and phase-specific performance goals, the optimization of W shall be aimed to reduce mode switching under nominal conditions. The stability property and the satisfaction of the set-based tasks holds at all times, including during mode switching, as shown by the stability proof and the simulation results.

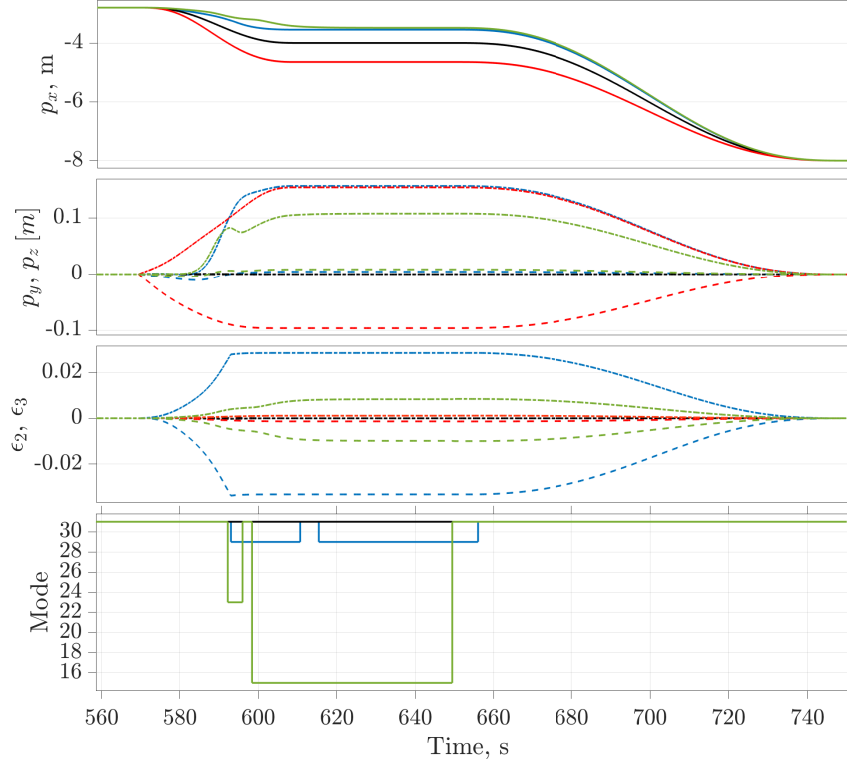


Fig. 9 Escape Phase: Allocated trajectories q_{SR} with ϵ_1 (solid), ϵ_2 (dashed), ϵ_3 (dot dashed) and p_{RS} with p_x (solid), p_y (dashed), p_z (dot dashed) based on $W = \text{diag}\{w_1, w_2, w_3, 1, w_4, w_5, 1, \dots, 1\}$ with (i) $w_1 = w_2 = w_3 = 1$, $w_4 = w_5 = 1000$ (red), (ii) $w_1 = w_2 = w_3 = 20$, $w_4 = w_5 = 1000$ (green), (iii) $w_1 = w_2 = w_3 = w_4 = w_5 = 10$ (blue), (iv) guided base (black) and its allocated modes with prior specified colors.

Table 4 Allocated Modes for Set-based Tasks

Mode i	Set-based tasks	Mode description
15	σ_e	WLI active
21	σ_b and σ_d active	FoV and WLI active
23	σ_d active	WLI active
27	σ_e active	JL active
28	σ_a and σ_b active	CA and FoV active
29	σ_b active	FoV active
30	σ_a active	CA active
31	no activation	no set-based tasks violated

VI. Conclusion

A method for the control allocation of free-flying space manipulator systems has been presented. Based on the proposed approach, we enable the use of an unlimited workspace, while explicitly including system and safety constraints such as collision avoidance, field of view, and manipulator workspace and joint limits. In a reference scenario simulation, we have shown (i) stability of tracking tasks, (ii) reliable satisfaction of the system and safety constraints, (iii) variability of the allocation based on a weighting matrix to optimize for the mission-specific performance goals, such as fuel consumption and mobility. Future experimental work will be done to verify our results from SiL simulations with our EPOS 2.0 HiL test facility including camera-based pose estimation data. Furthermore, we will investigate coordinated control based on model predictive control schemes that explicitly incorporate system dynamics, with a focus on real-time capabilities and robustness.

References

- [1] Z. Vafa and S. Dubowsky, "The kinematics and dynamics of space manipulators: The virtual manipulator approach," in *The International Journal of Robotics Research*, vol.9, pp. 3-21, 1990.
- [2] Y. Masutani, F. Miyazaki, and S. Arimoto, "Sensory feedback control for space manipulators," in *Proceedings of the 1989 International Conference on Robotics and Automation*, Scottsdale, AZ, USA, vol.3, pp. 1346-1351, May 1989.
- [3] Y. Xu, H.-Y. Shum, J.-J. Lee, and T. Kanade, "Adaptive control of space robot system with an attitude controlled base," in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, vol. 3, pp. 2005-2010, May 1992.
- [4] F. Aghili, "Coordination control of a free-flying manipulator and its base attitude to capture and detumble a noncooperative satellite," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, pp. 2365-2372, Oct. 2009.
- [5] E. Papadopoulos and S. Dubowsky, "Coordinated manipulator/spacecraft motion control for space robotic systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Sacramento, CA, USA, vol. 2, pp. 1696-1701, Apr. 1991.
- [6] A. M. Giordano, C. Ott, and A. Albu-Schäffer, "Coordinated Control of Spacecraft's Attitude and End-Effector for Space Robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2108-2115, Apr. 2019.
- [7] M. De Stefano, L. Sciavicco, and G. Antonelli, "Multi-rate tracking control for a space robot on a controlled satellite: A passivity-based strategy," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1319-1326, Apr. 2019.
- [8] R. Lampariello and G. Hirzinger, "Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, pp. 5652-5659, 2013.
- [9] G. Antonelli, "Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems," in *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985-994, Oct. 2009.
- [10] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen and J. Schrimpf, "Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: General formulation, stability analysis, and experimental results," in *Frontiers in Robotics and AI*, vol. 3, pp. 5652-5659, 2016.
- [11] H. Benninghoff, S. Boge, J. Furmans, D. Kirschner, J. B. Kuehn, A. Kuerten, and T. Zwick, "End-to-end simulation and verification of GNC and robotic systems considering both space segment and ground segment," *CEAS Space Journal*, vol. 10, no. 2, pp. 535-553, 2018.
- [12] Kötting, Peter. "Model Predictive Control based Coordinated Control for Free-Flying Space Manipulator Systems." IFAC-PapersOnLine 58.18 (2024): 133-138.
- [13] G. Hirzinger *et al.*, "DLR's torque-controlled light weight robot III—are we reaching the technological limits now?," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, USA, vol. 2, pp. 1710-1716, May 2002.