ESA GNC & ICATT Conference 2026
28 Sep. – 2 Oct. 2026, Sevilla, Spain
https://atpi.eventsair.com/gncicatt/

Towards Safe On-Board Autonomy in Robotic On-Orbit Servicing

Peter Kötting[1,*], Alexander Meinert[2], Peter Stadler[2], Niklas Baldauf[2], Fabio Schneider[1], and Roberto Lampariello[1]

[1] *German Aerospace Center (DLR), Institute for Mechatronics and Robotics, Münchener Straße 20, 82234 Weßling-Oberpfaffenhofen, Germany*
[2] *e:fs TechHub GmbH, Space Applications Group, Dr.-Ludwig-Kraus-Straße 6, 85080 Gaimersheim, Germany*
[*] *Corresponding author: email@address.com*

ABSTRACT

Safe on-board motion control for robotic On-Orbit Servicing is challenging due to the coupled multibody dynamics of the servicer and its operational safety constraints. This paper presents and evaluates control approaches based on Model Predictive Control and Constrained Imitation Learning for on-board motion control. MPC formulations are developed for different control architectures of increasing complexity, ranging from relative-position and spacecraft-attitude control to a combined control scheme. We address stability, recursive feasibility, actuator limits, joint constraints, collision avoidance, and field-of-view requirements. The constrained imitation-learning approach approximates the optimal control policy with a neural network and incorporates control-barrier- and control-Lyapunov-function-based safety and stability certificates during training and deployment to overcome the computational burden of solving the MPC online. As a benchmark case, we consider a detumbling scenario of ESA's Envisat satellite. The control frameworks are evaluated in a high-fidelity Modelica-based simulation environment following a representative concept of operations. Performance metrics include propellant consumption, computational runtime, and constraint violations.

1 Introduction

On-Orbit Servicing (OOS) missions are rising in importance as Earth orbits become more crowded and assets need to remain flexible and sustainable. Robotic OOS enables key future applications such as active debris removal (ADR), in-space servicing, assembly, and manufacturing (ISAM), and satellite lifetime extension through repair and refuelling. A spacecraft equipped with a robotic manipulator is a complex, over-actuated system with strongly coupled nonlinear dynamics. Classical control approaches are often limited in their ability to account for these nonlinearities, systematically exploit over-actuation, or handling system and safety constraints. The project *KI-Safety in MPC für InOrbit-Servicing* (KISMOS) aims at developing control strategies that address these challenges across all mission phases, including rendezvous, approach, tracking, capture, detumbling, and abort. Notably, the concept of operations builds upon the recently published ESA Guidelines on Safe Close Proximity Operations.

As a benchmark case in the publication we choose a detumbling scenario, with ESA's ENVISAT satellite. Within e.Deorbit [1] [2], ENVISAT, with a mass of approximately 8000 kg and a length of about 27 m, was selected as the target satellite. Due to the target satellite's large mass and dimensions, the servicer spacecraft must perform synchronized flight relative to a predefined grasping point on the target while the robotic manipulator captures it. Due to the target satellite's large appendages and tumbling motion, the approach maneuver is constrained by a tight approach cone. Similar scenarios can be envisioned within future Active Debris Removal activities, given

© German Aerospace Center (DLR), e:fs TechHub GmbH

that large satellites pose the highest threat, but also in On-Orbit Servicing missions, in which target satellites may be tumbling (see ESA ClearSpace-1 [3], ESA CAT-IOD and CNES DIANE [4]).

During close proximity operations of the servicer, motion control algorithms need to control the position and attitude of the servicer-base as well as the robotic manipulator, while satisfying operational constraints, such as keep-out-zone, approach corridor, field-of-view, and input saturation constraints. The safety constraints are in particular enforced to investigate the case of autonomous operations without the need of on-ground trajectory planning for close-proximity operations. From a system perspective, the constrained control problem of the servicer robot can be structured in different ways: (i) separated position, attitude, and robotic joint control; (ii) spacecraft pose (position–attitude) and robotic joint control; or (iii) a combined control scheme that jointly controls position, attitude, and manipulator joints.

Regarding the decoupled subsystem-based controller design (i), an extensive body of literature exists for MPC in spacecraft control applications. In [6], the authors propose a robust tube-based MPC for relative position control and autonomous spacecraft rendezvous with a target, while [26] provides feasibility and stability guarantees for an MPC policy in a rendezvous scenario with the Envisat spacecraft. For the attitude control system and pointing tasks, the works in [24, 29, 25] derive a dual-loop attitude MPC suitable for onboard implementation on a commercial off-the-shelf processor. For the robotic subsystem, the results in [27, 28] demonstrate safe control strategies for a space manipulator using Control Barrier Functions (CBF). The combined control has been addressed by an output tracking MPC [7], which tracks a desired operational space end-effector pose with an artificial reference.

However, a comparison in a full orbital servicing scenario has not been done to evaluate the capabilities and differences of the system architectures (i)-(iii). Hence, as a first contribution, we develop MPC for all system architectures, i.e., with increasing complexity levels. We propose concrete design procedures to incorporate formal guarantees such as stability and recursive feasibility for the respective operational constraints and system dynamics.

As a second contribution, we use imitation learning to approximate MPC with a neural control policy, thereby mitigating the computational burden of MPC. To this end, we derive safety and stability certificates based on Control Barrier Functions and Control Lyapunov Functions (CLF) and embed them into a differentiable safety layer of the neural network (NN). This quadratic program (QP) output layer is intended to steer the training process towards inherently learning the safe operational zones and stability requirements of OOS, while guaranteeing constraint satisfaction during deployment. The approach is motivated by the work in [21], where an MPC for spacecraft close proximity operations is safely imitated by a neural network using a physics-informed loss function and a subsequent safety filter QP. Building on these results, we propose to embed the QP into the NN architecture, while accounting for related work. In [22], the authors show how incorporating a differentiable CBF layer yields less computational effort and increased closed-loop performance compared to a nominal MPC or a post-safety filter QP. The results in [23] reveal that a neural policy with an integrated safety layer achieves less conservative trajectories that are commonly obtained when applying a safety filter after a learning-based controller.

The control approaches are assessed in a high-fidelity Modelica-based simulation environment along a representative concept of operations. The comparison considers propellant consumption, computational runtime, and constraint satisfaction.

2 Problem Statement

This work addresses the full-body constrained control of servicer spacecraft with a robotic manipulator for OOS operations. First, the concept of operations is introduced to define the mission context in which the proposed control approaches are evaluated. Subsequently, the system dynamics of the underlying control problems are presented.

2.1 Concept of Operations

The overall mission objective is to detumble the satellite Envisat. The mission is divided into a sequence of operational phases, summarized in Table 1 and illustrated in Figure 1. Each phase is associated with a reference frame defining the desired relative position, attitude and joint configuration, as well as conditions governing the transition to the subsequent phase.

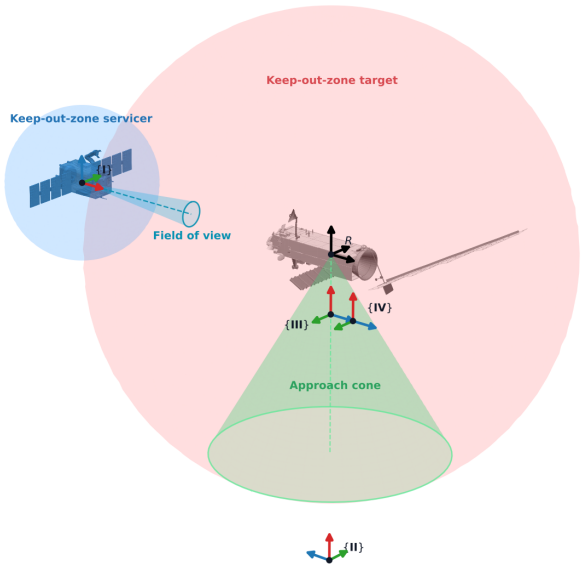


Figure 1 *Operational reference frames and geometric constraints. Center: the tumbling Envisat target, showing the body-fixed approach corridor in green and the target keep-out zone in red. Left: the servicer spacecraft equipped with a robotic manipulator, showing the servicer keep-out zone in blue and the relative-navigation sensor field-of-view cone.*

The operational constraints considered throughout the mission setup are defined as follows: (i) the Keep-Out Zone (KOZ) constraint, which defines a protected volume around the target and the servicer that accounts for the maximum geometric envelopes to ensure collision avoidance for all admissible relative attitude configurations; (ii) the Approach Corridor (AC) constraint, which defines the admissible region for controlled entry into the KOZ during close-proximity operations and ensures sufficient distance from the target; and (iii) the Field-of-View (FoV) constraint, which ensures that the target remains within the field of view of the relative navigation sensors, such as cameras or LiDAR, to maintain relative-pose estimation.

