

# Attitude Takeover Control for Noncooperative Space Targets Based on Gaussian Processes with Online Model Learning

**Citation for published version (APA):**

Liu, Y., Wang, P., Lee, C.-H., & Tóth, R. (2023). *Attitude Takeover Control for Noncooperative Space Targets Based on Gaussian Processes with Online Model Learning*. arXiv.org. <https://doi.org/10.48550/arXiv.2310.15883>

**Document license:**

CC BY-NC-SA

**DOI:**

[10.48550/arXiv.2310.15883](https://doi.org/10.48550/arXiv.2310.15883)

**Document status and date:**

Published: 24/10/2023

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Attitude Takeover Control for Noncooperative Space Targets Based on Gaussian Processes with Online Model Learning

YUHAN LIU

Eindhoven University of Technology, Eindhoven, The Netherlands

PENGYU WANG

Korea Advanced Institute of Science and Technology, Daejeon, Korea

CHANG-HUN LEE

Korea Advanced Institute of Science and Technology, Daejeon, Korea

ROLAND TÓTH

Eindhoven University of Technology, Eindhoven, The Netherlands  
Institute for Computer Science and Control, Budapest, Hungary

**Abstract**—One major challenge for autonomous attitude takeover control for on-orbit servicing of spacecraft is that an accurate dynamic motion model of the combined vehicles is highly nonlinear, complex and often costly to identify online, which makes traditional model-based control impractical for this task. To address this issue, a recursive online sparse Gaussian Process (GP)-based learning strategy for attitude takeover control of noncooperative targets with maneuverability is proposed, where the unknown dynamics are online compensated based on the learnt GP model in a semi-feedforward manner. The method enables the continuous use of on-orbit data to successively improve the learnt model during online operation and has reduced computational load compared to standard GP regression. Next to the GP-based feedforward, a feedback controller is proposed that varies its gains based on the predicted model confidence, ensuring robustness of the overall scheme. Moreover, rigorous theoretical proofs of Lyapunov stability and boundedness guarantees of the proposed method-driven closed-loop system are provided in the probabilistic sense. A simulation study based on a high-fidelity simulator is used to show the effectiveness of the proposed strategy and demonstrate its high performance.

**Index Terms**—Attitude takeover control, Gaussian process, machine learning, noncooperative space target, on-orbit servicing.

This research has received funding from the European Union’s EDF programme under grant agreement No 101103386 and has been supported by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002).  
(Corresponding author: Pengyu Wang)

Yuhan Liu is with the Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands. (Email: [y.liu1@tue.nl](mailto:y.liu1@tue.nl)); Pengyu Wang and Chang-hun Lee are with the Flight Dynamics and Control Laboratory, Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea. (Email: [wangpy@kaist.ac.kr](mailto:wangpy@kaist.ac.kr), [lckdngns@kaist.ac.kr](mailto:lckdngns@kaist.ac.kr)); Roland Tóth is with the Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands and the Systems and Control Laboratory, Institute for Computer Science and Control, Budapest, Hungary. (Email: [r.toth@tue.nl](mailto:r.toth@tue.nl)).

## I. INTRODUCTION

IN recent years, there has been a rapid development in *on-orbit servicing* (OOS) applications such as on-orbit refueling, on-orbit maintenance, on-orbit assembly, orbit transfer, and active space debris removal [1]. An entire OOS mission consists of the following four distinct phases: long-range guidance, final approaching, on-orbit capture, and post-capture. In the post-capture phase, a combined spacecraft is formed through the connection of the servicing spacecraft (referred to as the “servicer”) and the target using robot manipulators or tethers. The servicer is supposed to “*take over*” the attitude control of the target, such that the control torque for the combined spacecraft is completely provided by the servicer. This autonomous attitude takeover control for the combined spacecraft plays a key role in the subsequent tasks (such as refueling and debris removal) and has become an important component to ensure the success of OOS missions.

Capture and post-capture control for cooperative targets have significantly matured and have been applied in some executed OOS missions [2], [3]. However, for other OOS missions, such as on-orbit maintenance and debris removal, the targets are usually noncooperative and the mission needs to be conducted under the following specs: 1) sufficient knowledge of the structure of the target, mass properties, and state of motion is not *a priori* available; 2) no communication link can be established to send messages between the servicer and target; 3) no pre-designed capture interface is present on the target. With the increasing diversity of OOS missions, the targets can also display partial failure characteristics, i.e., still have weak attitude controllability despite the failure of actuators. Hence, for upcoming OOS missions, the attitude control for the post-capture combined spacecraft is likely to be more challenging. First, noncooperative characteristics of the target in terms of 2) necessitate robustness and disturbance rejection capabilities of the servicer. Furthermore, in view of 1) and 3), no accurate model can be assumed to be available for the dynamics of the combined spacecraft. Hence, traditional model-based control methods can suffer from the unknown uncertainties and unmodeled dynamics that can be encountered during such missions, and can result in significant loss of performance or even stability. Additionally, it is not feasible to effectively identify the mass properties of the combined spacecraft in real-time due to the external unmeasured input caused by the attitude maneuverability of the target.

In scientific literature, already promising studies have been obtained for post-capture attitude takeover control. The pioneering work in this field has mainly concentrated on designing model-based controllers on the basis of an accurately identified model. For this purpose, Bergmann *et al.* in [4] have proposed an online inertia identification algorithm, where the mass properties of a rigid spacecraft are estimated by the analytic solution of the motion equation under free rotation. In [5], Murotsu *et al.* have developed a parameter identification method based on the conservation

of momentum. Ma *et al.* in [6] have further extended this work to scenarios under unknown spacecraft systems by a two-step identification method. Christidi-Loumpasefski *et al.* in [7], [8] have proposed momentum-conservation-based methods to fully identify the parameters of free-flying system dynamics with unmeasurable sloshing states. In recent years, visual CCD cameras [9] and deep learning [10] have also been proposed for estimation of inertia parameters of the combined spacecraft. However, these methods are not applicable to scenarios where the target still has attitude maneuverability.

In contrast to parameter identification and model-based control approaches, an effective alternative approach to deal with the unknown dynamics of the combined spacecraft is adaptive control. In [11], a backstepping-based robust adaptive controller has been proposed for attitude stabilization of a tumbling tethered combined spacecraft. By involving an improved adaptive sliding mode controller to reduce the total angular momentum of the system, Zhang *et al.* in [12] have presented a coordinated control approach for the combined spacecraft without precise inertia information. Kang *et al.* in [13] have considered the situation of non-cooperative body attachment, and designed an adaptive control strategy for rapid stabilization with high precision in different scenarios. A hybrid controller has been derived in [14] for a flexible combined spacecraft in the presence of model uncertainties, input constraints, external disturbances, and actuator faults. Moreover, to further improve the adaptability of the controller, *neural network* (NN) and fuzzy logic-based control approaches have attracted great attention recently (see, e.g., [15]–[17]). However, it is usually essential for adaptive control approaches to make assumptions on the existence of upper bounds on the model uncertainty, external disturbances, etc. to ensure the stability of the closed-loop system, which can be conservative for an attitude takeover task. On the other hand, parametric modeling methods such as NN and fuzzy logic inherently have the disadvantages of being complex and fragile in terms of their generalization capability beyond the training region, and can be sensitive in terms of the selected network structure, activation functions, hyperparameter tuning, initialization, and signal normalization.

