Control Engineering Practice 135 (2023) 105515

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Adaptive control for a payload carrying spacecraft with state constraints

Viswa Narayanan Sankaranarayanan [a,*], Avijit Banerjee [a], Sumeet Satpute [a], Spandan Roy [b], George Nikolakopoulos [a]

[a] *Robotics and AI Team, Department of Computer Science, Electrical Engineering and Space Engineering, Lulea University of Technology, Lulea, Sweden*
[b] *Robotics Research Center, International Institute of Information Technology, Hyderabad, India*

ARTICLE INFO

Keywords:
Space robots
Nonlinear control
Barrier control
Barrier Lyapunov function
Euler–Lagrangian system

ABSTRACT

In this article, a novel adaptive trajectory tracking controller is designed for a payload-carrying spacecraft under full state constraints. The proposed controller can tackle state-dependent uncertainties without a priori knowledge of their structures and upper bounds. The controller ensures time-varying constraints on all states and their time derivatives. The closed-loop stability of the proposed scheme is verified analytically via the Lyapunov method, and real-life experiments using a robotic testbed validated the effectiveness of the proposed adaptive controller over the state-of-the-art.

1. Introduction

The scope of space robotics is evolving rapidly with continuous technological advancements in the research of flying robots. The space community is swiftly approaching the realization of potential futuristic goals of autonomous space-robotic missions. Microgravity aerial robots are viable aids in transportation and other logistics operations in the envisioned Deep Space Gateway (DSG) (cf. Correia & Ventura, 2021). Moreover, the increase in orbital space activities introduced a demand for immediate strategies for space debris removal. The ongoing trend further predicts possible applications of space robots in other critical operations such as orbit servicing, manufacturing, and maintenance (Daniel, Michel, Sabine, Raffaele, & Javier, 2015).

Tracking any arbitrary trajectory without violation will be an expected property of the space robot. Such a constraint in its position ensures that the robot does not collide with any obstacle and operates efficiently. Similarly, the orientation constraint is mandatory for visual locking on a target and effectively detecting and locating the obstacles in a given path. Further, most of these applications require the space robot to carry or operate with a payload. Such an operation introduces rapid variation in the system dynamics along with the modeling uncertainties, actuator nonlinearity and irregularities in the thruster outputs, among others. These uncertainties enhance the difficulties in maintaining the state constraints, which motivate us toward state-constrained control techniques which can also tackle substantial variations in the system dynamics.

There have been many notable contributions to robust (cf. Du et al., 2020; Huang, Yan, & Huang, 2017; Kai, 2018; Kraïem, Rognant, Biannic, & Brière, 2021; Malekzadeh & Sadeghian, 2019; Shi, Yao, Shan, Gao, & Jin, 2022) and adaptive controllers (cf. Al Issa & Kar, 2021; Chu, Ma, & Cui, 2018; Wang, Xu, Cheng, Wang, & Sun, 2022; Zhang, Liu, Gao, & Ju, 2020) for space robots to deal with the parametric uncertainties and disturbances. However, these robust control techniques require a nominal estimate of the dynamic parameters along with the upper bounds on the uncertainties, whereas the adaptive controllers require an a priori knowledge of the structure of the uncertainties. Such requirements are often difficult to be fulfilled. In other robotic systems, adaptive controllers (cf. Roy, Baldi, & Fridman, 2020; Sankaranarayanan, Roy, & Baldi, 2020) have been proposed to tackle uncertainties without any a priori knowledge of uncertainties. However, these traditional robust and adaptive controllers need extensive tuning to keep the states or tracking errors in a predetermined desired range, making them unsuitable for applications where state constraints are to be imposed.

In view of the above discussions, Barrier Lyapunov Function (BLF) based controllers have become a popular solution to handle state constraints, leading to guaranteeing a prescribed performance in continuous-time dynamical systems (Sachan & Padhi, 2020; Shahna & Abedi, 2021; Yao, 2021; Zhang & Tang, 2018). Nevertheless, for the feasibility of control solutions dealing with state constraints, initial values must lie within the constraints. Consequently, BLF-based works dealing with fixed-value constraints were found to be unfavorable in practice and, as a solution, time-varying state constraints have been employed recently in Gao, Liu, Jing, and Dimirovski (2021a, 2021b), Wu, Zhang, and Wu (2021) and Sankaranarayanan, Yadav, Swayampakula, Ganguly, and Roy (2022). Unfortunately, these controllers cannot handle unknown state-dependent uncertainties that are not a priori bounded. Note that state-dependent uncertainties occur naturally in Euler–Lagrange systems (cf. Roy et al., 2020; Spong, Hutchinson,

[*] Corresponding author.
E-mail address: vissan@ltu.se (V.N. Sankaranarayanan).

https://doi.org/10.1016/j.conengprac.2023.105515
Received 1 August 2022; Received in revised form 15 March 2023; Accepted 21 March 2023
Available online xxxx

& Vidyasagar, 2008) and hence, in the concerned space robots (cf. discussion in Remark 4).

In light of the above observations, and to the best of the authors' knowledge, an adaptive controller for a payload-carrying spacecraft under state constraints and, in the presence of unknown state-dependent uncertainties and of unknown external disturbances, is still missing in the literature. With this premise, the main contribution of this work is a BLF-based adaptive controller that (i) ensures a predefined accuracy on all the states and their derivatives via time-varying constraints; (ii) can tackle state-dependent uncertainties and external disturbances without their a priori knowledge. The closed-loop stability analysis is carried out via the Lyapunov method, and the effectiveness of the proposed adaptive controller is validated against the state-of-the-art via extensive real-time experiments using a three degrees-of-freedom (DoFs) planar floating satellite platform (Slider Banerjee et al., 2022).

The rest of the paper is organized as follows: Section 2 introduces the model of the space robot and the control problem; Section 3 presents the design and analysis of the proposed controller; Section 5 presents the experimental results, and Section 6 provides concluding remarks.

The following notations are used in this paper: the saturation function $\mathrm{sat}(s,\epsilon) = (s/\|s\|)$ if $\|s\| \geq \epsilon$ and $\mathrm{sat}(s,\epsilon) = (s/\epsilon)$ if $\|s\| < \epsilon$ with $\epsilon \in \mathbb{R}^{+}$; $\|(\cdot)\|$ and $\lambda_{\min}(\cdot)$ denote 2-norm and minimum eigenvalue of $(\cdot)$, respectively; I denotes identity matrix with appropriate dimension and $\mathrm{diag}\{\cdot,\ldots,\cdot\}$ denotes a diagonal matrix with diagonal elements $\{\cdot,\ldots,\cdot\}$; the variable i is used for indexing the states, such that $i \in \{x, y, \theta\}$.

2. Modeling of the slider robot and problem formulation

The Slider is a robotic testbed that emulates the zero-gravity motion of a spacecraft over a flat friction-less surface, as shown in Fig. 1. It hovers over a smooth flat surface with the support of three air bearings placed on the bottom of the platform at 120^{o} angle from each other. The air bearings consist of a porous function face through which compressed air is released uniformly to generate a micrometer-thick air cushion between the air bearings and a smooth table top, providing frictionless motion to the Slider platform (Banerjee, Haluska, Satpute, Kominiak, & Nikolakopoulos, 2021). The platform has 3 DoFs given by its two-dimensional position $\mathbf{p(t)} \triangleq [x(t), y(t)]^{T}$ and orientation (heading) $\theta(t)$ along the $\mathbf{Z_B}$-axis in the inertial frame $(\mathbf{X}-\mathbf{Y}-\mathbf{Z})$. The origin of the inertial frame is located at the center of the experimental table, and its $\mathbf{X}, \mathbf{Y}$ axes are aligned with the East and North directions, respectively. The $\mathbf{Z}$ axis is chosen by the right-hand rule pointing upward. The origin of the body frame is located at the center of gravity of the unloaded Slider. The $\mathbf{Z}$-axis of the body frame $(\mathbf{Z_B})$ is aligned with that of the inertial frame. $\mathbf{X_B}$ axis is aligned along the nozzle of the air chamber as shown in Fig. 1(a). The slider is actuated using eight unidirectional thrusters placed along the $\mathbf{X_B}-\mathbf{Y_B}$ plane 1(b), whose mapping is provided in Table 1 (CW and ACW denote clockwise and anti-clockwise directions, respectively). Each thruster releases compressed air when actuated, providing a force in the opposite direction. A combination of two or more thrusters is actuated to generate the desired forces and moment. The platform is designed to be symmetric to distribute its weight equally on the bearings and to maintain the Center of Gravity along the $\mathbf{Z_B}$-axis.