Mission Phase μ	**Motion**	**Constraints**	**Transition**
Rendezvous phase	Frame {**I**} – {**II**}	KOZ, FoV	Frame {**II**}, AC, RTA
Approach phase	Frame {**II**} – {**III**}	FoV, AC	Frame {**III**}
Tracking phase	Frame {**III**} – {**IV**}	FoV, AC	Frame {**IV**}
Capture phase	Frame {**IV**} – {**V**}	FoV, AC	Frame {**IV**}, TCC
Detumble phase	Frame {**V**} – {**VI**}	FoV, AC	Frame {**VI**}
(Abort phase)*	Frame {...} – {**II**}	FoV, ABC	Frame {**II**}

Table 1 *Mission Phase Overview*

The operations start from an initial V-bar hold point at a safe distance from the target, marking the beginning of the rendezvous phase. The servicer then approaches the relative pose associated with frame {**II**} while satisfying the FoV and KOZ constraints. Once {**II**} is reached, the approach phase is initiated, provided that the servicer is within the AC and the Relative Target Attitude (RTA) satisfies specified bounds. During the approach phase, the servicer tracks the target-fixed, tumbling frame {**III**} and synchronizes its motion. Once synchronization is achieved at {**III**}, the tracking phase is initiated. During this phase, the servicer approaches the relative pose defined by the target-fixed frame {**IV**}, such that the target's capture interface is located within the workspace of the robotic manipulator. In the subsequent capture phase, the robotic manipulator is commanded toward the capture configuration while the servicer maintains synchronized motion with the tumbling target according to frame {**IV**}. The detumbling phase is initiated once the Target Capture Complete (TCC) condition is satisfied. The coupled servicer–target stack is then detumbled by reducing its angular rates toward zero. The reference

frame $\{\mathbf{V}\}$ is computed along a damped prediction of the target motion. In the event of an operational-constraint violation within the KOZ or a system-level fault detected by the FDIR system, an abort phase is initiated to transfer the servicer through the Abort Corridor (ABC) to a safe hold point outside the critical proximity region, such as the relative pose associated with frame **II**. This phase is omitted here for brevity, as from a control perspective it can be treated as a reversed approach maneuver with the ABC defined by the AC.

2.2 Modeling and Control Setup

From a system perspective, a control scheme can be structured in several ways, as shown in Figure 2: (i) separate relative-position, attitude, and robotic-arm control; (ii) spatial spacecraft (position–attitude) control and robotic-arm control; or (iii) a combined control scheme that jointly controls the position, attitude, and manipulator joints. In the following, we introduce the underlying system dynamics for the prediction models corresponding to cases (i)–(iii).

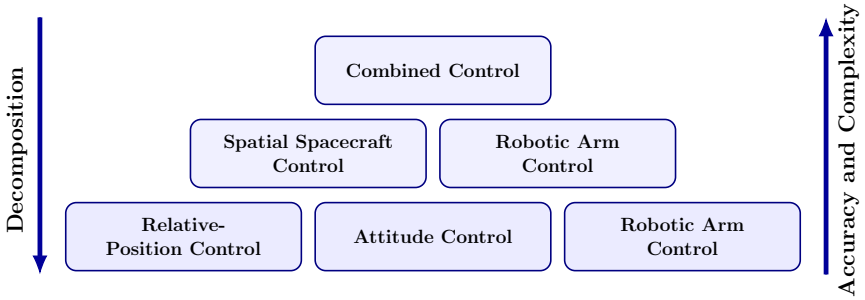


Figure 2 *Complexity levels of the free-flying orbital robot control problem*

Relative-Position Clohessy–Wiltshire Dynamics

The relative translational motion of the servicer spacecraft with respect to the target spacecraft can be described in the local-vertical/local-horizontal (LVLH) frame. Assuming that the target follows a circular reference orbit and that the relative distance is small compared with the orbital radius, the linearized Clohessy–Wiltshire/Hill equations are given by

$$\dot{x}_{cw} = A_{\mathrm{cw}} x_{cw} + B_{\mathrm{cw}} u_{cw}, \tag{1}$$

with the state vector $x_{cw} = [p_{cw}^{\top},\ \dot{p}_{cw}^{\top}]^{\top} \in \mathbb{R}^6$, where p_{cw} and $\dot{p}_{cw}$ denote the relative position and velocity in the LVLH frame, respectively, and $u_{cw} \in \mathbb{R}^3$ represents the applied force in the LVLH frame. The system matrices are defined in [5].

Attitude Dynamics

For the attitude dynamics, we employ a unit-quaternion attitude representation since it provides a singularity-free, four-parameter representation of orientation. Accordingly, the attitude quaternion is defined as

$$q = [\eta, \epsilon^{\top}]^{\top} \in \mathbb{S}^3 = \{q \in \mathbb{R}^4 : q^{\top} q = 1\} \subset \mathbb{R}^4, \tag{2}$$

where $\eta \in \mathbb{R}$ denotes the scalar part and $\epsilon \in \mathbb{R}^3$ denotes the vector part. The rotation matrix $R(q) \in \mathrm{SO}(3)$ associated with the attitude quaternion is given by

$$R(q) = (\eta^2 - \|\epsilon\|^2)\mathbf{I}_3 + 2\epsilon\epsilon^{\top} + 2\eta[\epsilon]_{\times}. \tag{3}$$

with the cross product matrix $[\epsilon]_{\times}$. For the body-fixed angular velocity $\omega \in \mathbb{R}^3$, the quaternion kinematics are expressed as

$$\dot{q} = \frac{1}{2}\,\Xi(q)\,\omega, \qquad \Xi(q) = \begin{pmatrix} -\epsilon^{\top} \\ \eta\mathbf{I}_3 + [\epsilon]_{\times} \end{pmatrix}. \tag{4}$$

The motion of a rigid body is described by the Euler dynamics

$$H_r\dot{\omega} + [\omega]_\times H_r\omega = \tau_b. \tag{5}$$

By combining the Euler dynamics (5) and the quaternion kinematics (4), we obtain the attitude prediction model

$$\dot{x}_a = f_a^c(x_a, u_a) = \begin{pmatrix} \frac{1}{2}\,\Xi(q)\,\omega \\ H_r^{-1}(u_a - [\omega]_\times H_r\omega) \end{pmatrix}, \tag{6}$$

with the state vector $x_a = [q^\top,\, \omega^\top]^\top \in \mathbb{R}^7$ and the input vector $u_a = \tau_b \in \mathbb{R}^3$ representing the body-frame torque.

Spatial Spacecraft Dynamics

To derive the spatial spacecraft dynamics, we omit the CW/Hill orbital acceleration terms and combine the resulting translational motion model with the rigid-body attitude dynamics. We use, the spatial velocity $\nu = [v^\top,\, \omega^\top]^\top \in \mathbb{R}^6$ expressed in the spacecraft body frame, whereas the position p and attitude q are represented with respect to an orbital reference frame. Accordingly, we combine the relative translational dynamics without the orbital terms from (1) with the attitude dynamics (6) to derive the spatial spacecraft model

$$\dot{x}_b = f_b^c(x_b, u_b) = \begin{pmatrix} R(q)\,v \\ \frac{1}{2}\,\Xi(q)\,\omega \\ H_b^{-1}(u_b - C_b(\nu)\,\nu) \end{pmatrix}, \tag{7}$$

with the state vector $x_b = [p^\top,\, q^\top,\, \nu^\top]^\top \in \mathbb{R}^{13}$ and the body-frame force and torque $u_b = [f_b^\top,\, \tau_b^\top]^\top \in \mathbb{R}^6$. The inertia and Coriolis–centrifugal matrices

$$H_b = \begin{bmatrix} H_t & H_{tr} \\ H_{tr}^\top & H_r \end{bmatrix} \in 6\times 6, \qquad C_b = \begin{bmatrix} C_t & C_{tr} \\ C_{rt} & C_r \end{bmatrix} \in 6\times 6, \tag{8}$$

include the translational–rotational coupling blocks C_{tr}, C_{rt} and H_{tr}, which arise when the origin of the body-fixed frame does not coincide with the spacecraft center-of-mass. The body-frame origin is selected based on an actuator-related reference, whereas the center-of-mass location varies due to manipulator reconfiguration, propellant consumption, and fuel sloshing.

Orbital Robot Dynamics

In the following, we extend the spacecraft model by incorporating a robotic manipulator with n_m joints. This introduces the joint angles $\theta \in \mathbb{R}^{n_m}$ and joint velocities $\dot{\theta} \in \mathbb{R}^{n_m}$ as additional states, together with the joint torques $\tau \in \mathbb{R}^{n_m}$ as additional control inputs. We define the generalized velocity as $\phi = [\nu^\top,\, \dot{\theta}^\top]^\top \in \mathbb{R}^{6+n_m}$. The coupled orbital-robot prediction model is then given by

$$\dot{x}_c = f_c^c(x_c, u_c) = \begin{pmatrix} R(q)\,v \\ \frac{1}{2}\,\Xi(q)\,\omega \\ \dot{\theta} \\ H_c(\theta)^{-1}\,(u_c - C_c(\theta,\phi)\,\phi) \end{pmatrix}, \tag{9}$$

where the combined spacecraft-manipulator state and input vectors are defined by

$$x_c = \begin{bmatrix} p^\top & q^\top & \theta^\top & \phi^\top \end{bmatrix}^\top \in \mathbb{R}^{13+2n_m}, \qquad u_c = \begin{bmatrix} f_b^\top & \tau_b^\top & \tau^\top \end{bmatrix}^\top \in \mathbb{R}^{6+n_m}. \tag{10}$$

© *German Aerospace Center (DLR), e:fs TechHub GmbH*

The generalized inertia and Coriolis–centrifugal matrices are partitioned as

$$H_c(\theta) = \begin{bmatrix} H_b(\theta) & H_{bm}(\theta) \\ H_{bm}^\top(\theta) & H_m(\theta) \end{bmatrix} \in \mathbb{R}^{(6+n_m)\times(6+n_m)}, \qquad C_c(\theta,\phi) = \begin{bmatrix} C_b(\theta,\phi) & C_{bm}(\theta,\phi) \\ C_{mb}(\theta,\phi) & C_m(\theta,\phi) \end{bmatrix} \in \mathbb{R}^{(6+n_m)\times(6+n_m)}. \quad (11)$$

Here, H_{bm}, C_{bm}, and C_{mb} describe the dynamic coupling between the spacecraft base and the robotic manipulator. In particular, manipulator motion induces reaction forces and torques on the spacecraft, while spacecraft motion affects the manipulator dynamics. The previously introduced decoupled prediction models neglect the dynamic coupling between the spacecraft base and the manipulator. The considered prediction models therefore form a hierarchy of increasing dynamic coupling and model complexity, ranging from separate translational and rotational spacecraft models to a coupled rigid-multibody formulation of the full orbital robot. Additional system and environmental effects are represented in the high-fidelity simulation environment described in Section 4.1. However, they are omitted from the prediction models to limit the online computational demand.