Note that the above references involve system models to facilitate the controller design, i.e., model-based control. In contrast, there are some results of “model-free” control for the combined spacecraft. In [18], a model-free prescribed performance adaptive attitude controller has been proposed for the flexible combined spacecraft dynamics. In [19], a low-complexity model-free control strategy has been presented based on the prescribed performance technique. In [20], an inertia-free control approach has been derived for the combined spacecraft which facilitates “appointed-time” stability of the system. However, it should be mentioned that these model-free control strategies are implicitly dependent on the assumed inertia of the system through their hyperparameter choices. Hence, the parameters in these model-free control laws

need to be tuned based on model information and practical experience, which makes the “model freeness” of these methods questionable.

In view of all the aforementioned developed approaches and the identified challenges, the current shortcomings in the area of combined spacecraft attitude takeover control are summarized as:

(i) General identification-based control approaches are not applicable to the task scenario where the target still has attitude maneuverability (low frequent unknown excitation).

(ii) Performance of adaptive control methods is highly sensitive to hyperparameter choices. Inadequate selection may lead to degradation of the overall control performance and even loss of stability.

To address these issues, the emerging approaches of machine learning-based control offer better capabilities for capturing unmodeled dynamics and achieving superior performance over the existing methods. As one of the promising tools, GP [21] has been increasingly successful in the field of nonparametric modeling. It is a flexible function estimator that also provides a characterization of the uncertainty of the estimate in a computationally efficient manner compared to NNs and fuzzy logics. Powerful results have been achieved for GP-based learning control for robot arms [22], quadrotors [23], race cars [24], etc. However, there are still some technical barriers to the design of GP-based learning control for the attitude takeover problem: (a) The standard GP is not suitable for large training data sets, which will lead to high computational load for the onboard computer. (b) Most GP-based learning control methods do not perform online updating which is needed to address the time-varying disturbances during the system operation. To solve this issue, various online GP approaches have been developed to update the GP model during control operation. A sparse online GP (SOGP) method has been first proposed in [25], which efficiently approximates the Kullback-Leibler divergence, i.e., the distance, between the current GP model and the new data pair and updates the GP estimate based on this distance. This work has been further extended and applied in model reference adaptive control [26], [27], incremental backstepping control [28], etc. Similarly, another online GP approach is evolving GP [29], [30], which updates the training data set online using various types of information criteria, and has been utilized in model predictive control [31]. However, both of the above methods still involve a “dictionary” update and re-computation of the Gram matrix at each time step, which is a computationally costly operation.

To overcome the aforementioned challenges, this paper proposes an innovative GP-based online learning strategy for post-capture attitude takeover control with unknown dynamics and attitude maneuverability of the target, as an extension of our previous work presented in [32]. The main contributions of this paper are as follows:

- 1) A novel GP-based learning control strategy for attitude takeover that is applicable even under attitude maneuverability of the target.
- 2) A novel recursive online sparse form of the GP estimator that facilitates efficient continuous learning of unknown time-varying uncertainties during operation. A crucial advantage of the approach is that no online re-tuning of the hyperparameters, nor update of the data-dictionary is required at every time-moment, which ensures low online computational cost.
- 3) Proven stability guarantee of the closed-loop operation with the proposed method, ensuring that the attitude orientation error remains ultimately bounded around the origin with high probability.
- 4) Verification of the proposed method in a high-fidelity simulation study.

The remainder of this paper is organized as follows. Section II covers the problem formulation and control objectives. The proposed recursive online sparse GP regression algorithm is detailed in Section III. The GP-based adaptive learning control procedure and its rigorous stability analysis are presented in Section IV. Numerical simulation results are provided in Section V, followed by the conclusions in Section VI.

*Notation:*  $\mathbb{R}$ ,  $\mathbb{Z}$ ,  $\mathbb{N}$ , and  $\mathbb{Q}^3$  denote the sets of real numbers, integers, nonnegative integers, and unit quaternions,  $\mathbb{R}_0^+$  corresponds to nonnegative real numbers, while  $\mathbb{S}^{n \times n}$  is the set of real symmetric matrices of dimension  $n \times n$ . The 2-norm of a vector or a matrix is denoted as  $\|\cdot\|$ , while, for a given Hilbert space  $\mathcal{H}$ , the corresponding norm is denoted by  $\|\cdot\|_{\mathcal{H}}$ .  $\lambda_{\min}(\cdot)$  and  $\lambda_{\max}(\cdot)$  are the minimum and maximum eigenvalues of a matrix, respectively.  $\text{vec}(\mathbf{x}_1, \dots, \mathbf{x}_n) = [\mathbf{x}_1^\top \dots \mathbf{x}_n^\top]^\top$  denotes the column-wise composition of vectors. Additionally,  $\mathbf{I}_n$  is a  $n \times n$  identity matrix and the projection  $\cdot^\times : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$  gives a skew-symmetric matrix, ensuring that  $\mathbf{a}^\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$  for all  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$  where  $\times$  corresponds to the cross-product operator.  $\mathbb{I}_1^j = \{s \in \mathbb{Z} \mid i \leq s \leq j\}$  denotes an index set.

## II. Problem Formulation

### A. Uncertain System Model

After successful docking to the target, the combined spacecraft, as considered in this paper, contains three parts: the servicer, the target, and the manipulators. As shown in Fig. 1, the servicer can capture the interface ring on the target by using its two manipulators. To describe the motion dynamics of the combined spacecraft, first assume that it can be described as a single rigid body, which also means that the manipulator arms do not introduce additional dynamics. This simplified motion dynamics of the combined spacecraft are given by [33]

$$\dot{\mathbf{q}} = \frac{1}{2}(q_0 \mathbf{I}_3 + \mathbf{q}^\times) \boldsymbol{\omega}, \quad \dot{q}_0 = -\frac{1}{2} \mathbf{q}^\top \boldsymbol{\omega}, \quad (1)$$

where  $\mathbf{Q} = [q_0 \ \mathbf{q}^\top]^\top \in \mathbb{Q}^3$  denotes the unit quaternion describing the attitude of the spacecraft in terms of rotation

of the body frame  $\mathcal{F}_B$  w.r.t. the inertial frame  $\mathcal{F}_I$ , while  $\boldsymbol{\omega} \in \mathbb{R}^3$  is the angular velocity of the spacecraft expressed in  $\mathcal{F}_B$ .