The Euler–Lagrangian system dynamics (cf. Spong et al., 2008 for more details) of the slider platform is given by,

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{d} = \boldsymbol{\tau} \tag{1}$$

where $\mathbf{q}(t) \triangleq [x(t), y(t), \theta(t)]^{T} \in \mathbb{R}^{3}$ is the pose of the Slider in the inertial frame; $\mathbf{M} \triangleq \mathrm{diag}(m, m, I_{zz}) \in \mathbb{R}^{3\times 3}$ is the inertia matrix with m being the mass and I_{zz} being the inertia along the $\mathbf{Z}$ axis of the slider; the vector $\mathbf{d}(t) \in \mathbb{R}^{3}$ is the unknown external disturbances and $\boldsymbol{\tau}(t) \in \mathbb{R}^{3}$ is the control input in the inertial frame. The actuation from the thrusters are directly mapped to forces $(f_x(t), f_y(t))$, and moment $(\tau_z(t))$ in the body-frame via the following relation

Table 1
Mapping between thruster actuation and direction of motion.

Direction	Thrusters							
	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
$+\mathbf{X}_B$	0	0	1	0	1	0	0	0
$-\mathbf{X_B}$	1	0	0	0	0	0	1	0
$+\mathbf{Y}_B$	0	1	0	1	0	0	0	0
$-\mathbf{Y}_B$	0	0	0	0	0	1	0	1
$+\theta$ (CW)	1	0	0	1	1	0	0	1
$-\theta$ (ACW)	0	1	1	0	0	1	1	0

$$\boldsymbol{\tau} = \mathbf{R}\begin{bmatrix} f_x \\ f_y \\ \tau_z \end{bmatrix}, \tag{2}$$

where $\mathbf{R}$ is the orthogonal rotation matrix from the body-frame to the inertial frame, given by,

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

The relationship between the thruster outputs and the control inputs in the body frame is given by

$$\begin{bmatrix} f_x \\ f_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} \sum_{l=1}^{8} T_l \cos\beta_l \\ \sum_{l=1}^{8} T_l \sin\beta_l \\ \left(\sum_{l=1}^{8} \left(T_l r_{T_l}^{y} \cos\beta_l - T_l r_{T_l}^{x} \cos\beta_l \right) \right) \end{bmatrix} \tag{4}$$

where subscript $l \in \{1,\ldots,8\}$ denotes the respective lth thruster for which T_l is the thrust magnitude; $(r_{T_l}^{x}, r_{T_l}^{y})$ indicates its position in the $\mathbf{X_B}-\mathbf{Y_B}$ plane and β_l represents its orientation with respect to the $\mathbf{X_B}$ axis. The velocity mapping between the body frame and the inertial frame is given by,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \mathbf{R}\begin{bmatrix} v_x \\ v_y \\ v_\theta \end{bmatrix} \tag{5}$$

where (v_x, v_y) represent the linear velocities along $(\mathbf{X}_B, \mathbf{Y}_B)$ axes and v_θ is the rotational velocity along $\mathbf{Z}_B$.

The dynamic system (1) satisfies the following properties (cf. Spong et al., 2008, Sect. 9.5):

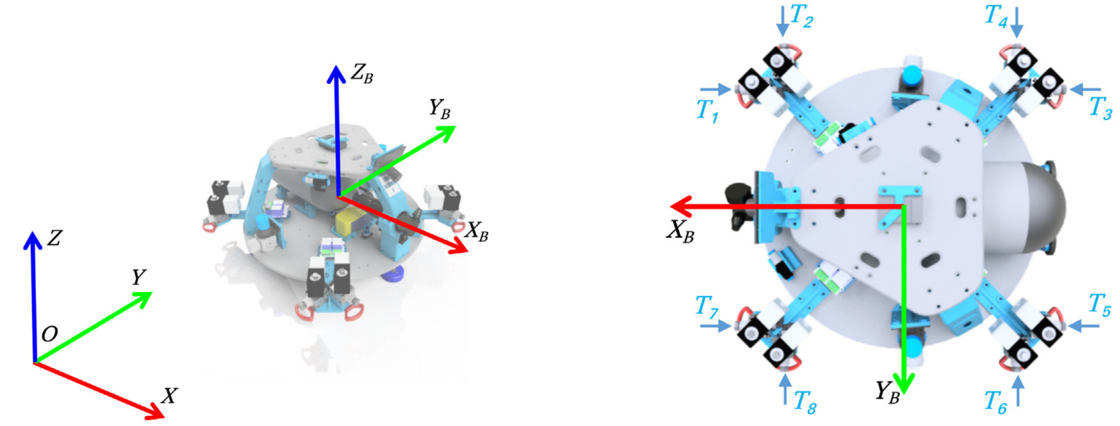
Property 1. *The mass-inertia matrix, defined by* $\mathbf{M}$ *is uniformly positive definite and there exist* $\underline{m}, \bar{m} \in \mathcal{R}^{+}$ *such that* $0 \leq \underline{m}\mathbf{I} \leq \mathbf{M} \leq \bar{m}\mathbf{I}$.

Property 2. *There exists* $\bar{d} \in \mathcal{R}^{+}$ *such that* $\|\mathbf{d}\| \leq \bar{d}$.

Further, let us state the available parametric knowledge of the systems for control design:

Assumption 1 (*Uncertainties*)**.** The dynamic terms, $\mathbf{M}, \mathbf{d}$ and their bounds $\underline{m}, \bar{m}, \bar{d}$ are unknown for control design.

Remark 1. Assumption 1 gets rid of the need for any a priori knowledge of system dynamics parameters and of external disturbances. The assumption is indeed a control design challenge. However, the location and orientation of each of the thrusters (canting angle) are known (cf. Table 1) and used for mapping the forces in the body frame to the thruster pulses; torque mapping is performed assuming no load conditions.



(a) Three-dimensional representation of the Slider with coordinate frames

(b) Top-view of the Slider: The blue arrows represent the direction of thrust corresponding to the thruster

Fig. 1. Slider platform with the inertial frame, $\mathbf{X-Y-Z}$ and body frame, $\mathbf{X_B-Y_B-Z_B}$ and the locations and directions of thrusters.

Assumption 2. The desired trajectories $\{x_{des}(t), y_{des}(t), \theta_{des}(t)\}$ are designed to be smooth and bounded (Wu et al., 2021).

The control problem of maintaining the states within predefined constraints can be equivalently posed as keeping the tracking errors within a predefined accuracy (Sankaranarayanan, Yadav, et al., 2022; Wu et al., 2021). Accordingly, the tracking errors are defined as $\mathbf{e} \triangleq \begin{bmatrix} e_x & e_y & e_\theta \end{bmatrix}^T$ where $e_i = i - i_{des}, \quad i \in \{x, y, \theta\}$ and time-varying error bounds (a.k.a. constraints or predefined accuracy) b_i and $b_{di}, \quad i \in \{x, y, \theta\}$ as:

$$b_i(t) = (\rho_{0i} - \rho_{ssi})e^{-\alpha_i t} + \rho_{ssi}, \ \ \rho_{0i}(0) > e_i(0) \tag{6a}$$

$$b_{di}(t) = (\rho_{0di} - \rho_{ssdi})e^{-\alpha_{di} t} + \rho_{ssdi}, \ \ \rho_{0di}(0) > \dot{e}_i(0) \tag{6b}$$

where (ρ_{0i}, ρ_{di}) are the initial values and $(\rho_{ssi}, \rho_{ssdi})$ are the steady-state values of the respective constraint bounds; $\alpha_{di}, \ \alpha_i \in \mathbb{R}^+$ are user-defined scalars which decide the rate of convergence of the error bounds.

Remark 2 (*Importance of Time-varying Constraints*)**.** Since the tracking errors are needed to be constrained within their upper bounds, a necessary condition to formulate a barrier-Lyapunov function is that the initial errors remain within the bounds 2. This condition is unfavorable while using a standard constant barrier (or error bound). Hence, a time-varying bound whose initial value is greater than the initial error and converges exponentially to the prescribed accuracy is well-suited. Also, as the parameter α_i, α_{di} is design-specific, it enables the user to decide the convergence rate to the steady-state error bound based on the mechanical constraints. Nevertheless, faster convergence will demand higher initial control input and thus, α_i, α_{di} should be designed as per application requirement.

Remark 3 (*Constraints on the Velocity Errors*)**.** It can be noted that b_{di}, in (6b), is a separate constraint on the velocity tracking errors and not a time-derivative of the constraints on the tracking errors in (6a). Having a separate constraint on the velocity tracking errors allows to ensure a smooth operation of the robot.

Subsequently, the control problem is defined as the following:

Control Problem: Design an adaptive controller for the floating satellite platform (slider) under Property 1 to track a desired trajectory (cf. Assumption 2) within a time-varying pre-defined accuracy as defined in 2 without a priori knowledge of the system dynamic parameters and their bounds (cf. Assumption 1).

3. Proposed controller design

The implementation of the controller is divided into two levels, namely, Force and Moment Generation and Thruster Selection Logic. The Force and Moment Generation deals with the generation of control inputs in the inertial frame, and the Thruster Selection Logic transforms these control inputs into body-frame and finally into thruster logic. The proposed controller framework is elaborated in the following subsections.