3 Method

In the following, we present the proposed approaches for addressing the constrained motion problems introduced in Figure 2. We first formulate the MPC approaches in Section 3.1 and then turn to constrained imitation learning in Section 3.2.

3.1 Model Predictive Control

Nonlinear Model Predictive Control

We first present the MPC problem to be solved for the spatial spacecraft and combined control problem. The discrete-time optimal control problem (OCP) solved at each sampling instant t is formulated as

$$V_N(x_t, r_t) = \min_{u_{\cdot|t},\, x_{\cdot|t}} J_N(x_{\cdot|t}, r_{\cdot|t}, u_{\cdot|t}) \quad (12a)$$

$$s.t. \quad x_{0|t} = x_t, \quad (12b)$$

$$x_{k+1|t} = f^d(x_{k|t}, u_{k|t}), \quad (12c)$$

$$(x_{k|t}, u_{k|t}) \in \mathcal{Z}, \quad (12d)$$

$$x_{N|t} \in \mathcal{X}_f(r_{N|t}), \quad (12e)$$

$$k = 0, \dots, N-1,$$

where t denotes the time step, k the prediction step, $N \in \mathbb{N}$ the prediction horizon. The constraints are composed by the discrete time prediction model (12c), the path constraints (12d), and the terminal region (12e). The solution of (12) consists of the optimal state trajectory $x^*_{\cdot|t}$, input trajectory $u^*_{\cdot|t}$, and the optimal value function $V^*_N(x_t, r_t)$. The resulting closed-loop system is given by

$$x_{t+1} = f_w(x_t, u_t), \quad u_t = u^*_0, \quad (13)$$

with the real-world system dynamics f_w, which will be approximated as good as possible by the high-fidelity simulation model in Section 4.1. For a prediction horizon $N \in \mathbb{N}$, the finite-horizon cost is

$$J_N(x, r, u) := \sum_{k=0}^{N-1} \ell(x_{k|t}, r_{k|t}, u_{k|t}) + V_f(x_{N|t}, r_{N|t}), \quad (14)$$

where $\ell(\cdot)$ denotes the stage cost and $V_f(\cdot)$ the terminal cost. We use the quadratic stage cost

$$\ell(x, r, u) := e_x^\top(r, u) Q\, e_x(r, u) x + u^\top R u, \quad (15)$$

where $e_x(r, u)$ is the state error and $Q \succ 0$ and $R \succ 0$ are (symmetric positive definite) design matrices chosen to track the operation-specific references.

Depending on the considered problem class, as illustrated in Figure 2, the prediction model can be based on the discretized version of the relative-position dynamics (1), attitude dynamics (6), spatial dynamics (7), or combined orbital robot dynamics (9). Depending on the selected control architecture, the path constraints (12d) incorporate the following system and safety requirements: (i) polytopic actuator constraints accounting for thruster and joint motor saturation; (ii) conic constraints defining the FoV, AC and ABC; and (iii) KOZ constraints by spherical object sets.

Terminal Ingredients

In the following, we construct a terminal region for the considered problem classes based on the theory in [8], thereby guaranteeing recursive feasibility and nominal stability. The terminal ingredients are defined by the terminal set and terminal cost

$$\mathcal{X}_f(r_{N|t}) := \|e_x(r_{N|t}, x_{N|t})\|_{P_f} \leq \alpha, \qquad V_f(r_{N|t}, x_{N|t}) = e_x(r_{N|t}, x_{N|t})^\top P_f e_x(r_{N|t}, x_{N|t}), \tag{16}$$

respectively, with a positive definite matrix $P_f \succ 0$ and some constant α. In the following we show how to compute the ellipsoidal terminal region $P_f \succ 0$, where the respective constant α can be computed according to [8]. Since the position, orientation and joint angles result from the system dynamics only through kinematics, we construct the terminal ingredients for the velocity states only. Stabilization to a constant velocity reference ensures bounded and predictable system evolution; in particular, stabilization to zero velocity yields convergence to a constant configuration, although not necessarily to the prescribed reference configuration. Restricting the terminal ingredients to the velocity states simplifies the terminal-region computation and leaves the position, orientation and joint angles unconstrained by the terminal set, which enables tracking of distant references as needed to track the operational poses directly. Accordingly, we construct the terminal ingredients based on the dynamic subsystem. Consider the corresponding system Jacobians

$$A_{\text{dyn}}(x) = \left[\frac{\partial f_{\text{dyn}}(x)}{\partial x}\right], \qquad B_{\text{dyn}}(x) = \left[\frac{\partial f_{\text{dyn}}(x)}{\partial u}\right], \tag{17}$$

where f_{dyn} denotes the dynamic subsystem. Based on the jacobian linearization (17) the semi-definite program

$$\begin{aligned} &\min_{Y_f, X_f} \; -\log\det X_f \\ &s.t. \quad \begin{bmatrix} X_f & X_f A_{\text{dyn}}^\top(r) + Y_f^\top B_{\text{dyn}}^\top(r) & (Q+\epsilon_Q I)^{\frac{1}{2}} X_f & R^{\frac{1}{2}} Y_f \\ * & X_f & \mathbf{0} & \mathbf{0} \\ * & * & \mathbf{I} & \mathbf{0} \\ * & * & * & \mathbf{I} \end{bmatrix} \succeq 0, \quad r \in \mathcal{Z}, \end{aligned} \tag{18}$$

with some positive constant ϵ_Q, computes a ellipsoidal terminal region by the matrix $P_f = X_f^{-1}$ and the corresponding stabilizing feedback gain $K_f = Y_f X_f^{-1}$ to guarantee decrease of the terminal cost for every reference state x in the admissible set $\mathcal{Z}_x$. The objective $(-\log\det X_f)$ maximizes the volume of the ellipsoid. The Linear Matrix Inequality (LMI) (18) must be verified for all admissible references $r \in \mathcal{Z}$. This verification can be performed either by deriving a convex representation of the problem or by gridding the admissible reference set. For the attitude-control problem, with $\omega_r \in \mathcal{Z}_\omega$, and the spatial-control problem, with $\nu_r \in \mathcal{Z}_\nu$, a convex formulation can be obtained since the corresponding system Jacobians admit a velocity-affine representation. For the combined control problem, however, no analogous affine representation in the generalized velocities $\phi_r \in \mathcal{Z}_\phi$ is available. Consequently, the LMI must be verified for the worst-case manipulator configurations by gridding the admissible joint-state set $\theta_r \in \mathcal{Z}_\theta$ and generalized velocities set $\phi_r \in \mathcal{Z}_\phi$.

Implementations

The MPC algorithms corresponding to the two highest complexity levels in Figure 2 share the same implementation details. The MPC for the 6-DoF spatial spacecraft dynamics (7) is referred to as the Model Predictive Control for

the Spacecraft (MPCS), whereas the MPC for the full $(6 + n_m)$-DoF orbital-robot dynamics (9) is referred to as the Model Predictive Combined Control (MPCC). Both controllers employ a velocity-level terminal region computed according to the semidefinite program (18), using YALMIP [9] as the LMI modeling layer and MOSEK [10] as the semidefinite-programming solver.

Special attention is given to the real-time capability of the on-board algorithms by exploiting algorithmic differentiation, a time-coarsening strategy, and the block-sparse structure of the resulting optimization problem. We formulate the optimal control problem (12) using CasADi's symbolic framework, which provides efficient algorithmic differentiation. The OCP (12) is transcribed as a multiple-shooting problem using a time-coarsening strategy [11], in which the discretization grid becomes exponentially coarser toward the end of the prediction horizon. This strategy exploits the exponential decay of sensitivities, i.e., the influence of a parametric perturbation at a future prediction step on earlier stages decreases exponentially with their temporal separation. The prediction horizon N and the coarsening exponent are optimized offline using Bayesian optimization to minimize the open-loop–closed-loop performance gap and the computation time. The resulting discretized NLP is solved using Fatrop [12]. Fatrop is a robust and efficient interior-point solver that exploits the block-sparse structure induced by the stage-wise optimal control formulation. The RMPCS integrates the full nonlinear 6-DoF spatial dynamics (7). In the MPCC, the computational burden of the prediction model (9) is reduced by integrating the nonlinear kinematics while computing the system matrices (11) at sampling time (t) and keeping them fixed over the prediction horizon.