In this paper, the problem of attitude orientation control is considered, i.e., the combined spacecraft is required to realize a commanded desired orientation. Let  $\mathbf{Q}_d = [q_{d0} \ \mathbf{q}_d^\top]^\top \in \mathbb{Q}^3$  denote the desired attitude, corresponding to a desired body frame  $\mathcal{F}_D$ , and  $\boldsymbol{\omega}_d \in \mathbb{R}^3$  the desired angular velocity, respectively. Thus, the attitude error  $\mathbf{Q}_e = [q_{e0} \ \mathbf{q}_e^\top]^\top$  of  $\mathcal{F}_B$  w.r.t.  $\mathcal{F}_D$  can be calculated by  $\mathbf{Q}_e = \mathbf{Q}_d^* \otimes \mathbf{Q}$ , where  $\mathbf{Q}_d^* = [q_{d0} \ -\mathbf{q}_d^\top]^\top$  denotes the conjugate of  $\mathbf{Q}_d$ , and the symbol “ $\otimes$ ” refers to the product operator for any two quaternions  $\mathbf{Q}_i = [q_{i0} \ \mathbf{q}_i^\top]^\top$  and  $\mathbf{Q}_j = [q_{j0} \ \mathbf{q}_j^\top]^\top$ :

$$\mathbf{Q}_i \otimes \mathbf{Q}_j = \begin{bmatrix} q_{i0}q_{j0} - \mathbf{q}_i^\top \mathbf{q}_j \\ q_{i0}\mathbf{q}_j + q_{j0}\mathbf{q}_i + \mathbf{q}_i^\times \mathbf{q}_j \end{bmatrix}.$$

The attitude error kinematics can be derived as:

$$\dot{\mathbf{q}}_e = \frac{1}{2}(q_{e0} \mathbf{I}_3 + \mathbf{q}_e^\times) \boldsymbol{\omega}_e, \quad \dot{q}_{e0} = -\frac{1}{2} \mathbf{q}_e^\top \boldsymbol{\omega}_e \quad (2)$$

where  $\boldsymbol{\omega}_e = \boldsymbol{\omega} - \mathbf{C}(\mathbf{Q}_e) \boldsymbol{\omega}_d$ , and the rotation matrix is  $\mathbf{C}(\mathbf{Q}_e) = \mathbf{I}_3 - 2q_{e0} \mathbf{q}_e^\times + 2\mathbf{q}_e^\times \mathbf{q}_e^\times$ . Because the problem of set point control of the attitude is studied in this paper, the desired angular velocity is set as  $\boldsymbol{\omega}_d = \mathbf{0}$ , giving  $\boldsymbol{\omega}_e = \boldsymbol{\omega}$ . Hence, the attitude error kinematics (2) can be written as:

$$\dot{\mathbf{q}}_e = \frac{1}{2}(q_{e0} \mathbf{I}_3 + \mathbf{q}_e^\times) \boldsymbol{\omega}, \quad \dot{q}_{e0} = -\frac{1}{2} \mathbf{q}_e^\top \boldsymbol{\omega}. \quad (3)$$

Furthermore, the attitude dynamics are given by:

$$\mathbf{J}_c \dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}^\times \mathbf{J}_c \boldsymbol{\omega} + \mathbf{u} + \boldsymbol{\tau}_d, \quad (4)$$

where  $\mathbf{J}_c \in \mathbb{R}^{3 \times 3}$  is the positive-definite symmetric inertia matrix,  $\mathbf{u} \in \mathbb{R}^3$  is the control torque, while  $\boldsymbol{\tau}_d \in \mathbb{R}^3$  is an external torque expressed in  $\mathcal{F}_B$  representing time-varying disturbances. Note that  $\boldsymbol{\tau}_d$  can be seen as a collection of additional unconsidered dynamics such as manipulator arm dynamics, rotating solar panels, solar pressure, gravity gradient, etc. Obviously, if  $\boldsymbol{\tau}_d = \mathbf{0}$  and  $\mathbf{J}_c$  is accurately known or identified, global asymptotic stability can be easily guaranteed by model-based control, ensuring that the closed-loop trajectory  $(\mathbf{Q}_e, \boldsymbol{\omega})$  converges to the stable equilibrium  $(\mathbf{1}, \mathbf{0})$ .

However, the true value of  $\mathbf{J}_c$  is hard to determine accurately by online parameter identification when the servicer is executing an OOS mission, particularly under maneuvering capability of the target. Similarly,  $\boldsymbol{\tau}_d$  can be partly reconstructed by space environment models, but some residual uncertainties always remain.

To be able to express some known baseline dynamics w.r.t. the real dynamics of the spacecraft, introduce  $\mathbf{J}_c = \mathbf{J}_{c0} + \tilde{\mathbf{J}}_c$ , where  $\mathbf{J}_{c0} \in \mathbb{R}^{3 \times 3}$  represents the nonsingular symmetric nominal inertia matrix of the combined spacecraft, and  $\tilde{\mathbf{J}}_c \in \mathbb{R}^{3 \times 3}$  denotes the inertia deviation resulting from the capture and non-nominal characteristics of the target. The inverse of  $\mathbf{J}_c$  can be computed as:

$$\mathbf{J}_c^{-1} = \mathbf{J}_{c0}^{-1} + \tilde{\mathbf{J}}_c^* \quad (5)$$

where  $\tilde{\mathbf{J}}_c^* = -(\mathbf{I}_3 + \mathbf{J}_{c0}^{-1} \tilde{\mathbf{J}}_c)^{-1} \mathbf{J}_{c0}^{-1} \tilde{\mathbf{J}}_c \mathbf{J}_{c0}^{-1}$ .

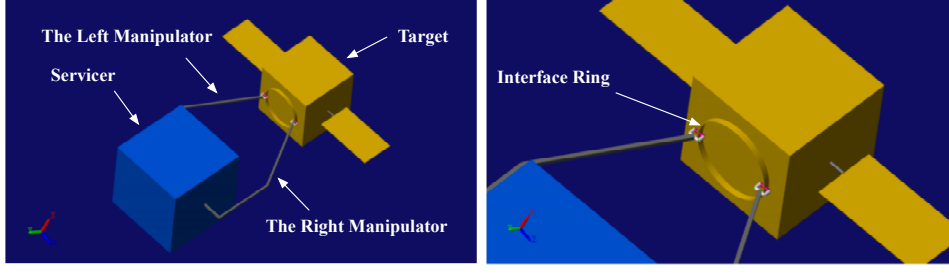


Fig. 1: A snapshot of the combined spacecraft

Thus, the attitude error dynamics of combined spacecraft with uncertainties can be formulated as

$$\begin{cases} \dot{\mathbf{q}}_e &= \frac{1}{2}(\mathbf{q}_{e0}\mathbf{I}_3 + \mathbf{q}_e^\times)\boldsymbol{\omega} \\ \dot{\boldsymbol{\omega}} &= -\mathbf{J}_{c0}^{-1}\boldsymbol{\omega}^\times\mathbf{J}_{c0}\boldsymbol{\omega} + \mathbf{J}_{c0}^{-1}\mathbf{u} + \mathbf{J}_{c0}^{-1}\mathbf{d}(\boldsymbol{\omega}, \mathbf{u}) \end{cases} \quad (6)$$

where  $\mathbf{d}(\boldsymbol{\omega}, \mathbf{u}) = -\mathbf{J}_{c0}\tilde{\mathbf{J}}_c^*(\boldsymbol{\omega}^\times\mathbf{J}_{c0}\boldsymbol{\omega}) - \boldsymbol{\omega}^\times\tilde{\mathbf{J}}_c^*\boldsymbol{\omega} - \mathbf{J}_{c0}\tilde{\mathbf{J}}_c^*\boldsymbol{\omega}^\times\tilde{\mathbf{J}}_c^*\boldsymbol{\omega} + \mathbf{J}_{c0}\tilde{\mathbf{J}}_c^*\mathbf{u} + (\mathbf{I}_3 + \mathbf{J}_{c0}\tilde{\mathbf{J}}_c^*)\boldsymbol{\tau}_d$ . The dynamic model can be written in a compact form:

$$\dot{\mathbf{x}} = \underbrace{\mathbf{f}(\mathbf{x}, \mathbf{u})}_{\text{nominal model}} + \mathbf{B} \underbrace{\tilde{\Delta}(\mathbf{x}, \mathbf{u})}_{\text{unknown model}} \quad (7)$$

where  $\mathbf{x} = \text{vec}(\mathbf{q}_e, \boldsymbol{\omega}) \in \mathbb{R}^6$  is the state vector of the combined spacecraft,  $\mathbf{B} = [\mathbf{0}_{3 \times 3} \quad \mathbf{I}_3]^\top \in \mathbb{R}^{6 \times 3}$  is a projection matrix,  $\mathbf{f}(\mathbf{x}, \mathbf{u})$  denotes the known, nominal part of the dynamics which has the following form:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \frac{1}{2}(\mathbf{q}_{e0}\mathbf{I}_3 + \mathbf{q}_e^\times)\boldsymbol{\omega} \\ -\mathbf{J}_{c0}^{-1}\boldsymbol{\omega}^\times\mathbf{J}_{c0}\boldsymbol{\omega} + \mathbf{J}_{c0}^{-1}\mathbf{u} \end{bmatrix} \in \mathbb{R}^6. \quad (8)$$

The state and control-dependent unknown model  $\tilde{\Delta}(\mathbf{x}, \mathbf{u}) = \mathbf{J}_{c0}^{-1}\mathbf{d}(\boldsymbol{\omega}, \mathbf{u}) \in \mathbb{R}^3$  includes the model uncertainties and the reactive attitude maneuvering torque of the target. Note that  $\tilde{\Delta}$  also implicitly depends on  $t$  because it contains the time-varying external disturbance  $\boldsymbol{\tau}_d(t)$ . In order to be able to design a controller, the system is required to satisfy the following non-restrictive conditions:

**CONDITION 1** There exists known and bounded constants  $\lambda_J, \lambda_c > 0$ , such that the nominal part of the inertia matrix satisfies  $\lambda_{\max}(\mathbf{J}_{c0}) \leq \lambda_J$  and  $\lambda_{\min}(\mathbf{J}_{c0}) \geq \lambda_c$ .

**CONDITION 2** The unknown function  $\tilde{\Delta}$  is globally bounded.

The combined spacecraft performs as a ‘‘black box’’ data-generating system, i.e., only input and output data is available from it, just in case of a real spacecraft.

**ASSUMPTION 1** Measurements of the state (sampling of  $\mathbf{x}(t)$ ) are obtained online under the sampling time  $T_s \in \mathbb{R}^+$ . The sampling provided value of the states is denoted as  $\mathbf{q}_e[k] := \mathbf{q}_e(kT_s)$ ,  $\boldsymbol{\omega}[k] := \boldsymbol{\omega}(kT_s)$ , and  $\mathbf{x}[k] := \mathbf{x}(kT_s)$ . Also, we assume that an approximation of the state derivatives  $\dot{\mathbf{x}}[k]$  can be obtained by numerical differentiation. The approximation error can be considered as being part of the measurement noise. Additionally, the control input  $\mathbf{u}$  is generated via ideal *zero-order-hold* (ZOH) actuation with no delay and synchronized with the

sampling, which means that the discrete control signal will be kept constant until the next sampling moment:

$$\mathbf{u}(t) := \mathbf{u}[k], \quad \forall t \in (kT_s, (k+1)T_s]. \quad (9)$$

## B. Control Objectives

The primary goal of this paper is to design an adaptive online learning attitude takeover control strategy for the combined spacecraft in the presence of unknown dynamics and attitude maneuverability of the target, such that the attitude of the combined spacecraft follows the desired orientation  $\mathbf{Q}_d$ , while the attitude error  $\mathbf{Q}_e$  and the angular velocity  $\boldsymbol{\omega}$  are ultimately uniformly bounded, and converge to a small set containing the origin.

## III. Gaussian Process for Online Model Learning

To achieve the aforementioned control objectives, in this section, our goal is to construct a data-driven and probabilistic model  $\Delta$  of the unknown dynamics  $\tilde{\Delta}$  from previously collected measurement data, and improve the accuracy of the regressed model gradually as more data becomes available and track possible time variation of  $\tilde{\Delta}$ .

### A. Gaussian Process Regression

*GP regression* (GPR) is a powerful non-parametric framework for learning nonlinear functions from data, where the GP itself can be seen as a distribution over functions [21]. The main advantage of GPR is that it not only provides the estimated mean of the unknown function, but also its variance, which implies the regression accuracy, namely, the model confidence. To approximate the unknown function  $\tilde{\Delta}$ , we consider  $\Delta(\tilde{\mathbf{x}})$  as a GP which is trained based on the following data set consisting of  $N$  collected sampled measurements:

$$\mathcal{D}_N = \{\mathbf{y}[i], \tilde{\mathbf{x}}[i]\}_{i=1}^N \quad (10)$$

where the state-input pairs  $\tilde{\mathbf{x}}[k] = \text{vec}(\mathbf{x}[k], \mathbf{u}[k]) \in \mathbb{R}^9$  and  $\mathbf{y}[k] = \tilde{\Delta}(\tilde{\mathbf{x}}[k]) + \boldsymbol{\epsilon}[k] \triangleq \dot{\mathbf{x}}[k] - \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k]) + \boldsymbol{\epsilon}[k]$  denote the training inputs and outputs, respectively, and  $\boldsymbol{\epsilon}[k]$  represents the *independent and identically distributed* (i.i.d.) measurement noise with  $\boldsymbol{\epsilon}[k] \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$  and additional approximation error resulting from numerical differentiation.

A vectorial *Gaussian process*  $\mathcal{GP} : \mathbb{R}^9 \rightarrow \mathbb{R}^3$  assigns to every point  $\tilde{\mathbf{x}} \in \mathbb{R}^9$  a random variable  $\mathcal{GP}(\tilde{\mathbf{x}})$  taking values

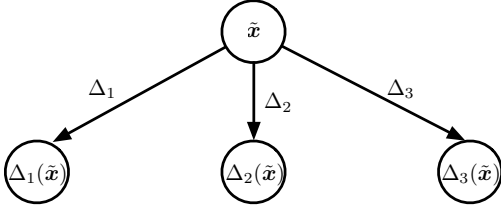


Fig. 2: Illustration of the multi-dimensional GP structure.