3.1. Force and moment generation:

Let us define the tracking error as

$$\mathbf{s} = \mathbf{z_b}\dot{\mathbf{e}} + \boldsymbol{\Phi}\left(\mathbf{z_a}\mathbf{e}\right) \tag{7a}$$

$$\mathbf{z_b} = \mathrm{diag}\left\{\frac{1}{b_{dx}^2 - \dot{e}_x^2}, \ \frac{1}{b_{dy}^2 - \dot{e}_y^2}, \ \frac{1}{b_{d\theta}^2 - \dot{e}_\theta^2}\right\}, \tag{7b}$$

$$\mathbf{z_a} = \mathrm{diag}\left\{\frac{1}{b_x^2 - e_x^2}, \ \frac{1}{b_y^2 - e_y^2}, \ \frac{1}{b_\theta^2 - e_\theta^2}\right\}, \tag{7c}$$

where $\boldsymbol{\Phi}$ is a user-defined positive definite gain matrix. Using (1), 2 and 3.1, the following relationship can be obtained

$$\dot{\mathbf{s}} = \mathbf{z_b}\left(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_{des}\right) + (\dot{\mathbf{z}}_\mathbf{b} + \boldsymbol{\Phi}\mathbf{z_a})\dot{\mathbf{e}} + \boldsymbol{\Phi}\dot{\mathbf{z}}_\mathbf{a}\mathbf{e} \tag{8}$$

where

$$\dot{\mathbf{z}}_\mathbf{a} = \mathrm{diag}\left\{\dot{z}_{ax}, \dot{z}_{ay}, \dot{z}_{a\theta}\right\},$$
$$\dot{z}_{ai} = -\frac{2b_i\dot{b}_i - 2e_i\dot{e}_i}{(b_i^2 - e_i^2)^2},$$
$$\dot{b}_i = -\alpha_i(\rho_{0i} - \rho_{ssi})e^{-\alpha_i t}, \ \forall i \in \{x, y, \theta\}.$$

The terms $\mathbf{z_a}, \mathbf{z_b}$ can be rewritten as,

$$\dot{\mathbf{z}}_\mathbf{a} = \mathbf{z}_a^2(\mathbf{p}_a + 2\,\mathrm{diag}\,\{\mathbf{e}\}\,\mathrm{diag}\,\{\dot{\mathbf{e}}\}) \tag{9}$$

where $\mathbf{p}_a = -2\,\mathrm{diag}\left\{b_x\dot{b}_x, \ b_y\dot{b}_y, \ b_\theta\dot{b}_\theta\right\},$

$$\dot{\mathbf{z}}_\mathbf{b} = \mathbf{z}_b^2(\mathbf{p}_b + 2\,\mathrm{diag}\,\{\dot{\mathbf{e}}\}\,\mathrm{diag}\,\{\ddot{\mathbf{e}}\}), \tag{10}$$

where $\mathbf{p}_b = -2\,\mathrm{diag}\left\{b_{dx}\dot{b}_{dx}, \ b_{dy}\dot{b}_{dy}, \ b_{d\theta}\dot{b}_{d\theta}\right\}.$

Multiplying both sides of (8) by $\mathbf{M}$ and substituting (9) and (10) gives

$$\begin{aligned}\mathbf{M}\dot{\mathbf{s}} &= \mathbf{M}(\mathbf{z_b}\left(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_d\right) + (\mathbf{z}_b^2(\mathbf{p}_b + 2\,\mathrm{diag}\,\{\dot{\mathbf{e}}\}\,\mathrm{diag}\,\{\ddot{\mathbf{e}}\}) \\ &\quad + \boldsymbol{\Phi}\mathbf{z_a})\dot{\mathbf{e}} + \boldsymbol{\Phi}\mathbf{z}_a^2(\mathbf{p}_a + \mathrm{diag}\,\{\mathbf{e}\}\,\mathrm{diag}\,\{\dot{\mathbf{e}}\})\mathbf{e}) \\ &= \mathbf{M}((\mathbf{z_b} + 2\mathbf{z_b}^2\,\mathrm{diag}\,\{\dot{\mathbf{e}}\}^2)\left(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_d\right) + (\mathbf{z_b}^2\mathbf{p}_b + \boldsymbol{\Phi}\mathbf{z_a})\dot{\mathbf{e}} \\ &\quad + \boldsymbol{\Phi}\mathbf{z}_a^2(\mathbf{p}_a + \mathrm{diag}\,\{\mathbf{e}\}\,\mathrm{diag}\,\{\dot{\mathbf{e}}\})\mathbf{e}).\end{aligned} \tag{11}$$

Since $\mathbf{M}$ is a diagonal matrix, the following relationship can be obtained from (11),

$$\mathbf{M}\dot{\mathbf{s}} = (\mathbf{z_b} + 2\mathbf{z_b^2}\,\mathrm{diag}\,\{\dot{\mathbf{e}}\}^2)\boldsymbol{\tau} + \boldsymbol{\varphi}, \tag{12}$$

$$\text{where} \quad \boldsymbol{\varphi} = -\left(\mathbf{z_b} + 2\mathbf{z_b^2}\,\mathrm{diag}\,\{\dot{\mathbf{e}}\}^2\right)\left(\mathbf{d} + \mathbf{M}\ddot{\mathbf{q}}_d\right) + \mathbf{M}\left((\mathbf{z_b^2}\mathbf{p}_b + \boldsymbol{\Phi}\mathbf{z_a})\dot{\mathbf{e}} + \boldsymbol{\Phi}\mathbf{z}_a^2(\mathbf{p}_a + \mathrm{diag}\,\{\mathbf{e}\}\,\mathrm{diag}\,\{\dot{\mathbf{e}}\})\mathbf{e}\right). \tag{13}$$

is the overall uncertainty in the system.

Using the relationship (13) and Property 1, the upper bound structure of $\|\boldsymbol{\varphi}\|$ can be bounded as

$$\begin{aligned}\|\boldsymbol{\varphi}\| \leq & \underbrace{\boldsymbol{\Phi}\bar{m}}_{K_1}\underbrace{\|\mathbf{z_a}\|\|\xi\|}_{\chi_1} + \underbrace{2\bar{m}\|\mathbf{b}\|\|\dot{\mathbf{b}}\|}_{K_2}\underbrace{\|\mathbf{z_a}\|^2\|\xi\|}_{\chi_2} + \underbrace{2\bar{m}}_{K_3}\underbrace{\|\mathbf{z_a}\|^2\|\xi\|^3}_{\chi_3} \\ & + \underbrace{(\bar{d} + \bar{m}\|\ddot{\mathbf{q}}_d\|)}_{K_4}\underbrace{\|\mathbf{z_b}\|}_{\chi_4} + \underbrace{2\bar{m}\|\mathbf{b}_d\|\|\dot{\mathbf{b}}_d\|}_{K_5}\underbrace{\|\mathbf{z_b}\|^2\|\xi\|}_{\chi_5} \\ & + \underbrace{2(\bar{d} + \bar{m}\|\ddot{\mathbf{q}}_d\|)}_{K_6}\underbrace{\|\mathbf{z_b}\|^2\|\xi\|^2}_{\chi_6} \\ \leq & \sum_{j=1}^{6} K_j\chi_j, \end{aligned} \tag{14}$$

where $\xi = \begin{bmatrix}\mathbf{e}^T & \dot{\mathbf{e}}^T\end{bmatrix}^T$, K_j are unknown constants and χ_j are the regressor terms. The control law is designed as

$$\boldsymbol{\tau} = (\mathbf{z_b} + 2\mathbf{z_b^2}\,\mathrm{diag}\,\{\dot{\mathbf{e}}\}^2)^{-1}(-\boldsymbol{\Lambda}\mathbf{s} - \sum_{j=1}^{6}\hat{K}_j\chi_j(\mathbf{s}/\|\mathbf{s}\|)), \tag{15}$$

where $\hat{K}_j$ are the estimates of K_j and they are evaluated using the adaptive laws

$$\dot{\hat{K}}_j = \|\mathbf{s}\|\chi_j - \eta_j\hat{K}_j, \tag{16}$$

with η_j being user-defined positive design scalars.

Remark 4 (*State-dependent Uncertainty*). The inequality in (14) implies that state-dependencies occur inherently in the upper bounds of the uncertainties via ξ, z_a, and z_b. Therefore, conventional time-varying BLF based controllers (Gao et al., 2021a, 2021b; Sankaranarayanan, Yadav, et al., 2022; Wu et al., 2021) are not feasible as these uncertainties cannot be bounded a priori. On the other hand, the gains, $\hat{K}_j$, $i \in \{1, \ldots, 6\}$ in (15) are designed according to the structure of state-dependent uncertainties.