Figure 3 illustrates the control architecture. The OCP is solved at a sampling rate of 10 [Hz], yielding the optimal spacecraft thrust force f_b^* and torque τ_b^*. The optimal joint trajectory, defined by the joint angles θ^*, joint velocities $\dot{\theta}^*$, and joint torques τ^*, is interpolated at 1000 [Hz]. The resulting desired joint torque τ_d is applied to the joint motors and augmented by a PD control law that tracks the optimal joint trajectory provided by the MPC while compensating for high-frequency disturbances and noise. In contrast to [7], this approach includes a terminal region to certify recursive feasibility and stability, incorporates the full robotic joint dynamics, and employs a joint-space formulation instead of the operational-space formulation in [7].

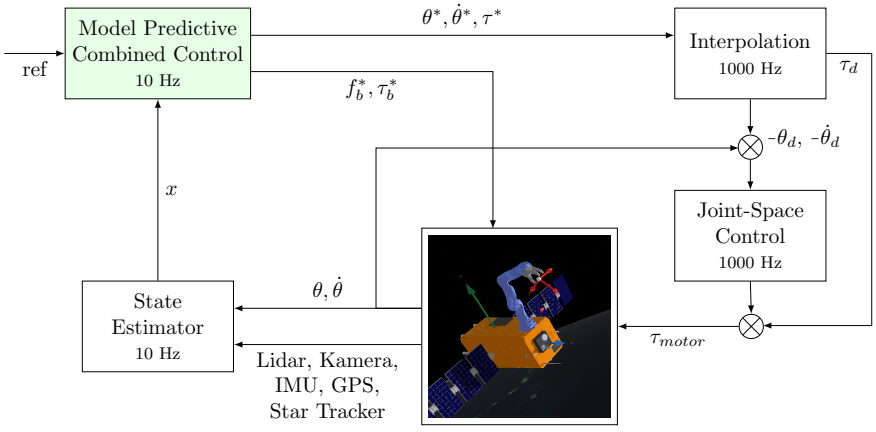


Figure 3 *Cascaded Model Predictive Combined Control*

3.2 Constrained Imitation Learning

While the MPC policies provide optimal solutions subject to the operational and physical system constraints, they convey a heavy computational effort. This becomes particularly paramount to space processors with strict resource limits. Evaluating MPC algorithms onboard the servicing spacecraft in an online receeding horizon manner can thus be intractable. In the following we investigate the constrained imitation learning methodology within the separated relative-position and attitude control problem.

In order to reduce the online computational demand of MPC, we approximate its feedback policy by supervised imitation learning. Based on expert demonstrations from the MPC, a dataset of states and control inputs is gener-

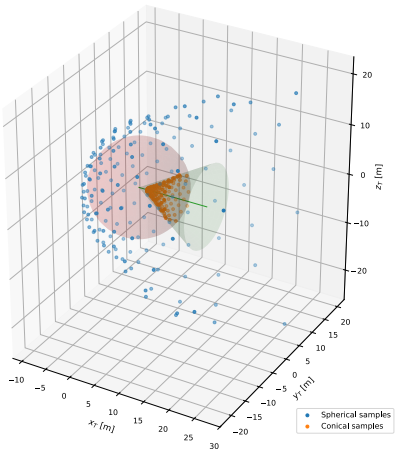
ated. The neural network (NN) is trained to learn the state-to-input mapping of the constrained optimal control problem while requiring a shorter computation time than the expert controller. In this study, we demonstrate the constrained imitation learning approach for the relative-translation MPC. The sweep over the initial positions covers both the rendezvous region outside the spherical KOZ and the target-fixed AC, with additional samples close to the respective constraint boundaries. For each sample, the network receives the relative position and velocity tracking error in the Hill frame, as derived in the translational equations of relative motion in (1). The feature vector is further augmented by second-order polynomial features and a binary operational mode (KOZ / AC) indicator. As output, the network predicts the control input acceleration. The data are divided into training, validation, and test sets, while the NN architecture and hyperparameters are managed and recorded with MLflow.

To account for the operational safety zones, constraint satisfaction is incorporated into the training and deployment of the learned policy through a differentiable QP layer. In contrast to a subsequent safety filter that modifies the nominal, learned input such that a number of constraints are guaranteed, we augment the feedfoward neural network by a QP safety layer. Hence, we build upon the results in [21] with the benefit of using the differentiable QP layer in the backpropagation during the training process. For the nominal network output $\hat{u}_{\mathrm{NN}}$, the applied input is obtained by solving

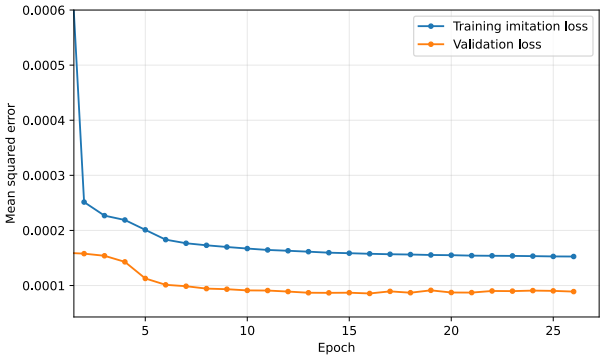
$$\begin{aligned} u_{\mathrm{safe}}(x) = \underset{u,\,\delta\geq 0}{\operatorname{argmin}} \; & \|u-\hat{u}_{\mathrm{NN}}(x)\|_2^2 + \rho\delta^2, \\ \text{s.t.} \quad & h_{\mathrm{CBF}}(x,u) \geq 0, \quad u \in \mathcal{U}, \\ & \dot{V}(x,u) \leq -\alpha V(x) + \delta. \end{aligned} \tag{19}$$

with hard Control Barrier Function constraints $h_{\mathrm{CBF}}(x,u)$ for safety guarantees and soft constraints for the Control Lyapunov Function $V(x,u)$ relaxed by a slack variable to preserve feasibility while promoting stable convergence to the desired reference. The active CBF is selected according to the operational phase: a spherical CBF enforces the KOZ during rendezvous, whereas a conical CBF keeps the servicer inside the AC starting from the approach phase, both derived with relative degree 2. An additional CBF with relative degree 1 constrains the relative velocity. The derivation of the CBFs is presented in [21]. The QP is constructed in CasADi ([31]), solved with DAQP ([32]), and differentiated during training, such that the imitation loss is evaluated on the constrained input and its gradient accounts for safety layer interventions. As the QP is quite small the low-level implementation exhibits lower runtimes compared to frameworks as OSQP's Torch layer, ProxSuite's QPLayer, dQP, OptNet, PiNet and jaxopt's BoxOSQP. The resulting policy, including the same QP layer, is then deployed in the Modelica/Dymola OOS simulation through its FMU interface. For the data generation and training of the neural policy we use the astrodynamics framework Basilisk [30], where we setup a simulation over the OOS mission phases. The neural network controller is deployed in the Modelica/Dymola simulation environment introduced in Section 4.1. While separating training and closed-loop evaluation using two different high-fidelity astrodynamics simulators does not claim to remove the sim-to-real gap, it provides a cross-simulator test of whether the learned policy generalized beyond the environment from which its demonstrations were obtained.

Figure 4 summarizes the data generation and training of the neural policy. The sampled initial positions cover both operational regions and include additional samples close to the KOZ and AC boundaries. In total, 400 initial samples and trajectories were generated, where half of it correspond to the KOZ as well as the AC, respectively. This leads to a total number of 266770 training samples considering the MPC expert trajectories. The neural policy is designed to switch the operational constraint mode based on a binary parameter in the feature input vector of the NN architecture. To this end, it is not required to train various imitation learning policies but rather one model that can account for multiple keep-out and keep-in zones throughout the OOS mission lifecycle. The safety constraint mode yields a switch of the active CBF constraint in the differentiable QP layer of the NN. Besides, the NN is modelled as a feedfoward network with 4 hidden layers and 256 neurons each, where ReLU activation functions and the Adam optimizer are implemented. A dropout rate of 0.1 during training and layer normalization are implemented to improve training stability. An overall RMSE of 0.01344 m/s^2 evaluates the performance of the neural policy on the test set, which corresponds to 10% of the training data. The training and validation losses decrease rapidly during the first epochs and subsequently converge to similar low values.



(a) *Distribution of sampled initial positions for imitation learning data generation, covering the rendezvous region outside the KOZ and the target-fixed AC with 200 initial samples each.*



(b) *Training and validation mean-squared imitation loss over the training epochs.*

Figure 4 *Dataset generation and training performance of the constrained imitation learning for the translational subsystem MPC. The model achieves an overall RMSE of 0.01344* kg m/s^2 *on the test set.*

4 Orbital Servicing Simulation

4.1 Simulation Environment

The On-Orbit Servicing Simulation Module is implemented in Modelica/Dymola and integrated as a Functional Mock-up Unit (FMU). The Modelica modeling approach decomposes the overall system into a set of acausal components. As shown in Figure 5, the top-level model consists of a world model based on the Environment Library [13], a servicer model, a target model, and a contact dynamics model implemented with the Manipulation2 Library [14], each equipped with visualization modules. The servicer model comprises the satellite dynamics and robotic-arm dynamics and can be extended to include additional multiphysics effects, such as thermal models. The satellite dynamics are further decomposed into rigid-body dynamics, sloshing dynamics, and flexible-appendage dynamics. An specific actuator dynamics is not included, but will be addressed in future work.