in  $\mathbb{R}^3$  such that, for any finite set  $\{\tilde{\mathbf{x}}[\tau]\}_{\tau=1}^N \subset \mathbb{R}^3$ , the joint probability distribution of  $\mathcal{GP}(\tilde{\mathbf{x}}[1]), \dots, \mathcal{GP}(\tilde{\mathbf{x}}[N])$  is multi-dimensional Gaussian. This constitutes a prior distribution over functions, which is denoted by:

$$\Delta(\tilde{\mathbf{x}}) \sim \mathcal{GP}(\boldsymbol{\mu}(\tilde{\mathbf{x}}), \boldsymbol{\kappa}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')), \quad (11)$$

where  $\boldsymbol{\mu}(\tilde{\mathbf{x}})$  is the mean function and  $\boldsymbol{\kappa}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') \triangleq \text{cov}(\Delta(\tilde{\mathbf{x}}), \Delta(\tilde{\mathbf{x}}'))$  is the positive semi-definite covariance function which corresponds to a measure of correlation of any two data points  $(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')$ . Furthermore, the GP model is usually implemented for each dimension of the GP output separately, i.e. in terms of scalar-valued  $\Delta_j(\tilde{\mathbf{x}}) \sim \mathcal{GP}(\mu_j(\tilde{\mathbf{x}}), \kappa_j(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}'))$  which approximates the corresponding  $\Delta_j$  with  $j \in \mathbb{I}_1^3$ . The structure of the multi-dimensional GP is illustrated in Fig. 2, where the outputs are assumed to be uncorrelated, i.e.,  $\Delta_j$  is mutually independent, resulting in the choice of  $\boldsymbol{\kappa} = \text{diag}(\kappa_1, \kappa_2, \kappa_3)$ . It should be noted that this assumption is commonly used in GP-based control approaches and it is mild for a real spacecraft task. In this paper, the kernel function is chosen from the exponential family, i.e., the *Squared Exponential Automatic Relevance Determination* (SEARD) is taken as a prior due to its universal approximation capability:

$$\kappa_j(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \sigma_{f,j}^2 \exp\left(-\frac{1}{2}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')^\top \boldsymbol{\Lambda}_j^{-1}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}')\right) \quad (12)$$

where the diagonal matrix  $\boldsymbol{\Lambda}_j = \text{diag}(\lambda_{j,1}^2, \dots, \lambda_{j,9}^2)$  and  $\sigma_{f,j}^2 \in \mathbb{R}^+$  are the length-scale hyperparameters and signal variance, respectively. In order to be able to capture the unknown function  $\Delta$  using the chosen kernel defined GP (11), the following condition is required to be satisfied:

**CONDITION 3** Each  $\Delta_j$  has a bounded *reproducing kernel Hilbert Space* (RKHS) norm w.r.t. the chosen kernel  $\kappa_j(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')$ , that is,  $\|\Delta_j\|_{\mathcal{H}} < \infty$ .

**REMARK 1** The RKHS norm of  $\Delta_j$  can be interpreted as a quantitative assessment of the function smoothness, indicating that the function is “well-behaved” w.r.t. the selected kernel.

According to the training data set (10), we define  $\tilde{\mathbf{X}} = \text{vec}(\tilde{\mathbf{x}}[1], \dots, \tilde{\mathbf{x}}[N])$  and  $\mathbf{Y}_j = \text{vec}(\mathbf{y}_j[1], \dots, \mathbf{y}_j[N])$  for  $j \in \mathbb{I}_1^3$ . The Gaussian prior for the function  $\Delta$  and the model likelihood of  $\mathcal{D}_N$  is denoted as:

$$\mathbb{P}(\Delta_j) = \mathcal{N}(\Delta_j | \mathbf{0}, \mathbf{K}_{N,j}) \quad (13a)$$

$$\mathbb{P}(\mathbf{Y}_j | \Delta_j) = \mathcal{N}(\mathbf{Y}_j | \Delta_j, \sigma_{\epsilon,j}^2 \mathbf{I}_N) \quad (13b)$$

where the so-called Gram matrix  $\mathbf{K}_{N,j} \in \mathbb{S}^{N \times N}$  represents the symmetric and semi-definite covariance matrix on the training set  $\mathcal{D}_N$ :

$$\mathbf{K}_{N,j} = \begin{bmatrix} \kappa_j(\tilde{\mathbf{x}}[1], \tilde{\mathbf{x}}[1]) & \cdots & \kappa_j(\tilde{\mathbf{x}}[1], \tilde{\mathbf{x}}[N]) \\ \vdots & \ddots & \vdots \\ \kappa_j(\tilde{\mathbf{x}}[N], \tilde{\mathbf{x}}[1]) & \cdots & \kappa_j(\tilde{\mathbf{x}}[N], \tilde{\mathbf{x}}[N]) \end{bmatrix} \quad (14)$$

and  $\boldsymbol{\Delta}_j = \text{vec}(\Delta_j(\tilde{\mathbf{x}}[1]), \dots, \Delta_j(\tilde{\mathbf{x}}[N]))$ .

According to Bayes' Theorem, the posterior distribution can be obtained by *maximum a posterior* (MAP) estimation:

$$\begin{aligned} \mathbb{P}(\Delta_j | \mathbf{Y}_j) &= \frac{\mathbb{P}(\mathbf{Y}_j | \Delta_j) \mathbb{P}(\Delta_j)}{\mathbb{P}(\mathbf{Y}_j)} \\ &\propto \mathcal{N}(\Delta_j | \mathbf{K}_{N,j} (\mathbf{K}_{N,j} + \sigma_{\epsilon,j}^2 \mathbf{I}_N)^{-1} \mathbf{Y}_j, \\ &\quad \sigma_{\epsilon,j}^2 \mathbf{K}_{N,j} (\mathbf{K}_{N,j} + \sigma_{\epsilon,j}^2 \mathbf{I}_N)^{-1}). \end{aligned} \quad (15)$$

Then, the predictive distribution of  $\Delta_j(\tilde{\mathbf{x}}^*)$  on a test point  $\tilde{\mathbf{x}}^*$  can be derived as:

$$\begin{aligned} \mathbb{P}(\Delta_j^* | \tilde{\mathbf{x}}^*, \mathcal{D}_N) &= \int \mathbb{P}(\Delta_j^* | \tilde{\mathbf{x}}^*, \Delta_j, \tilde{\mathbf{X}}) \mathbb{P}(\Delta_j | \mathbf{Y}_j) d\Delta_j \\ &= \mathcal{N}(\mu_{\Delta,j}(\tilde{\mathbf{x}}^*), \sigma_{\Delta,j}^2(\tilde{\mathbf{x}}^*)) \end{aligned} \quad (16)$$

where the first term inside the integral satisfies the joint distribution:

$$\begin{aligned} \mathbb{P}(\Delta_j^* | \tilde{\mathbf{x}}^*, \Delta_j, \tilde{\mathbf{X}}) &= \mathcal{N}(\Delta_j^* | \mathbf{K}_{*N,j} \mathbf{K}_{N,j}^{-1} \Delta_j, k_{**j} - \mathbf{K}_{*N,j} \mathbf{K}_{N,j}^{-1} \mathbf{K}_{N*,j}). \end{aligned} \quad (17)$$

Combining (15), (16) and (17), we can obtain the posterior mean and variance function for each dimension  $j$ :

$$\mu_{\Delta,j}(\tilde{\mathbf{x}}^*) = \mathbf{K}_{*N,j} (\mathbf{K}_{N,j} + \sigma_{\epsilon,j}^2 \mathbf{I})^{-1} \mathbf{Y}_j, \quad (18a)$$

$$\sigma_{\Delta,j}^2(\tilde{\mathbf{x}}^*) = k_{**j} - \mathbf{K}_{*N,j} (\mathbf{K}_{N,j} + \sigma_{\epsilon,j}^2 \mathbf{I})^{-1} \mathbf{K}_{N*,j}, \quad (18b)$$

which gives a predictive vectorial GP with mean and variance:

$$\boldsymbol{\mu}_{\Delta}(\tilde{\mathbf{x}}^*) = \text{vec}(\mu_{\Delta,1}(\tilde{\mathbf{x}}^*), \mu_{\Delta,2}(\tilde{\mathbf{x}}^*), \mu_{\Delta,3}(\tilde{\mathbf{x}}^*)), \quad (19a)$$

$$\boldsymbol{\Sigma}_{\Delta}(\tilde{\mathbf{x}}^*) = \text{diag}(\sigma_{\Delta,1}^2(\tilde{\mathbf{x}}^*), \sigma_{\Delta,2}^2(\tilde{\mathbf{x}}^*), \sigma_{\Delta,3}^2(\tilde{\mathbf{x}}^*)). \quad (19b)$$