The control output from (15) is mapped to discretized ON-OFF control for the distributed thruster mechanism using the thruster selection logic as mentioned in Table 1. A detailed explanation of the thruster actuation is mentioned in the next subsection.

3.2. Thruster selection logic

As mentioned earlier, the slider is actuated by using its ON/OFF type distributed thruster mechanisms. Hence, the forces and torque provided from the control law presented in (15) need to be suitably converted into a set of thruster inputs based on the relationship between the Eqs. (2) and (4). Therefore, the following constrained optimization is performed:

$$\text{Min } J = T^T R_T T \tag{17}$$

$$\text{such that } \; WT = \boldsymbol{\tau}^{\upsilon} \tag{18}$$

$$T^b_{\max} \geq T_l \geq T^b_{\min} \tag{19}$$

where, $T = \left[T_1, \ldots, T_8\right]^T$ is the thrust vector, υ is the vector-mapping that converts a diagonal matrix into a column vector of appropriate dimensions, R_T represents a weighting parameter typically considered as unity matrix and W denotes the thrust allocation matrix given by

$$W(:,l) = \begin{bmatrix} \cos\beta_l \\ \sin\beta_l \\ \left(r^y_{T_l}\cos\beta_l - r^x_{T_1}\cos\beta_l\right) \end{bmatrix} \forall l = 1\ldots 8 \tag{20}$$

where β_l is the alignment of l_{th} thruster with respect to the X_B axis, chosen based on Table 1. Thus, (17)–(20) ensure that the commanded forces and torque are optimally converted to thruster actuation, while also enforcing the physical limitations of the bounded thrust magnitudes (T^b_{min}, T^b_{max}) as constraints. These continuous thrust commands are further converted into On-Off actuation using pulse-width-modulation (PWM) technique (Anthony, Wie, & Carroll, 1990; Ieko, Ochi, & Kanai, 1999). A similar multiple On-Off thruster-based control allocation framework can be found in Curti, Romano, and Bevilacqua (2010). A minimum ON-time constraint is additionally imposed in the thruster activation logic to account for the time constants of the physical relays responsible for the switching of the thruster states, as follows:

$$T_l = \begin{cases} 1, & \text{if } 0 \leq t \leq t^l_{ON} \text{ and } t^l_{ON} \geq \Delta t_c \\ 0, & \text{otherwise} \end{cases}$$

where, Δt_c denotes the minimum ON-time for effective activation. The thruster selection logic, combined with the PWM-based On-Off actuation, is also implemented and experimentally evaluated in Banerjee et al. (2022). Finally, the thruster output is sampled at discrete time-interval and fed as PWM signals to the actuators.

4. Stability analysis

Theorem 1. *Under Property 1 and (Assumption 1, 2), and using the control law (15) and the adaptive law (16), the trajectory of the closed-loop system (12) remains Uniformly Ultimately Bounded (UUB), and the error trajectory* $\mathbf{e}$ *and its time derivative* $\dot{\mathbf{e}}$ *remain within the bounds defined in ((6a), (6b)) for all* $t \geq 0$.

The solutions for the adaptive gains (16) can be derived to be,

$$\begin{aligned}\hat{K}_j &= \underbrace{exp(-\eta_j t)\hat{K}(0)}_{\geq 0} + \underbrace{\int_0^t -\eta_j(t-\vartheta)(\|\mathbf{s}\|\chi_j)d\vartheta}_{\geq 0} \\ \implies \hat{K}_j &\geq 0,\; j \in \{1, \ldots, 6\},\; \forall t \geq 0. \end{aligned} \tag{21}$$

Along with the relationship in (21), the closed-loop stability of the system is analyzed using the following Lyapunov function,

$$V = \frac{1}{2}\mathbf{s}^T\mathbf{M}\,\mathbf{s} + \frac{1}{2}\sum_{j=1}^{6}\left(\hat{K}_j - K_j\right)^2 \tag{22}$$

Using the system dynamics (1) and Property 1, the control law 3.1–(15), the time derivative of V can be expressed as

$$\begin{aligned}\dot{V} &= \mathbf{s}^T\mathbf{M}\dot{\mathbf{s}} + \sum_{j=1}^{6}\dot{\hat{K}}_j\left(\hat{K}_j - K_j\right) \\ &= \mathbf{s}^T\left(-\boldsymbol{\Lambda}\mathbf{s} - \sum_{j=1}^{7}\hat{K}_j\chi_j(\mathbf{s}/\|\mathbf{s}\|) + \boldsymbol{\varphi}\right) + \sum_{j=1}^{6}\dot{\hat{K}}_j\left(\hat{K}_j - K_j\right), \end{aligned} \tag{23}$$

$$\dot{V} \leq -\lambda_{\min}(\boldsymbol{\Lambda})\|\mathbf{s}\|^2 - \sum_{j=1}^{6}\left\{(\|\mathbf{s}\|\chi_j - \dot{\hat{K}}_j)(\hat{K}_j - K_j)\right\}. \tag{24}$$

From the adaptive law (16), (24) can be simplified as,

$$\begin{aligned}\dot{V} &\leq -\lambda_{\min}(\boldsymbol{\Lambda})\|\mathbf{s}\|^2 - \sum_{j=1}^{6}\eta_j\hat{K}_j(\hat{K}_j - K_j) \\ &\leq -\lambda_{\min}(\boldsymbol{\Lambda})\|\mathbf{s}\|^2 - \sum_{j=1}^{6}(\eta_j\hat{K}_j^2 - \eta_j\hat{K}_jK_j) \\ &\leq -\lambda_{\min}(\boldsymbol{\Lambda})\|\mathbf{s}\|^2 - \sum_{j=1}^{6}\frac{\eta_j}{2}\left(\left(\hat{K}_j - K_j\right)^2 - K^2\right). \end{aligned} \tag{25}$$

From (22), V can be upper bounded as,

$$V \leq \frac{\bar{m}}{2}\|s\|^2 + \sum_{j=1}^{6}\frac{1}{2}(\hat{K}_j - K_j)^2. \tag{26}$$

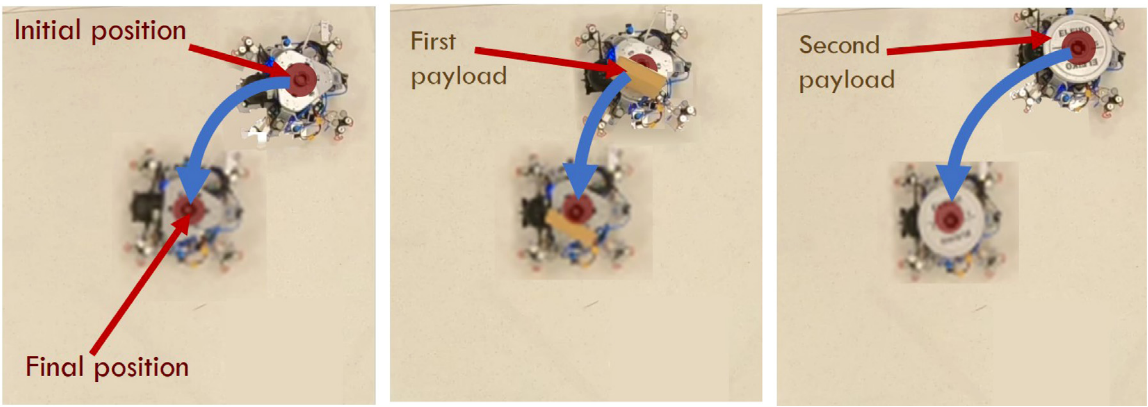


Fig. 2. A schematic of the robot with different payloads settling to the desired setpoint from their respective initial position.

Using (26), and defining a scalar κ as $0 < \kappa < \varrho$, $\dot{V}$ from (25), can be simplified to,

$$\dot{V} \leq -\varrho V + \sum_{j=1}^{6} \frac{\eta_j}{2} K_j^2 \tag{27}$$

$$= -\varrho V - (\varrho - \kappa)V + \sum_{j=1}^{6} \frac{\eta_j}{2} K_j^2, \tag{28}$$

where $\varrho \triangleq \frac{\min\{\lambda_{\min}(\boldsymbol{\Lambda}), \eta_j\}}{\max\{\bar{m}/2, 1/2\}}$ and κ is a scalar defined by $0 < \kappa < \varrho$. Defining a scalar, $\bar{\mathcal{B}} \triangleq \frac{\sum_{j=1}^{6} \eta_j K_j^2}{2(\varrho - \kappa)}$, it can be seen that $\dot{V} < -\kappa V(t)$ when $V > \bar{\mathcal{B}}$, so that

$$V \leq max\{V(0), \bar{\mathcal{B}}\}, \ \forall t > 0 \tag{29}$$

implying that the closed-loop system is UUB (cf. Khalil, 2002 for definition) implying that $\mathbf{s}$ and K_j remain bounded. This further implies that the tracking error trajectories e_i and their time-derivatives $\dot{e}_i$ with $i = \{x, y, \theta\}$ do not violate the state constraints b_i and b_{di} respectively for all $t \geq 0$.