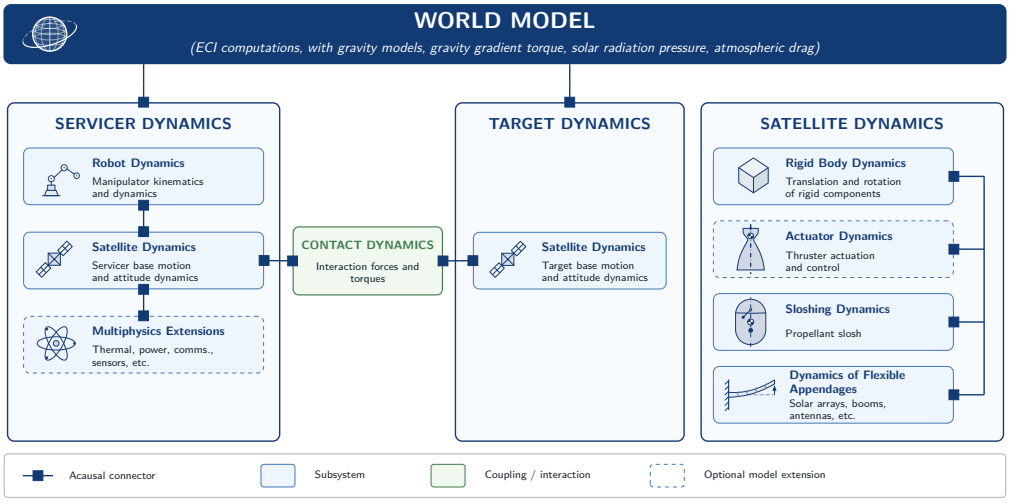


Figure 5 *Dymola/Modelica based On-Orbit Servicing Simulation Model*

4.2 Scenario Parametrization

We consider a scaled Envisat detumbling scenario with an initial target angular velocity of 3 deg s^{-1}. We assume that the current attitude and angular velocity of the target is estimated. Expressed in the target body frame, the initial angular velocity is

$$\boldsymbol{\omega}_T(0) = \begin{bmatrix} 3 & 0 & 0 \end{bmatrix}^\mathsf{T} \text{deg, s}^{-1},$$

corresponding to a rotation about an axis normal to the orbital plane. The target geometry is scaled down by a factor $s = 3$. Assuming geometric similarity and constant density, the mass and inertia scale as

$$m_{\text{scaled}} = \frac{1}{s^3} m_{\text{full}}, \qquad \mathbf{J}_{\text{scaled}} = \frac{1}{s^5} \mathbf{J}_{\text{full}}. \tag{20}$$

Hence, the target parameters used in the simulation are

$$m_{\text{target}} = \frac{1}{27} m_{\text{Envisat}}, \qquad \mathbf{J}_{\text{target}} = \frac{1}{243} \mathbf{J}_{\text{Envisat}}, \tag{21}$$

based on the Envisat mass and inertia properties reported in [15]. Since the inertia matrix expressed in the body-fixed frame contains off-diagonal terms, a tumbling motion occurs. Accordingly, a target-fixed approach corridor is implemented, which tumbles with the target reference frame.

The keep-out radius $r_{\text{KOZ}} = 9.17$m remains constant throughout the mission. The constraint half-angles depend on the active mission phase μ:

$$\left(\alpha_{\text{app}}, \alpha_{\text{FOV}}\right) = \begin{cases} (30°,\ 5°), & \mu \in \{\text{Rendezvous, Approach}\}, \\ (45°,\ 45°), & \mu \in \{\text{Tracking, Capture, Detumble}\}. \end{cases}$$

The servicer system is modeled with a 7 DoF robotic arm and adopts the geometric dimensions and robotic arm configuration of the OOS-Sim facility [16], enabling further validation of the proposed methods on the OOS-Sim hardware-in-the-loop facility. The spacecraft base is chosen to be comparatively light relative to the robotic arm in order to induce significant dynamic coupling between the base and the manipulator. The inertia tensor of the spacecraft base, expressed in the servicer frame about the base center of mass, and the corresponding mass are given by

$$H_r = \begin{bmatrix} 85 & 0 & 0 \\ 0 & 94 & 0 \\ 0 & 0 & 92 \end{bmatrix} \text{kg m}^2, \qquad m_s = 105 \text{ kg}.$$

4.3 Comparison

The three control architectures are evaluated in terms of constraint satisfaction, tracking performance, control effort, and computational demand, following the OOS simulation setup described above. Overall, it can be summarized that all proposed controller architectures (i)-(iii) meet the variety of operational and system constraints, while succeeding the mission objective of safely capturing and detumbling the Envisat target spacecraft.

As shown in Figure 6, all controllers satisfy the operational constraints throughout all mission phases regarding the translational motion. Each remains outside the target keep-out zone during rendezvous and enters the approach corridor without violating the conical constraint. Figure 7 shows that the target remains within the admissible FoV region for all three controllers. The largest boresight deviations occur during mission phase changes, particularly at the transition from approach to tracking. Although the transient responses differ, all configurations recover to a similar boresight-to-target angle in the following phases.

The translational responses in Figure 8 show proper tracking of the commanded operational reference positions for all architectures. Tracking errors, relative velocity deviations, and force peaks are concentrated around reference changes at the phase transitions. The base-coupling and combined control MPC exhibit nearly identical behavior because they are parameterized in the same way, as described in Section 3.1. The separate MPC configuration responds more aggressively during these intervals, producing greater force activity and, consequently, faster tracking at the expense of higher accumulated control effort.

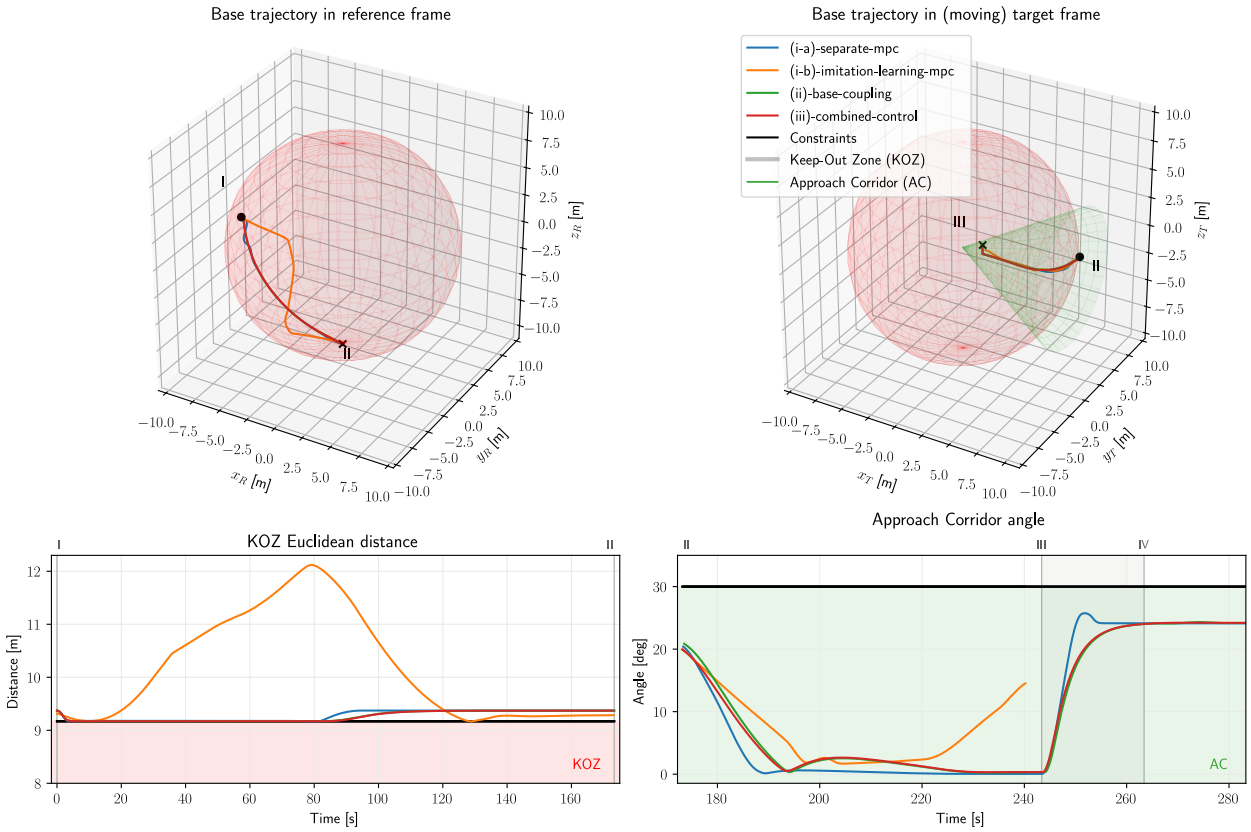


Figure 6 *Comparison of position-related operational-constraint satisfaction. Top left: Closed-loop position trajectory in the reference frame during rendezvous from* {**I**} *to* {**II**} *(Go for KOZ). Top right: Closed-loop position trajectory in the moving target frame (tumbling AC) during approach from* {**II**} *to* {**III**} *(Go for Capture). Bottom left: Euclidean distance to the keep-out zone. Bottom right: Approach-corridor angle over time.*

A similar pattern appears in the attitude response shown in Figure 9. All architectures track the commanded attitude and angular-rate references, with the largest deviations occurring during phase transitions. Torque inputs remain bounded and decay after each transient. The steady-state attitude behavior is therefore similar across the controllers, while their transient errors and accumulated angular control efforts differ.