Furthermore, to obtain an optimal choice of the hyperparameters<sup>1</sup>  $\boldsymbol{\theta}_j = \text{vec}(\sigma_{\epsilon,j}^2, \sigma_{f,j}^2, \lambda_{j,1}, \dots, \lambda_{j,9})$ , the GP model (11) is trained by maximizing the log-likelihood  $\mathcal{L}(\boldsymbol{\theta}_j) = \log \mathcal{N}(\mathbf{0}, \mathbf{K}_{N,j} + \sigma_{\epsilon,j}^2 \mathbf{I}_N)$  w.r.t.  $\boldsymbol{\theta}_j$  for each dimension  $j$ :

$$\hat{\boldsymbol{\theta}}_j \in \arg \max_{\boldsymbol{\theta}_j} \mathcal{L}(\boldsymbol{\theta}_j), \quad (20)$$

which optimization problem can be efficiently solved by a conjugate gradient-based algorithm [35].

<sup>1</sup>We initialize  $\boldsymbol{\theta}_j$  based on  $\mathcal{D}_N$ :  $\log \boldsymbol{\theta}_j^0 = \text{vec}(\log(\text{std}(\mathbf{Y}_j/10)), \log(\text{std}(\mathbf{Y}_j)), \log(\text{std}(\tilde{\mathbf{X}})))$ . In [34], it was illustrated that although the initial guess for the hyperparameters may have influences on the optimization results, its impact on the regression accuracy of the GP model is almost negligible. Therefore, it is generally advisable to choose a relatively simple initial guess.

## B. Recursive Online Sparse GP Regression

As we could see in (20), the computational complexity for the conjugate gradient-based algorithm is  $\mathcal{O}(N^3)$  per iteration, which is cubic in terms of the size of the training data set. According to (18a) and (18b), the computational complexity for predictive mean and variance per test case is  $\mathcal{O}(N)$  and  $\mathcal{O}(N^2)$ , respectively. Thus, the standard GP is not suitable for large training data sets. However, it is essential to collect a large number of data pairs to explore the state space as much as possible and ensure a sufficiently high regression accuracy. Furthermore, the GP model presented in Section III.A is an offline modeling approach, i.e., the trained GP model is kept fixed online and considered to be sufficient to describe the various on-orbit scenarios. Nevertheless, for the combined spacecraft takeover control missions, the unknown part of the dynamics  $\Delta$  may be time-varying due to further attitude maneuvers of the target. Therefore, this section proposes a recursive online sparse GP algorithm (denoted by ROSGP), which significantly reduces the computational complexity of the data-driven model learning while making full use of the online streaming data to update the GP model in real time.

### 1. Sparse GP with inducing points

As discussed in Section III.A, the diagonal assumption for the prior covariance function  $\kappa$  allows for the training of the predictive distribution independently for each dimension  $j$ . Thus, for the simplicity of notation, we will drop the indexing for the output dimension  $j$ .

We will introduce a novel generalization of the original *sparse GP with inducing inputs* (SPGP) [36] approach in terms of an efficient online update step. For this we first briefly summarize the SPGP method. The main idea of sparse GP is to find a set of inducing inputs  $\tilde{\mathbf{X}}_u = \text{vec}(\tilde{\mathbf{x}}_{u,1}, \dots, \tilde{\mathbf{x}}_{u,M})$  corresponding to inducing outputs  $\Delta_u = \text{vec}(\Delta_{u,1}, \dots, \Delta_{u,M})$  of size  $M \ll N$ . Akins to (13a), the inducing points follow the Gaussian prior distribution:

$$\mathbb{P}(\Delta_u) = \mathcal{N}(\Delta_u | \mathbf{0}, \mathbf{K}_M) \quad (21)$$

where  $\mathbf{K}_M \in \mathbb{S}^{M \times M}$  denotes the Gram matrix in terms of inducing inputs. According to the *fully independent training conditional approximation* (FITC), given the GP inputs with (21), the function values  $\Delta = \text{vec}(\Delta(\tilde{\mathbf{x}}[1]), \dots, \Delta(\tilde{\mathbf{x}}[N]))$  are i.i.d. Then, the model likelihood is

$$\begin{aligned} \mathbb{P}(\mathbf{Y} | \tilde{\mathbf{X}}, \Delta_u, \tilde{\mathbf{X}}_u) &= \prod_{i=1}^N \mathbb{P}(\mathbf{y}[i] | \tilde{\mathbf{x}}[i], \Delta_u, \tilde{\mathbf{X}}_u) \\ &= \mathcal{N}(\mathbf{Y} | \mathbf{K}_{NM} \mathbf{K}_M^{-1} \Delta_u, \mathbf{\Gamma} + \sigma_\epsilon^2 \mathbf{I}_N) \end{aligned} \quad (22)$$

where  $[\mathbf{K}_{MN}]_{i,j} = \kappa(\tilde{\mathbf{x}}_{u,i}, \tilde{\mathbf{x}}[j])$  denotes the covariance matrix between  $\mathcal{D}_M$  and  $\mathcal{D}_N$ ,  $\mathbf{Q}_N = \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{K}_{MN}$  can be seen as an approximation of  $\mathbf{K}_N$ , and  $\mathbf{\Gamma} = \text{diag}(\mathbf{K}_N - \mathbf{Q}_N) \in \mathbb{R}^{N \times N}$  represents the diagonal covariance matrix which obtains its diagonal structure from the independence between  $\Delta$  and  $\Delta_u$ .

On the basis of the Bayes's Theorem and the Gaussian prior (21), the approximated posterior distribution is given by the MAP estimate:

$$\begin{aligned} \mathbb{P}(\Delta_u | \mathbf{Y}, \tilde{\mathbf{X}}_u, \tilde{\mathbf{X}}) &\propto \mathbb{P}(\mathbf{Y} | \Delta_u, \tilde{\mathbf{X}}_u, \tilde{\mathbf{X}}) \mathbb{P}(\Delta_u) \\ &= \mathcal{N}(\Delta_u | \mathbf{K}_{NM} \mathbf{Q}_M^{-1} \mathbf{K}_{MN} (\mathbf{\Gamma} + \sigma_\epsilon^2 \mathbf{I}_N)^{-1} \mathbf{Y}, \\ &\quad \mathbf{K}_M \mathbf{Q}_M^{-1} \mathbf{K}_M) \end{aligned} \quad (23)$$

where  $\mathbf{Q}_M = \mathbf{K}_M + \mathbf{K}_{MN} (\mathbf{\Gamma} + \sigma_\epsilon^2 \mathbf{I}_N)^{-1} \mathbf{K}_{NM}$ .