Remark 5 (*Continuity in Control Law*)**.** To make the control law continuous in practice, the term $\mathbf{s}/\|\mathbf{s}\|$ is replaced by the continuous term $\mathrm{sat}(\mathbf{s}, \epsilon)$ for $\epsilon \in \mathbb{R}^+$ (cf. the notation definition in Section 1). This leads to a minor change in the stability analysis as presented in (cf. Roy, Roy, Lee, & Baldi, 2019) without changing the stability result. Therefore, it is not explicitly mentioned in this work.

Remark 6 (*Selection of the Control Parameters*)**.** It can be noted from (28) that a higher value of $\boldsymbol{\Lambda}$ improves the convergence of the tracking error. However, high values of $\boldsymbol{\Lambda}$ may lead to high control input demand (cf. (15)). On the other hand, the leakage η_j in (16) acts as a stabilizing component in the adaptive law by preventing $\hat{K}_j$ from becoming too high. Nevertheless, the choice of high-value of η_j compromises the rate of adaptation of $\hat{K}_j$ and hence, the transient performance. Therefore, the design parameters $\boldsymbol{\Lambda}, \eta_j$ need to be selected as per application requirements.

5. Experimental results

To demonstrate the performance of the proposed controller, it is tested on the Slider testbed in two different scenarios, which are explained in the following subsections. In the experimental setup, a Vicon camera based Motion capture system is used to obtain the pose and velocity of the Slider. The Slider is equipped with an onboard PC (Single-Board Computer Up-Board UP-CH01) running a Linux-based OS on a Quadcore Intel Atom x5-z8350 Processor and 4 GB DDR3L-1600 memory. The onboard computer receives the thruster control inputs from an offboard laptop PC and actuates the thruster relays using a microcontroller (Arduino). The controller runs on Mathworks Simulink software on the offboard laptop PC. The communication between the motion capture system, offboard PC, and the onboard computer is handled through ROS messages over WiFi. Further, different initial constraints are used for the scenarios to establish the purpose of having time-varying state constraints based on the use case. The performance of the controller is quantified using two metrics: Root Mean Squared Error (RMS) and Peak Error.

5.1. Position holding

In the first scenario, the slider is instructed to reach and hold the desired setpoint from an arbitrary initial pose (refer Fig. 2). The experiment is carried out with the unloaded slider and repeated with the two different payloads on the slider. The unloaded slider has a mass of approximately 4.528 kg. Its center of gravity is aligned with the origin of the $\mathbf{X_B} - \mathbf{Y_B}$ plane. The first payload is a rectangular box with a mass of approximately 0.7 kg. The payload is placed away from the $\mathbf{Z_B}$ axis to create an asymmetry in the mass distribution. The second payload is a 5 kg disk attached to the slider in alignment with the $\mathbf{Z_B}$ axis. The payloads significantly modify the dynamics from the unloaded dynamics and each other.

The parameters chosen for this scenario are given by: $\rho_{0x} = \rho_{0y} = 0.7$ m; $\rho_{ssx} = \rho_{ssy} = 0.07$ m ; $\rho_{0\theta} = 1.2$ rad (57.29 deg); $\rho_{ss\theta} = 0.4$ rad (22.91 deg); $\rho_{0dx} = \rho_{0dy} = 0.5$ m/s; $\rho_{ssdx} = \rho_{ssdy} = 0.15$ m/s; $\rho_{0d\theta} = 0.8$ rad/s (45.84 deg/s); $\rho_{ssd\theta} = 0.25$ rad/s (14.32 deg/s); $\boldsymbol{\Phi} = \mathrm{diag}\{1.2, 1.2, 1.5\}$; $\boldsymbol{\Lambda} = \mathrm{diag}\{2.0, 2.0, 0.3\}$; $\alpha_x = \alpha_y = \alpha_\theta = \alpha_{dx} = \alpha_{dy} = \alpha_{d\theta} = 0.1, K_j(0) = 0.1$; $\eta_j = 0.1 \ \forall j \in \{1-6\}$. The steady-state constraints are chosen to keep in mind that the robot has to be within a sufficient bound to collect space debris or object and carry it along a predefined path in a constrained environment. The constraints on orientation error is critical to properly orient the manipulator to carry out operations and to have a visual lock on a target for localization and safe navigation. The transition rate ($\alpha_i, \alpha_{di}, \ i \in \{x, y, \theta\}$) rates are chosen higher in this scenario to demonstrate faster convergence.

Fig. 3 shows the position and orientation trajectory taken by the slider to reach the desired pose from an arbitrary pose. The initial position errors are in the range of (0.323 m–0.53 m) in the X and Y coordinates for different payloads. It can be noted from the error profile shown in Fig. 4 that the robot reaches within the steady-state bounds almost at the same time and gradually converges to its final position irrespective of the variation in its dynamics. Further, the controller bounds all the state errors even when the initial velocities push the robot away from the setpoint, as seen in Fig. 5. The effect of variation in CoG is evident in the orientation trajectory and the corresponding errors (red line in the bottom subplots of the Figs. 3–6). The controller adapts reasonably fast to compensate for the asymmetry and correct the

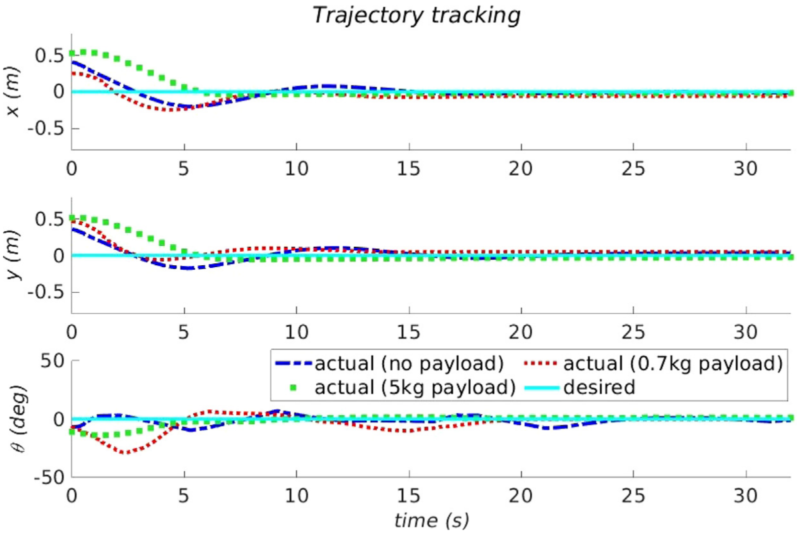


Fig. 3. The trajectory taken by the robot to move from its initial position to the desired point in the position-holding scenario using the proposed controller. The experiment is carried out on the robot without any payload, with 0.7 kg payload and 5 kg payload.

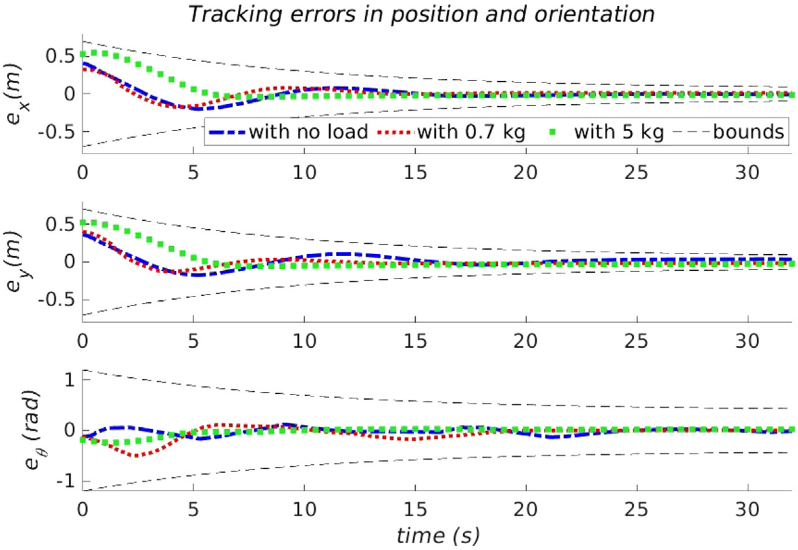


Fig. 4. Errors in position and orientation in the position holding scenario while reaching a desired pose from an arbitrary pose using the proposed controller.

trajectory. Similarly, the trajectory is damped heavily when the robot carries a payload of more than 100% of the original mass. However, the controller gives a higher input consistently to move the robot to the desired position. The adaptation and variation in the control inputs from the proposed controller while using different payloads are displayed in Fig. 6 for reference.