The design and tuning of a dedicated manipulator controller is not the scope of this work; therefore, a computed-torque controller was adopted as a baseline. The combined control MPC achieves substantially better target-detumbling performance. The target inertia was incorporated into the combined control formulation using an exemplary estimate equal to 70% of the nominal value used in the simulation. All considered control approaches successfully detumble Envisat; however, the computed-torque controller exhibits damped post-capture oscillations. Future work will investigate the robustness with respect to contact dynamics in the capture and detumbling phase in greater detail.

Finally, Figure 11 compares the computational cost of the three architectures. The base-coupling with computed robotic torque configuration has the lowest total runtime, followed by combined controller, while the separate MPC implementation is the most expensive. The runtime violin plots indicate that the translational MPC is the main source of this difference; the attitude controller contributes only a small share of the total computation time, while the robotic arm controller has an intermediate computational cost. Imitating the translational MPC with the neural policy vastly improves the runtime and supports the motivation of incorporating learning-based and data-driven controllers to reduce the computational demand compared to constrained optimal controllers.

The neural policy satisfies all considered hard constraints, including the operational zones, relative velocity bounds, and input force limits, and transfers successfully from the Basilisk training environment to the Modelica/FMU simulator for deployment. However, it does not satisfy the transition criterion from the approach to the tracking phase due to a remaining tracking error in the relative position and is therefore evaluated only for

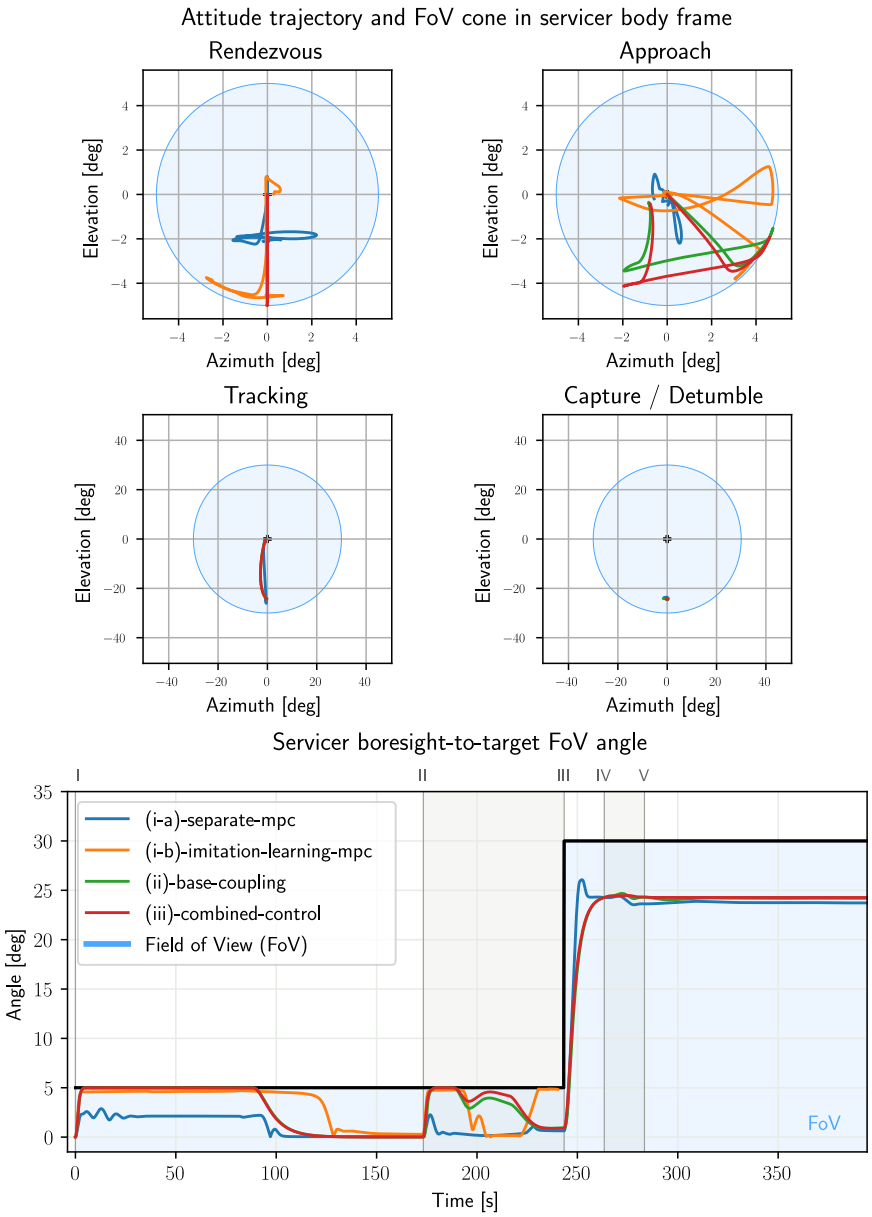


Figure 7 *Field-of-view constraint evaluation. Top: Target trajectory in the servicer body frame during the individual mission phases. Bottom: Corresponding boresight-to-target angle over time.*

the rendezvous and approach phases. Additionally, the chattering in the control input force can be explained by the combination of CBFs and CLFs, commonly seen in the safety filter literature, yet makes its application in spacecraft control impractical. Further tuning of the CLF parameters is recommended to reduce the control effort of the learning-based controller. As an outlook, the imitation-learning training setup should be extended from the expert MPC used here, which generated training data for a regulation problem, to an expert formulation that also addresses tracking problems, as required during the tracking, capture, and detumbling phases. This extension could enable the imitation-learning approach to cover all mission phases. Nevertheless, the constrained imitation-learning approach achieves a substantial reduction in runtime compared with the translational MPC expert while satisfying a combination of complex safety requirements relevant to OOS. These requirements go beyond the small-scale problems typically considered in state-of-the-art approaches using differentiable QP layers, thereby motivating further research in this direction.

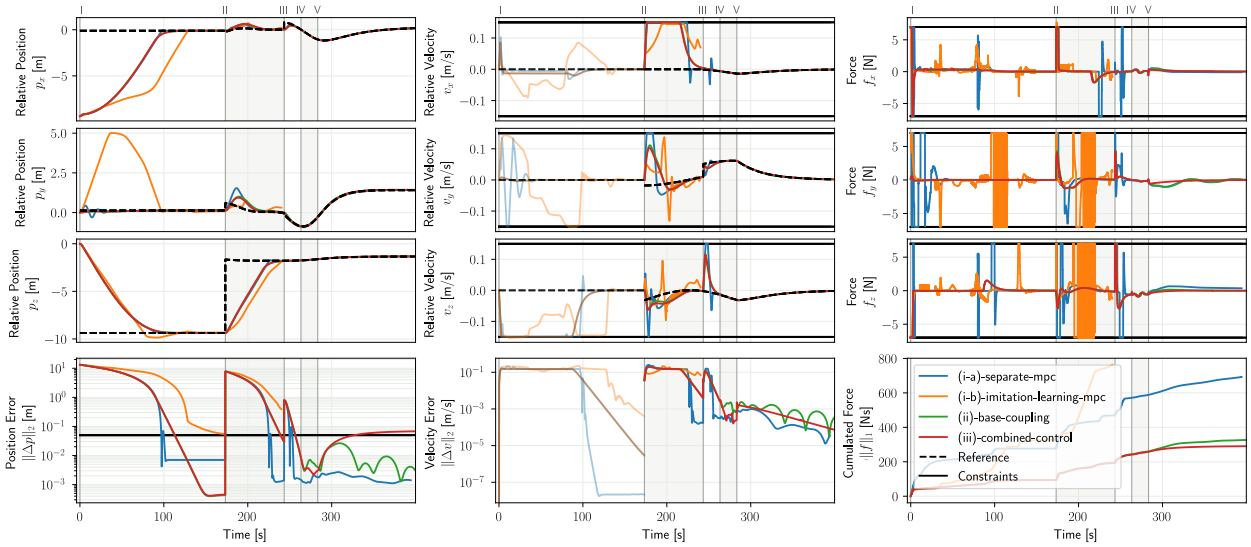


Figure 8 *Comparison of translational closed-loop behavior, showing relative position and velocity tracking, control input forces, tracking errors, and accumulated control effort.*