Consequently, the associated posterior distribution of  $\Delta(\tilde{\mathbf{x}}^*)$  at a new test point  $\tilde{\mathbf{x}}^*$  is computed by

$$\mathbb{P}(\Delta^* | \mathcal{D}_N, \tilde{\mathbf{x}}^*, \tilde{\mathbf{X}}_u) = \mathcal{N}(\mu_\Delta(\tilde{\mathbf{x}}^*), \sigma_\Delta^2(\tilde{\mathbf{x}}^*)) \quad (24)$$

with predictive mean and variance as:

$$\mu_\Delta(\tilde{\mathbf{x}}^*) = \mathbf{K}_{*M} \mathbf{Q}_M^{-1} \mathbf{K}_{MN} (\mathbf{\Gamma} + \sigma_\epsilon^2 \mathbf{I}_N)^{-1} \mathbf{Y}, \quad (25a)$$

$$\sigma_\Delta^2(\tilde{\mathbf{x}}^*) = k_{**} - \mathbf{K}_{*M} (\mathbf{K}_M^{-1} - \mathbf{Q}_M^{-1}) \mathbf{K}_{M*}. \quad (25b)$$

The inducing data points  $\mathcal{D}_M = \{\Delta_u, \tilde{\mathbf{X}}_u\}$  can be seen as additional hyper-parameters and are optimized along with the original GP hyperparameters  $\theta = \text{vec}(\sigma_\epsilon^2, \sigma_f^2, \lambda_1, \dots, \lambda_9)$  by maximizing the marginal likelihood function, which can be computed by integrating (21) and (22):

$$\begin{aligned} \mathbb{P}(\mathbf{Y}) &= \int \mathbb{P}(\mathbf{Y} | \tilde{\mathbf{X}}, \Delta_u, \tilde{\mathbf{X}}_u) \mathbb{P}(\Delta_u) d\Delta_u \\ &= \mathcal{N}(\mathbf{0}, \mathbf{Q}_N + \mathbf{\Gamma} + \sigma_\epsilon^2 \mathbf{I}_N). \end{aligned} \quad (26)$$

To initialize the optimization, one reliable approach is to pick random points from the original data set  $\mathcal{D}_N$ .

It is worth mentioning that the inverse of  $\mathbf{K}_N + \sigma_\epsilon^2 \mathbf{I}_N$  is reduced to the inverse of the diagonal matrix  $\mathbf{\Gamma} + \sigma_\epsilon^2 \mathbf{I}_N$ . In the SPGP based on the FITC assumption, the computational load  $\mathcal{O}(M^2N)$  mainly comes from the matrix multiplication  $\mathbf{K}_{MN} (\mathbf{\Gamma} + \sigma_\epsilon^2 \mathbf{I}_N)^{-1} \mathbf{K}_{NM}$ . Subsequently, for each test point  $\tilde{\mathbf{x}}^*$ , the computational complexity for corresponding predictive mean and variance is decreased to  $\mathcal{O}(M)$  and  $\mathcal{O}(M^2)$ , respectively. Moreover, compared to the marginal likelihood of the standard GP,  $\mathbf{Q}_N$  is a low-rank approximation of Gram matrix  $\mathbf{K}_N$ , which reduces the computational complexity from  $\mathcal{O}(N^3)$  to  $\mathcal{O}(M^2N)$  during the hyperparameter training. Similar to the standard GP, the SPGP model can be trained by maximizing the log form of the likelihood (26) to find the optimal hyperparameters and  $\mathcal{D}_M$  by means of a conjugate gradient-based algorithm, which has a computational complexity of  $\mathcal{O}(M^2N + 3MN)$  per iteration.

### 2. Recursive Online Sparse GP Regression Algorithm

In this subsection, a novel online update strategy for SPGP is proposed, where the offline trained GP model is recursively updated with the online measurement data sampled at the current time moment  $k \in \mathbb{N}$ .

Equation (25a) can be rewritten into a linear combination of  $M$  kernel functions with the current time moment  $k$ :

$$\mu_{\Delta,k}(\tilde{\mathbf{x}}) = \sum_{j=1}^M \alpha_j \kappa(\tilde{\mathbf{x}}_{u,j}, \tilde{\mathbf{x}}[k]) = \boldsymbol{\alpha}^\top [k] \mathbf{K}_M [k] \quad (27)$$

where  $\alpha[0] = \mathbf{Q}_M^{-1} \mathbf{K}_{MN} (\Gamma + \sigma_\epsilon^2 \mathbf{I}_N)^{-1} \mathbf{Y} \in \mathbb{R}^M$  is the initial weight vector obtained from offline training and  $[\mathbf{K}_{M[k]}]_j = \kappa(\tilde{\mathbf{x}}_{u,j}, \tilde{\mathbf{x}}[k])$  is the corresponding kernel slice evaluated at the input  $\tilde{\mathbf{x}}[k]$ .

At time moment  $k$ , consider the data set  $\mathcal{D}_k$  given in (10). Define a performance index  $\mathcal{W}(\alpha)$  w.r.t.  $\alpha$  over the extended data set  $\mathcal{D}_k = \mathcal{D}_{k-1} \cup (y[k], \tilde{\mathbf{x}}[k])$  which contains  $N_k$  data pairs:

$$\mathcal{W}(\alpha) = \sum_{i=1}^{N_k} \lambda^{N_k-i} (y[i] - \alpha^\top \mathbf{K}_{Mi})^2 + \varsigma \lambda^{N_k+1} \|\alpha\|^2 \quad (28)$$

where  $[\mathbf{K}_{Mi}]_j = \kappa(\tilde{\mathbf{x}}_{u,j}, \tilde{\mathbf{x}}[i])$ ,  $0 < \lambda \leq 1$  is a user-defined parameter, also known as the forgetting factor. The choice of parameter  $\varsigma$  will be discussed later in this section. The weight vector  $\alpha[k]$  at step  $k$  is taken as the minimizer of (28), i.e.:

$$\alpha[k] = \arg \min_{\alpha \in \mathbb{R}^M} \mathcal{W}(\alpha). \quad (29)$$

The optimization problem (29) has an analytical solution. Note that

$$\Phi[k] \alpha[k] \rho[k] \quad (30)$$

where

$$\Phi[k] = \sum_{i=1}^{N_k} \lambda^{N_k-i} \mathbf{K}_{Mi} \mathbf{K}_{Mi}^\top + \varsigma \lambda^{N_k+1} \mathbf{I}_M, \quad (31a)$$

$$\rho[k] = \sum_{i=1}^{N_k} \lambda^{N_k-i} \mathbf{K}_{Mi} y[i]. \quad (31b)$$

Next, (31a) and (31b) can be written in a separate form:

$$\Phi[k] = \lambda \Phi[k-1] + \mathbf{K}_{M[k]} \mathbf{K}_{M[k]}^\top, \quad (32a)$$

$$\rho[k] = \lambda \rho[k-1] + \mathbf{K}_{M[k]} y[k]. \quad (32b)$$

Using formula Woodbury's matrix inverse for (32a), one has:

$$\Phi^{-1}[k] = \lambda^{-1} \Phi^{-1}[k-1] - \lambda^{-1} \mathbf{L}[k] \mathbf{K}_{M[k]}^\top \Phi^{-1}[k-1] \quad (33)$$

where

$$\mathbf{L}[k] = \frac{\lambda^{-1} \Phi^{-1}[k-1] \mathbf{K}_{M[k]}}{1 + \lambda^{-1} \mathbf{K}_{M[k]}^\top \Phi^{-1}[k-1] \mathbf{K}_{M[k]}}. \quad (34)$$