The quantitative analysis of the controller with and without the payloads are tabulated in Table 2. The RMS values of the errors indicate that the performance of the controller does not deviate much from the no-load condition. However, the effect of asymmetry in the torque and the effect of heavy payload on the forces are notable from the table. The peak error is calculated after 15 s just to show that once the robot settles in near the setpoint, it does not violate much irrespective of its configuration. These metrics provide an intuition of how the different states are affected by the types of payloads.

Table 2
Position holding performance of the proposed controller.

Controller	Root mean squared error					
	x (m)	y (m)	θ (rad)	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{\theta}$ (rad/s)
No load	0.097	0.090	0.061	0.057	0.051	0.059
0 .7 kg	0.083	0.080	0.147	0.036	0.061	0.083
5.0 kg	0.142	0.132	0.068	0.037	0.044	0.028
	Peak absolute error					
	x (m)	y (m)	θ (rad)	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{\theta}$ (rad/s)
No load	0.026	0.039	0.133	0.038	0.041	0.153
0.7 kg	0.016	0.019	0.175	0.014	0.012	0.121
5.0 kg	0.021	0.038	0.031	0.012	0.012	0.088

5.2. Trajectory tracking

In the second scenario, the robot is commanded to follow a circular trajectory while carrying the payloads. The sequence of operation is explained below (cf. Fig. 7):

- From its initial state, the robot converges to a predefined circular trajectory.
- At approximately 40 s, the rectangular payload is added on the robot away from its Z-axis while it is following the trajectory.
- At approximately 87 s, the first payload is removed from the robot. The robot continues the trajectory without any payload.

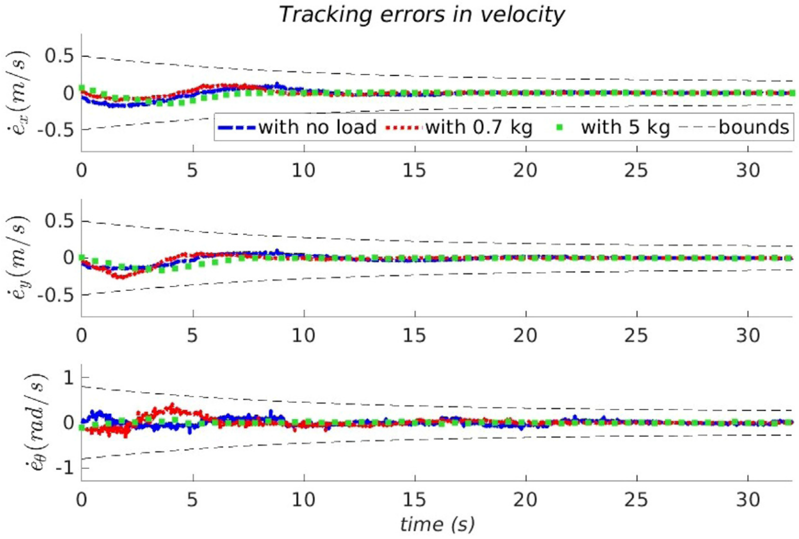


Fig. 5. Errors in velocities in the position holding scenario while reaching a desired pose from an initial state using the proposed controller.

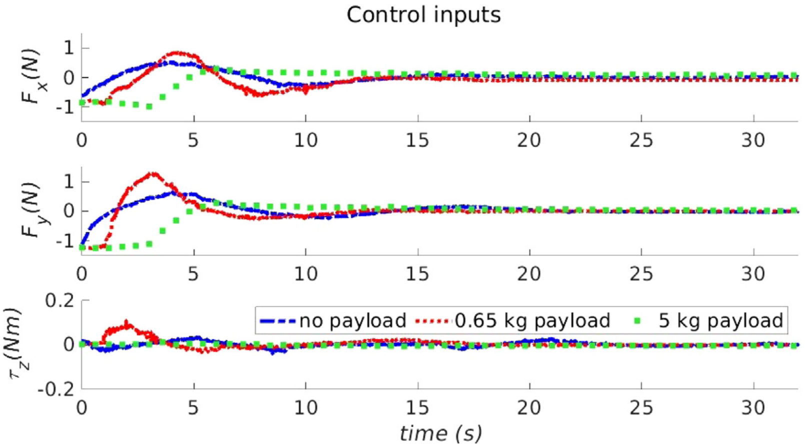


Fig. 6. Control inputs provided by the proposed controller in body-frame while using different payloads in the position holding scenario.

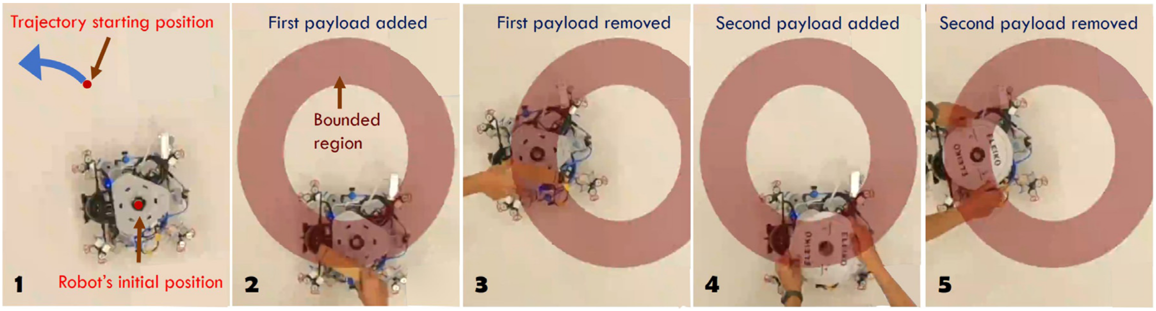


Fig. 7. The sequence of operations performed by the slider. (1) From its initial position, the robot moves to the trajectory; (2) First, a rectangular payload is added to the slider at around 40 s; (3) Rectangular payload is removed at around 87 s; (4) Second payload of 5 kg is added at around 102 s; (5) Second payload is removed at 150 s.

- At a time of approximately 102 s, the second payload of 5 kg mass is added to the robot.
- At a time of 150 s, the second payload is removed from the robot.
- The robot completes the rest of the trajectory without any load.

The steady-state error bounds and gains are chosen to be the same as the first scenario. However, to demonstrate the adaptability of the controller with respect to variation in initial conditions, the initial error bounds in the position and linear velocity are chosen as $\rho_x = \rho_y = 0.7$ m, $\rho_{dx} = \rho_{dy} = 0.5$ ms^{-1}. For orientation and angular velocity, $\rho_\theta = 1.2$ rad and $\rho_{d\theta} = 0.8$ rad/s. Similarly, the transition rates are modified to $\alpha_x = \alpha_y = \alpha_\theta = \alpha_{dx} = \alpha_{dy} = \alpha_{d\theta} = 0.05$. All other parameters are chosen to be the same as in the previous scenario. To the best of the authors' knowledge, the proposed controller is the first one to ensure prescribed time-varying state constraints while tackling state-dependent uncertainties without any a priori knowledge of the dynamic

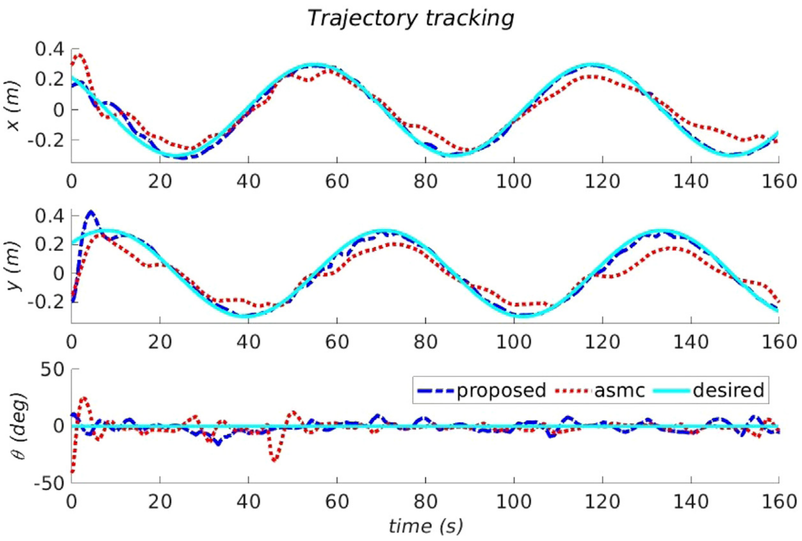


Fig. 8. Trajectory tracking performance.