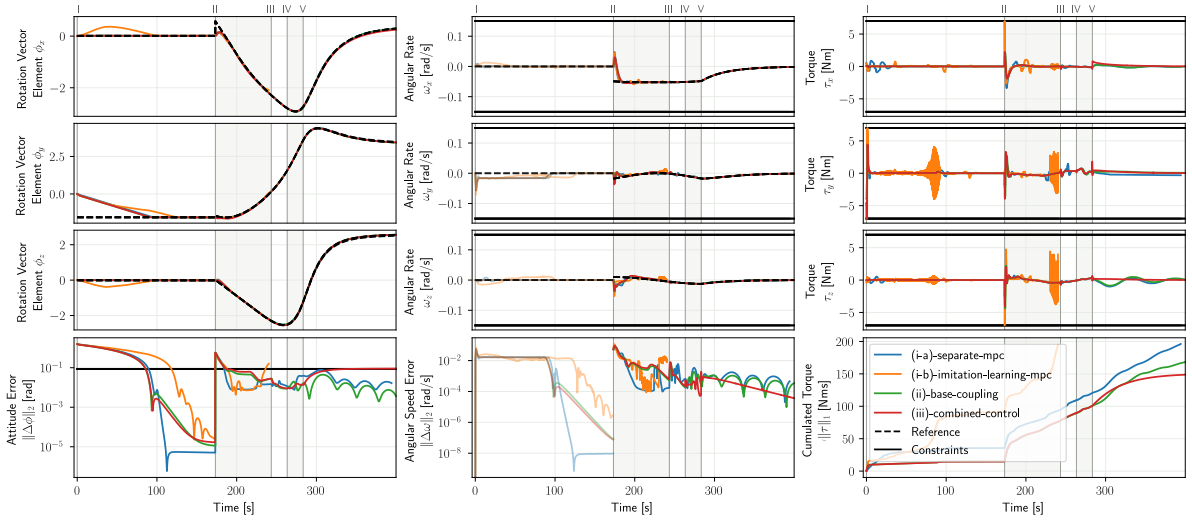


Figure 9 *Comparison of attitude control performance, including attitude and angular rate tracking, control input torques, tracking errors, and accumulated angular control effort.*

5 Conclusion

We have presented constrained motion-control approaches to advance the level of autonomy in robotic on-orbit servicing missions. A servicing spacecraft equipped with a robotic manipulator combines the position- and attitude-control problem of the free-flying spacecraft with the control of the manipulator, resulting in different levels of control complexity. To address these challenges, we implemented and compared MPC-based approaches for the considered levels of control complexity, ranging from fully separated to fully combined control. In addition, constrained imitation learning was employed to approximate the MPC policies for the separated control problem and thereby reducing the online computational demand. By using a representative detumbling scenario of ESA's Envisat satellite, we demonstrate that reliable closed-loop solutions can be obtained with MPC while satisfying all operational constraints. The imitation-learning approach successfully completes the rendezvous and approach phases, while future work will focus on extending this framework to the remaining mission phases. As second focus we are investigating robust MPC formulations and their analysis.

© *German Aerospace Center (DLR), e:fs TechHub GmbH*

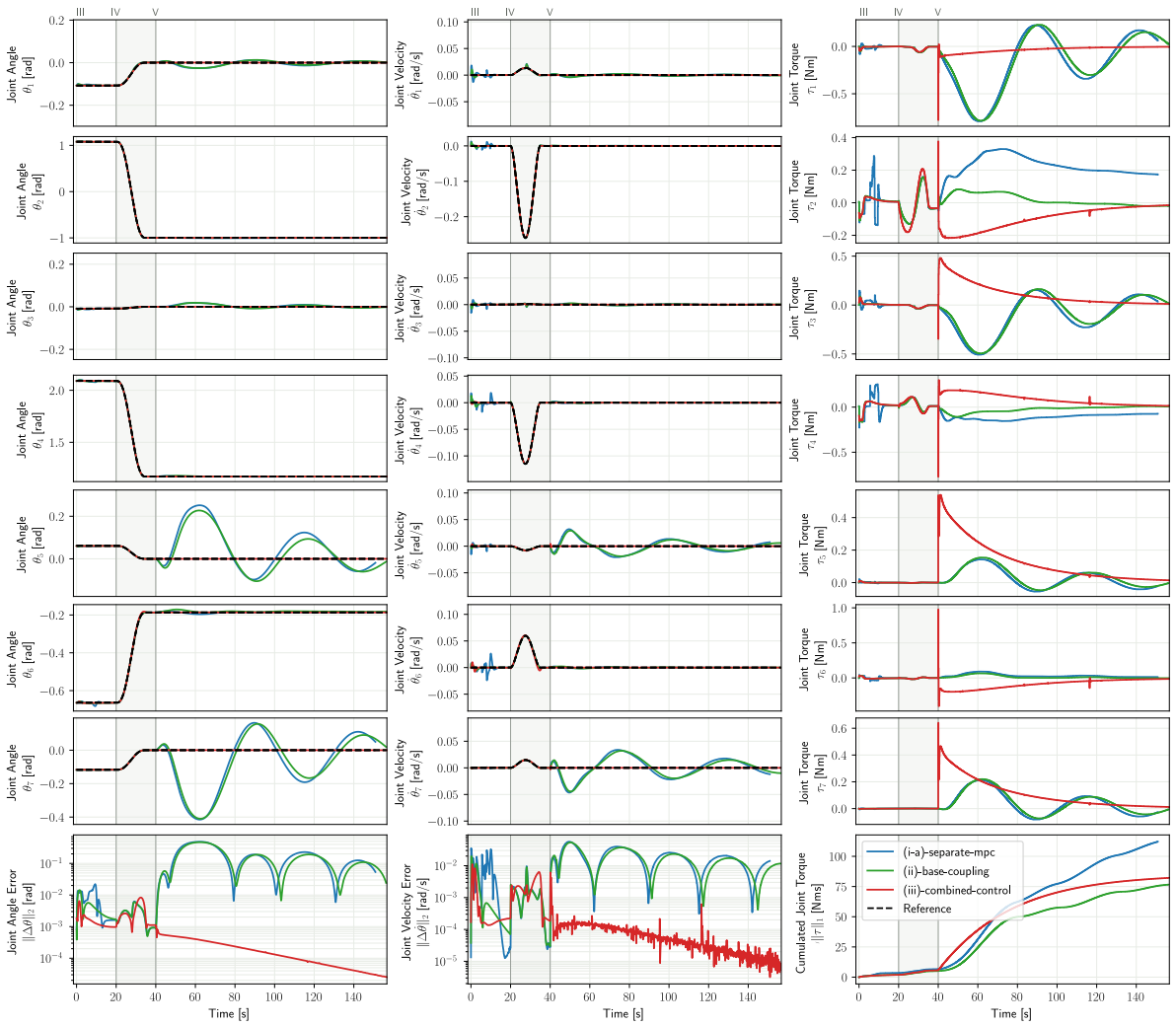


Figure 10 *Comparison of robotic manipulator motion for the different control architectures, showing joint positions, velocities, torques, tracking errors, and accumulated joint torque effort.*

6 Citations and references

References

[1] R. Biesbroek, L. Innocenti, A. Wolahan, and S. M. Serrano, "e.Deorbit—ESA's Active Debris Removal Mission," in *Proceedings of the 7th European Conference on Space Debris*, vol. 7, no. 1, Darmstadt, Germany, Apr. 2017, ESA Space Debris Office.

[2] S. Jaekel, R. Lampariello, W. Rackl, M. De Stefano, N. Oumer, A. M. Giordano, O. Porges, M. Pietras, B. Brunner, J. Ratti, Q. Muehlbauer, M. Thiel, S. Estable, R. Biesbroek, and A. Albu-Schaeffer, "Design and Operational Elements of the Robotic Subsystem for the e.Deorbit Debris Removal Mission." *Frontiers in Robotics and AI*, vol. 5, Art. no. 100, Aug. 2018, `https://doi.org/10.3389/frobt.2018.00100`.

[3] S. Woicke, J. Jipp, A. Winkler, and C. Steimle, "ADRIOS ClearSpace-1: In Orbit Demonstration of the Removal of a Non-Cooperative Spacecraft," in *Proceedings of the 9th European Conference on Space Debris*, vol. 9, no. 1, Bonn, Germany, Apr. 2025, ESA Space Debris Office.

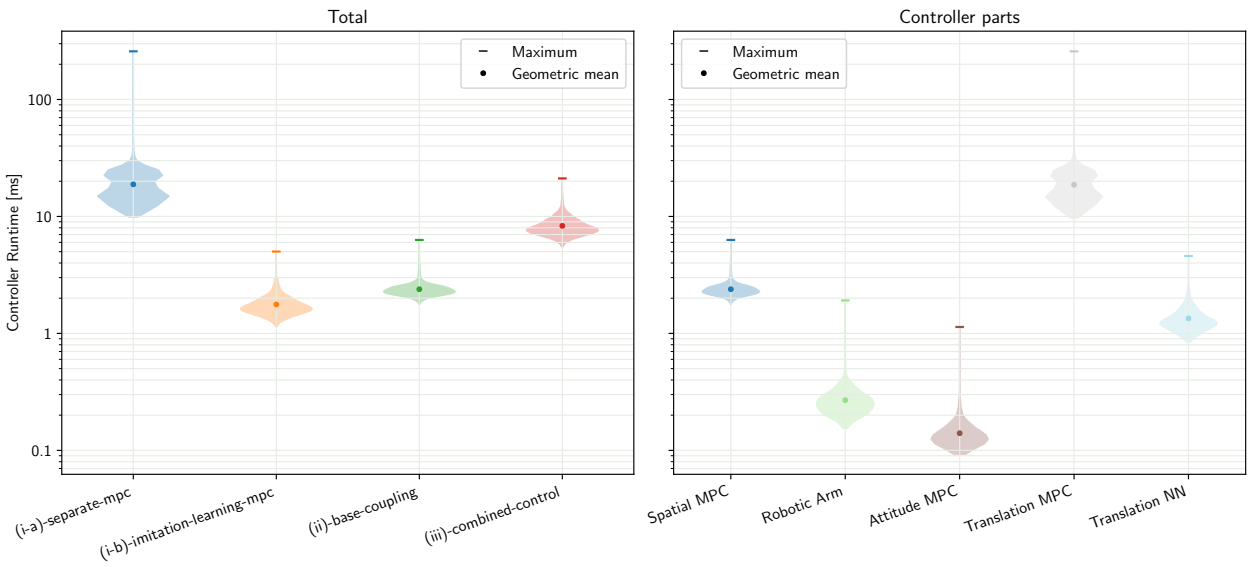


Figure 11 *Computational runtime comparison evaluated using violin plots, showing the maximum and geometric mean. Left: Total runtime of the control architectures. Right: Runtime of the individual controller components.*

[4] J. Mulvaney, D. Arney, C. Williams, J. Morel, C. Stockdale, C. Whitlock, and V. Balaji, "In-Space Servicing, Assembly, and Manufacturing (ISAM): State of Play, 2025 Edition, "NASA, White Paper, Document ID 20250008988, Oct. 2025. [Online]. Available: https://ntrs.nasa.gov/citations/20250008988.

[5] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*. Cambridge, U.K.: Cambridge University Press, 2003, https://doi.org/10.1017/CBO9780511543388.

[6] C. Specht, A. Bishnoi, and R. Lampariello, "Autonomous Spacecraft Rendezvous Using Tube-Based Model Predictive Control: Design and Application." *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 7, pp. 1243–1261, 2023, https://doi.org/10.2514/1.G007280.

[7] P. M. Kötting, "Model Predictive Control Based Coordinated Control for Free-Flying Space Manipulator Systems." *IFAC-PapersOnLine*, vol. 58, no. 18, pp. 133–138, 2024, https://doi.org/10.1016/j.ifacol.2024.09.021.

[8] J. Köhler, M. A. Müller, and F. Allgöwer, "A Nonlinear Model Predictive Control Framework Using Reference Generic Terminal Ingredients." *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp. 3576–3583, August 2020, https://doi.org/10.1109/TAC.2019.2949350.

[9] J. Löfberg, "YALMIP: A Toolbox for Modeling and Optimization in MATLAB," in *Proceedings of the 2004 IEEE International Symposium on Computer Aided Control Systems Design*, Taipei, Taiwan, pp. 284–289, September 2004, https://doi.org/10.1109/CACSD.2004.1393890.

[10] MOSEK ApS, *MOSEK Optimization Toolbox for MATLAB*, Version 11.2, Copenhagen, Denmark, 2026. [Online]. Available: https://docs.mosek.com/latest/toolbox/index.html.

[11] S. Shin and V. M. Zavala, "Diffusing-Horizon Model Predictive Control." *IEEE Transactions on Automatic Control*, vol. 68, no. 1, pp. 188–201, January 2023, https://doi.org/10.1109/TAC.2021.3137100.

[12] L. Vanroye, A. Sathya, J. De Schutter, and W. Decré, "FATROP: A Fast Constrained Optimal Control Problem Solver for Robot Trajectory Optimization and Control." in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10036–10043, Detroit, MI, USA, October 2023, https://doi.org/10.1109/IROS55552.2023.10342336.

[13] L. E. Briese, A. Klöckner, and M. Reiner, “The DLR Environment Library for Multi-Disciplinary Aerospace Applications.” in *Proceedings of the 12th International Modelica Conference*, pp. 929–938, Prague, Czech Republic, May 2017, https://doi.org/10.3384/ecp17132929.

[14] R. M. Reiser, “Stable and Efficient Simulation of Manipulation and Assembly Processes in Object-Oriented Modeling Languages.” Ph.D. dissertation, Technical University of Munich (TUM), Munich, Germany, 2026, https://doi.org/10.14459/2026md1839788.

[15] P. Arroz and F. Gandía, “DETUMBLING: Executive Summary.” GMV, ESA Contract No. 4000113022/14/NL/MV, Version 1.0, September 2017.

[16] J. Artigas, M. De Stefano, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O. Porges, A. Giordano, C. Borst, and A. Albu-Schäffer, “The OOS-SIM: An On-Ground Simulation Facility for On-Orbit Servicing Robotic Operations.” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015, pp. 2854–2860, https://doi.org/10.1109/ICRA.2015.7139588.

[17] European Space Agency (ESA), “Guidelines on Safe Close Proximity Operations,” ESA-TECSYE-TN-022522, Issue 3, Rev. 0, Noordwijk, The Netherlands, Apr. 2024.

[18] J. Vasconcelos, S. Gaggi, T. Amaral, C. Bakouche, A. Cotuna, and A. Friaças, “Close-Proximity Operations Design, Analysis, and Validation for Non-Cooperative Targets with an Application to the ClearSpace-1 Mission,” *Aerospace*, vol. 12, no. 1, Art. no. 67, Jan. 2025, https://doi.org/10.3390/aerospace12010067.

[19] NASA, Roscosmos, JAXA, ESA, and CSA, *International Rendezvous System Interoperability Standards (IRSIS)*, Baseline, Mar. 2019.

[20] ISO 24330, "Space systems – Rendezvous and Proximity Operations (RPO) and On Orbit Servicing (OOS) – Programmatic principles and practices". *International Organization for Standardization*, 2022.

[21] A. Meinert, N. Baldauf, P. Stadler, and A. Turnwald, "Safety-Guaranteed Imitation Learning from Nonlinear Model Predictive Control for Spacecraft Close Proximity Operations". in *2026 European Control Conference (ECC)*, pp. 1453–1458, July 2026.

[22] S. Yang, S. Chen, V. M. Preciado, and R. Mangharam, “Differentiable Safe Controller Design Through Control Barrier Functions,” *IEEE Control Systems Letters*, vol. 7, pp. 1207–1212, 2023, https://doi.org/10.1109/LCSYS.2022.3233322

[23] W. Xiao, T.-H. Wang, R. M. Hasani, M. Chahine, A. Amini, X. Li, and D. Rus, “BarrierNet: Differentiable Control Barrier Functions for Learning of Safe Robot Control,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2289–2307, Jun. 2023, https://doi.org/10.1109/TRO.2023.3249564.

[24] A. Meinert, N. Baldauf, and A. Turnwald, "Robust Spacecraft Attitude Reference Tracking using Tube-Based Model Predictive Control". *2024 28th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 229–234, October 2024, https://doi.org/10.1109/ICSTCC62912.2024.10744677

[25] P. Stadler, A. Meinert, N. Baldauf, and A. Turnwald, "Lightweight Model Predictive Control for Spacecraft Rendezvous Attitude Synchronization". *2026 European Control Conference (ECC)*, pp. 3748–3753, July 2026.

[26] J. A. Rebollo, R. Vazquez, I. Alvarado and D. Limon, "Rendezvous Control for Non-Cooperative Tumbling Targets with Stability and Feasibility Guarantees," *Journal of Guidance, Control, and Dynamics*, vol. 49, no. 9, pp. 2494–2509, 2026, https://doi.org/10.2514/1.G008974

[27] M. Sharifi and S. Heshmati-Alamdari, "Safe Force/Position Tracking Control via Control Barrier Functions for Floating Base Mobile Manipulator Systems," in *2024 European Control Conference (ECC)*, pp. 3650–3655, 2024, https://doi.org/10.23919/ECC64448.2024.10591297

[28] J. L. Loettgen, K. Worrall, G. Aragon-Camerasa, M. Ceriotti, and R. Lampariello, "Control Barrier Functions for Safe Real-Time Control of Spacecraft Onboard Robotic Manipulators," in *2025 International Conference on Space Robotics (iSpaRo)*, pp. 539–546, 2025, https://doi.org/10.1109/iSpaRo66239.2025.11437383

[29] A. Turnwald, J. Scharnagl, P. Stadler, and A. N. Pečujlić, "Model Predictive Attitude Control for CubeSats: Feasibility and Requirements for In-Orbit Testing," in *Small Satellites Systems and Services Symposium (4S 2024)*, Proc. SPIE, vol. 13546, Art. no. 135463X, Mar. 2025, https://doi.org/10.1117/12.3062770.

[30] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A Flexible, Scalable and Modular Astrodynamics Simulation Framework," *Journal of Aerospace Information Systems*, vol. 17, no. 9, pp. 496–507, Sep. 2020, https://doi.org/10.2514/1.I010762.

[31] 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, https://doi.org/10.1007/s12532-018-0139-4.

[32] D. Arnström, A. Bemporad, and D. Axehill, "A Dual Active-Set Solver for Embedded Quadratic Programming Using Recursive LDL^T Updates," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4362–4369, Aug. 2022, https://doi.org/10.1109/TAC.2022.3176430.

© *German Aerospace Center (DLR), e:fs TechHub GmbH*