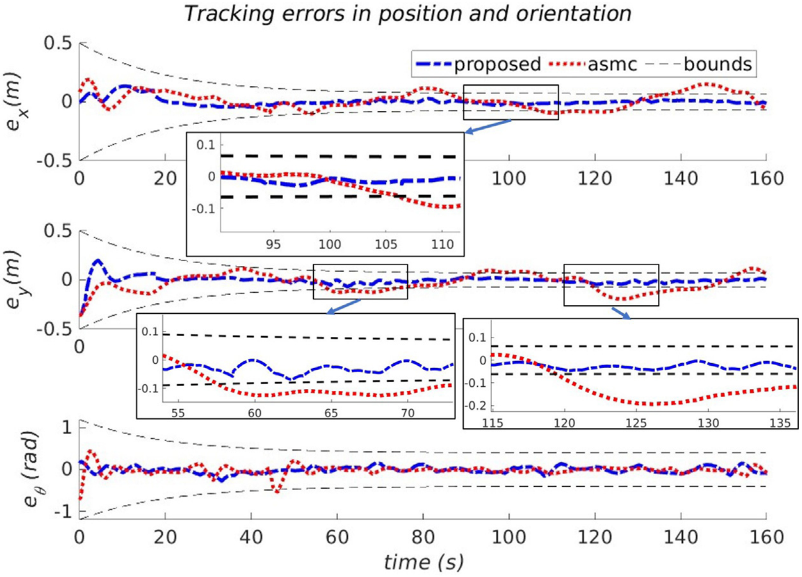


Fig. 9. Tracking error in position and orientation. The smaller snippets are added to show a magnified image of the regions where the trajectories are close to the bounds.

structure or the bounds of the uncertainties. Therefore, to highlight its advantages in terms of state constraints, the controller's performance is compared with that of a conventional adaptive robust controller (Adaptive Sliding Mode Controller (ASMC) of the form (Sankaranarayanan et al., 2020)). The parameters of the ASMC, $\boldsymbol{\Phi}, \boldsymbol{\Lambda}$ are chosen to be equal to the respective parameters of the proposed controller, and the gains of the adaptive parts are chosen as $\hat{K}_n(0) = 0.1$, $\alpha_n = 0.1$, $\forall n \in \{1, 2, 3\}$ for the best possible efforts. Since the controllers with constant bounds are hazardous when the initial conditions are outside the bounds, they are not used for comparison. Similarly, the non-adaptive controllers do not tackle state-dependent uncertainties. Hence, the error trajectories approach the bounds in case of rapid variations in the dynamics (while loading and unloading the payloads), the tracking error may approach the bounds causing infeasible control inputs and potential hazards. Hence, these controllers are not used for comparison.

Figs. 8–11 visualize the tracking performance of the proposed controller compared to the conventional adaptive controller (ASMC). From Fig. 8 it can be observed that the proposed controller follows closely to the desired trajectory in all the Degrees of Freedom, while the adaptive controller deviates at many regions. The convergence of the proposed controller's error trajectories within the bounds close to zero is notable in Figs. 9 and 10, while ASMC's error trajectories violate the bounds at multiple instants. Fig. 11 shows the desired path and the paths taken by the robot with the proposed controller and the ASMC controller along the XY-axis. The blue line shows the intervals in which the robot was carrying no load, the green line shows the interval with the first payload, and the red line shows the interval with the second payload. Since the initial errors are high, ASMC takes a long time to converge close to the desired path in the beginning. However, when the first payload is added, the robot deviates heavily while trying to compensate for the sudden change in the orientation (as observed in Figs. 8 and 9) caused by the variation in the moment of inertia. Though ASMC adapts for the second payload, its inputs are not sufficient enough to move and follow the trajectory effectively. Hence, it takes a trajectory that does not match up with the desired trajectory. However, in both cases, the proposed controller guarantees the prescribed performance without

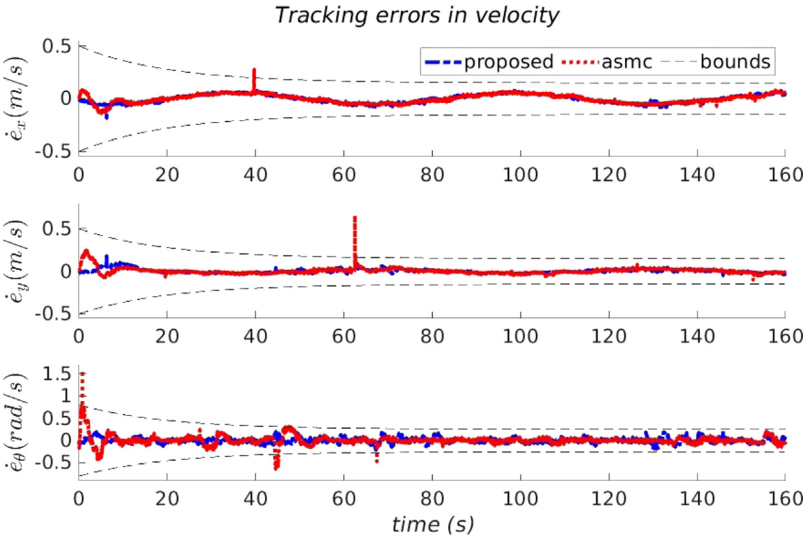


Fig. 10. Tracking error in linear and angular velocities.

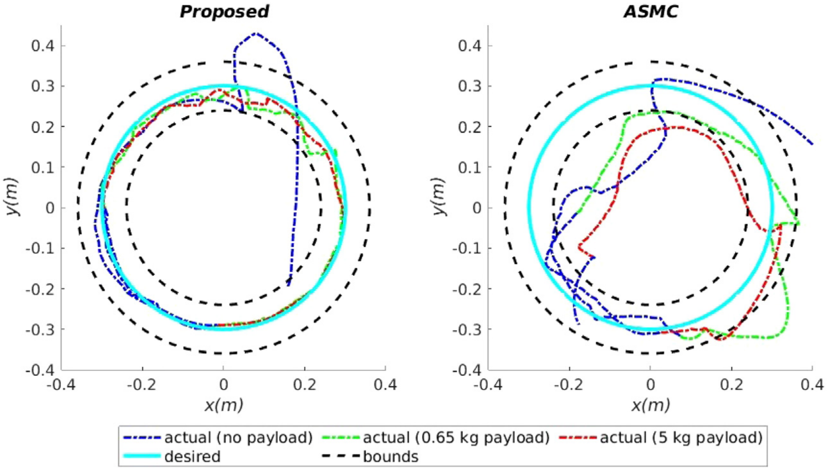


Fig. 11. Trajectory taken by the robot using the respective controllers. The bounding (black dotted circles) circles show the steady-state position error bounds. It is inferred that the proposed controller limits the error within the bound irrespective of the payload.

any violation. Fig. 12 shows that the level of control inputs from the proposed controller adapts quickly to match up for the variation in the dynamics. This control signal is passed through the thruster-selection block. Since the thrusters use an ON-OFF controller technique, the resultant control inputs are at specific discrete levels (±1.4 N, ±0.7 N or 0 N). Similarly, the resultant torques will also be in specific levels ((±0.196 Nm, ±0.098 Nm or 0 Nm)). Even after discretization, the controller still guarantees the performance. However, though ASMC adapts to different levels based on uncertainties, the adaptation is not sufficient to compensate for the deviations.

The numerical results of the second scenario are tabulated in Table 3. For a fair comparison between both controllers, the peak error considers the error values after 15 s. The RMS values show that the proposed controller forces the error trajectories to settle well below the steady-state bounds, while there is no such guarantee while using ASMC. Hence, the errors deviate significantly, and hence, the RMS of position errors are greater than the steady-state bounds. Further, the peak errors prove that the proposed controller maintains the error trajectories within the bounds, while ASMC violates the error bounds by high margins (highlighted with bold letters in the respective table). The improvement in the tracking performance with respect to ASMC is reflected in the percentage improvement metric denoted in Table 3. For the proposed controller, the fuel consumed is measured in terms of the total change in velocity required for the complete maneuver, denoted by ΔV (cf. Satpute, Bodin, & Nikolakopoulos, 2021), which is calculated to be 4.65 ms^{-1}. The experimental results are recorded and can be viewed for reference here (Sankaranarayanan, Avijit, Sumeet, & George, 2022).

Table 3
Trajectory Tracking Performance.

Controller	Root mean squared error					
	x (m)	y (m)	θ (rad)	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{\theta}$ (rad/s)
Proposed	0.033	0.044	0.078	0.037	0.034	0.091
ASMC	0.073	0.095	0.104	0.037	0.036	0.101
% Improvement	54.79	53.68	25	0	5.55	9.99
	Peak absolute error					
	x (m)	y (m)	θ (rad)	$\dot{x}$ (m/s)	$\dot{y}$ (m/s)	$\dot{\theta}$ (rad/s)
Proposed	0.048	0.069	0.302	0.089	0.071	0.222
ASMC	**0.151**	**0.194**	**0.532**	0.099	0.137	**0.659**

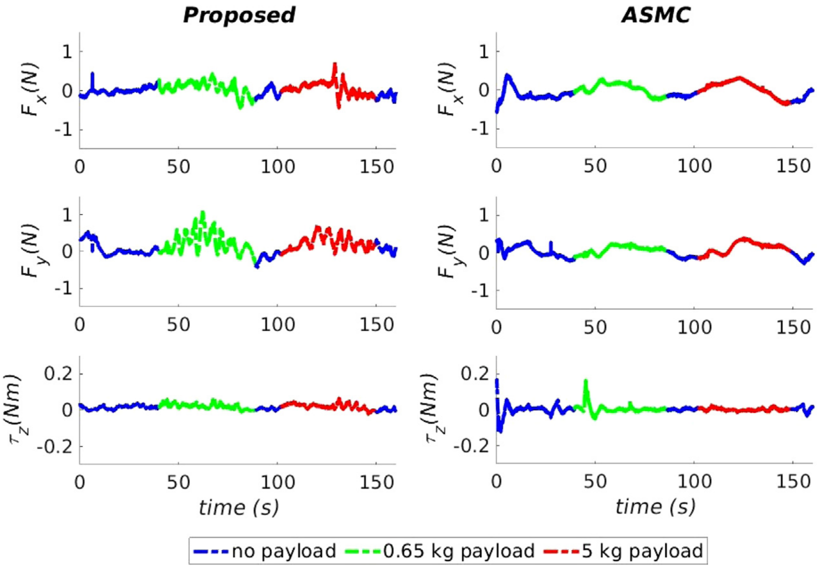


Fig. 12. The sequence of control inputs in body-frame on the slider as a resultant of the thruster outputs. The input at each instant is discrete and at a specified level. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6. Conclusion

In this paper, a BLF-based adaptive controller is proposed for a spacecraft to track a desired trajectory under time-varying state constraints while carrying different types of payloads. The proposed controller ensures that the constraints on the states and their time derivatives are guaranteed throughout the maneuver. Further, the controller does not require any a priori knowledge of the state-dependent uncertainties and their upper bounds. The closed-loop stability analysis of the system is performed using the Lyapunov method. Experimental results using a ground-based zero-gravity floating platform (Slider) under two scenarios have demonstrated significant performance improvement for the proposed controller against the conventional adaptive controller.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Al Issa, S., & Kar, I. (2021). Design and implementation of event-triggered adaptive controller for commercial mobile robots subject to input delays and limited communications. *Control Engineering Practice*, *114*, Article 104865.

Anthony, T. C., Wie, B., & Carroll, S. (1990). Pulse-modulated control synthesis for a flexible spacecraft. *Journal of Guidance, Control, and Dynamics*, *13*(6), 1014–1022.

Banerjee, A., Haluska, J., Satpute, S. G., Kominiak, D., & Nikolakopoulos, G. (2021). Slider: On the design and modeling of a 2D floating satellite platform. http://dx.doi.org/10.48550/arxiv.2101.06335, URL: https://arxiv.org/abs/2101.06335v1.

Banerjee, A., Satpute, S. G., Kanellakis, C., Tevetzidis, I., Haluska, J., Bodin, P., et al. (2022). On the design, modeling and experimental verification of a floating satellite platform. *IEEE Robotics and Automation Letters*, *7*(2), 1364–1371.

Chu, Z., Ma, Y., & Cui, J. (2018). Adaptive reactionless control strategy via the PSO-ELM algorithm for free-floating space robots during manipulation of unknown objects. *Nonlinear Dynamics*, *91*(2), 1321–1335.

Correia, R., & Ventura, R. (2021). Payload transportation in microgravity with single and multiple cooperative free-flyer robots. In *2021 IEEE international conference on autonomous robot systems and competitions* (pp. 173–178). IEEE.

Curti, F., Romano, M., & Bevilacqua, R. (2010). Lyapunov-based thrusters' selection for spacecraft control: analysis and experimentation. *Journal of Guidance, Control, and Dynamics*, *33*(4), 1143–1160.

Daniel, N., Michel, D., Sabine, M., Raffaele, M., & Javier, R. (2015). H2020 PERASPERA, Master Plan of SRC activities. *Science China Technological Sciences*, *D3.4*(1.0), 1–42.

Du, H., Zhang, J., Wu, D., Zhu, W., Li, H., & Chu, Z. (2020). Fixed-time attitude stabilization for a rigid spacecraft. *ISA Transactions*, *98*, 263–270. http://dx.doi.org/10.1016/J.ISATRA.2019.08.026.

Gao, S., Liu, X., Jing, Y., & Dimirovski, G. M. (2021a). Finite-time prescribed performance control for spacecraft attitude tracking. *IEEE/ASME Transactions on Mechatronics*.

Gao, S., Liu, X., Jing, Y., & Dimirovski, G. M. (2021b). A novel finite-time prescribed performance control scheme for spacecraft attitude tracking. *Aerospace Science and Technology*, *118*, Article 107044.

Huang, X., Yan, Y., & Huang, Z. (2017). Finite-time control of underactuated spacecraft hovering. *Control Engineering Practice*, *68*, 46–62.

Ieko, T., Ochi, Y., & Kanai, K. (1999). New design method for pulse-width modulation control systems via digital redesign. *Journal of Guidance, Control, and Dynamics*, *22*(1), 123–128.

Kai, T. (2018). Robust control of a 3D space robot with an initial angular momentum based on the nonlinear model predictive control method. *International Journal of Advanced Computer Science and Applications*, *9*(6), 7–15.

Khalil, H. K. (2002). *Nonlinear systems, vol. 3*. NJ: Prentice hall Upper Saddle River.

Kraïem, S., Rognant, M., Biannic, J. M., & Brière, Y. (2021). Robust control of rotation-floating space robots with flexible appendages for on-orbit servicing. *IFAC-PapersOnLine*, *54*(20), 134–140.

Malekzadeh, M., & Sadeghian, H. (2019). Attitude control of spacecraft simulator without angular velocity measurement. *Control Engineering Practice*, *84*, 72–81.

Roy, S., Baldi, S., & Fridman, L. M. (2020). On adaptive sliding mode control without a priori bounded uncertainty. *Automatica*, *111*, Article 108650.

Roy, S., Roy, S. B., Lee, J., & Baldi, S. (2019). Overcoming the underestimation and overestimation problems in adaptive sliding mode control. *IEEE/ASME Transactions on Mechatronics*, *24*(5), 2031–2039.

Sachan, K., & Padhi, R. (2020). Nonlinear robust neuro-adaptive flight control for hypersonic vehicles with state constraints. *Control Engineering Practice*, *102*, Article 104526.

Sankaranarayanan, V. N., Avijit, B., Sumeet, G., & George, N. (2022). Adaptive control for payload carrying spacecraft with state constraints - Luleå University of Technology. Retrieved from https://youtu.be/v5b94caNKAs.

Sankaranarayanan, V. N., Roy, S., & Baldi, S. (2020). Aerial transportation of unknown payloads: Adaptive path tracking for quadrotors. In *2020 IEEE/RSJ international conference on intelligent robots and systems* (pp. 7710–7715). IEEE.

Sankaranarayanan, V., Yadav, R., Swayampakula, R., Ganguly, S., & Roy, S. (2022). Robustifying payload carrying operations for quadrotors under time-varying state constraints and uncertainty. *IEEE Robotics and Automation Letters*, *7*, http://dx.doi.org/10.1109/LRA.2022.3154051.

Satpute, S. G., Bodin, P., & Nikolakopoulos, G. (2021). Cooperative planning for multi-site asteroid visual coverage. *Advanced Robotics*, *35*(21–22), 1332–1346.

Shahna, M. H., & Abedi, M. (2021). Design of a finite time passivity based adaptive sliding mode control implementing on a spacecraft attitude dynamic simulator. *Control Engineering Practice*, *114*, Article 104866.

Shi, L., Yao, H., Shan, M., Gao, Q., & Jin, X. (2022). Robust control of a space robot based on an optimized adaptive variable structure control method. *Aerospace Science and Technology*, *120*, Article 107267.

Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2008). *Robot dynamics and control*. John Wiley & Sons.

Wang, X., Xu, B., Cheng, Y., Wang, H., & Sun, F. (2022). Robust adaptive learning control of space robot for target capturing using neural network. *IEEE Transactions on Neural Networks and Learning Systems*.

Wu, Y. Y., Zhang, Y., & Wu, A. G. (2021). Preassigned finite-time attitude control for spacecraft based on time-varying barrier Lyapunov functions. *Aerospace Science and Technology*, *108*, Article 106331. http://dx.doi.org/10.1016/J.AST.2020.106331.

Yao, Q. (2021). Adaptive fuzzy neural network control for a space manipulator in the presence of output constraints and input nonlinearities. *Advances in Space Research*, *67*(6), 1830–1843.

Zhang, X., Liu, J., Gao, Q., & Ju, Z. (2020). Adaptive robust decoupling control of multi-arm space robots using time-delay estimation technique. *Nonlinear Dynamics*, *100*(3), 2449–2467.

Zhang, S., & Tang, Z. (2018). Adaptive neural control for robotic manipulators under constrained task space. In *International conference on social robotics* (pp. 599–608). Springer.