For the convenience of notation, we define  $\mathbf{P}[k] = \Phi^{-1}[k]$ . Thus, from (33) and (34), we have:

$$\mathbf{P}[k] = \lambda^{-1} \mathbf{P}[k-1] - \lambda^{-1} \mathbf{L}[k] \mathbf{K}_{M[k]}^\top \mathbf{P}[k-1], \quad (35a)$$

$$\mathbf{L}[k] = \mathbf{P}[k] \mathbf{K}_{M[k]}. \quad (35b)$$

Subsequently, combining (30), (32b), (33), and (34), one can derive:

$$\alpha[k] = \alpha[k-1] - \mathbf{L}[k] \mathbf{K}_{M[k]}^\top \alpha[k-1] + \mathbf{L}[k] y[k] \quad (36)$$

Finally, the weight vector  $\alpha[k]$  can be recursively updated by:

$$\alpha[k] = \alpha[k-1] + \mathbf{L}[k] r[k] \quad (37)$$

where  $r[k] = y[k] - \alpha_j^\top[k-1] \mathbf{K}_{M[k]}$ .

**REMARK 2** The online update routine starts from the initial weight vector  $\alpha[0]$  and the initial user-defined matrix  $\mathbf{P}[0]$ .  $\alpha[0]$  can be obtained by offline training of the GP and  $\mathbf{P}[0]$  is usually selected as  $\mathbf{P}[0] = \varsigma^{-1} \mathbf{I}_M$  with  $0 < \varsigma \leq 1$ . The choice of  $\varsigma$  is on the basis of the confidence level of the offline trained GP.

**REMARK 3** It is worth mentioning that, the convergence of the proposed recursive online sparse GP is inherently guaranteed because it minimizes the performance index (28) at each iterative step.

**REMARK 4** Compared with the existing online GP methods, such as SOGP [25], [27], and the evolving GP in [30], [31], the proposed recursive online sparse GP do not involve any re-optimization of the hyperparameters nor it requires data-dictionary update at every time step, which ensures low online computational cost.

## IV. GP-based Online Learning Control

Recalling the control objective given in Section II. B, the goal of this paper is to design a GP-based online learning control strategy to ensure that, for a given attitude set point  $\mathbf{Q}_d = [q_{d0} \mathbf{q}_d^\top]^\top \in \mathbb{Q}^3$ , the closed-loop error and velocity states  $(\mathbf{Q}_e, \boldsymbol{\omega})$  converge to  $(\mathbf{1}, \mathbf{0})$  even under the unknown dynamics and attitude maneuverability of target. Furthermore, the proposed ROSGP algorithm is employed to derive a probabilistic model of  $\check{\Delta}(\tilde{\mathbf{x}})$ , which is utilized for feedforward compensation of the unknown function.

### A. Controller Design

In order to design the controller, we begin by considering the following lemmas and conditions.

**LEMMA 1** Consider a GP trained on  $\mathcal{D}_N$  collected from system (7), which satisfies Condition 3. The estimation error  $\|\mu_\Delta(\tilde{\mathbf{x}}) - \check{\Delta}(\tilde{\mathbf{x}})\|$  is bounded for all  $\tilde{\mathbf{x}}$  on the compact set  $\Omega \subset \mathbb{R}^9$  with probability  $(1 - \delta)^3$ :

$$\mathbb{P}\{\forall \tilde{\mathbf{x}} \in \Omega, \|\mu_\Delta(\tilde{\mathbf{x}}) - \check{\Delta}(\tilde{\mathbf{x}})\| \leq \|\beta\| \|\Sigma_\Delta^{1/2}(\tilde{\mathbf{x}})\|\} \geq (1 - \delta)^3 \quad (38)$$

where  $\delta \in (0, 1)$ ,  $\beta = \text{vec}(\beta_1, \beta_2, \beta_3)$  denotes:

$$\beta_j = \sqrt{2 \|\check{\Delta}_j\|_{\mathcal{H}}^2 + 300 \gamma_j \log^3((N+1)/\delta)}, \quad (39)$$

and  $\gamma_j = \max \frac{1}{2} \log |\mathbf{I}_N + \sigma_{\epsilon,j}^{-2} \mathbf{K}_{N,j}|$  represents the maximum information gain w.r.t. the kernel  $\kappa_j, \forall j \in \mathbb{I}_3^1$ .

**PROOF** See [37, Lemma 2]. This lemma is a vectorial generalization of the scalar case given in [38]. ■

**PROPOSITION 1** Consider a GP recursively trained on the initial data set  $\mathcal{D}_N$  and online samples obtained during time interval  $[0, k]$  from system (7) which satisfies Condition 3. The estimation error  $\|\mu_{\Delta,k}(\tilde{\mathbf{x}}) - \check{\Delta}(\tilde{\mathbf{x}})\|$  is bounded for all



$\tilde{\mathbf{x}}$  on the compact set  $\Omega \subset \mathbb{R}^9$  with probability  $(1 - \delta)^3$ :

$$\mathbb{P}\{\forall \tilde{\mathbf{x}} \in \Omega, \|\boldsymbol{\mu}_{\Delta,k}(\tilde{\mathbf{x}}) - \check{\Delta}(\tilde{\mathbf{x}})\| \leq \|\boldsymbol{\beta}\| \|\boldsymbol{\Sigma}_{\Delta}^{1/2}(\tilde{\mathbf{x}})\| \} \geq (1 - \delta)^3 \quad (40)$$

where  $\boldsymbol{\mu}_{\Delta,k}(\tilde{\mathbf{x}})$  represents the predictive mean of the recursively updated GP given by (27) at time  $k \in \mathbb{N}$ .

The proof of Proposition 1 follows the lines of the proof of Lemma 1. Proposition 1 ensures the boundedness of the GP regression error between the true function  $\check{\Delta}(\tilde{\mathbf{x}})$  and the recursively updated predictive mean function  $\boldsymbol{\mu}_{\Delta,k}(\tilde{\mathbf{x}})$  with a high probability, where the error bound is proportional to the predictive standard deviation [39].

Based on the online adapted GP, we propose the following controller:

$$\mathbf{u}[k] = -\check{\zeta}_p(\check{k}_p, \boldsymbol{\Sigma}_{\Delta}(\tilde{\mathbf{x}}[k]))\mathbf{q}_e[k] - \check{\zeta}_d(\check{k}_d, \boldsymbol{\Sigma}_{\Delta}(\tilde{\mathbf{x}}[k]))\boldsymbol{\omega}[k] - \mathbf{J}_{c0}\boldsymbol{\mu}_{\Delta,k}(\tilde{\mathbf{x}}[k]) + \boldsymbol{\omega}^\times[k]\mathbf{J}_{c0}\boldsymbol{\omega}[k], \quad (41)$$

where  $k \in \mathbb{N}$  is the discrete time and  $[\boldsymbol{\mu}_{\Delta,k}]_j = \boldsymbol{\alpha}_j^\top[k]\mathbf{K}_{M^+}^j$ ,  $j \in \mathbb{I}_1^3$  is the predictive mean updated at  $k$ -th step according to the ROSGP algorithm proposed in Section III.B. The functions  $\check{\zeta}_p$  and  $\check{\zeta}_d$  correspond to the feedback gains of the proposed control law, parameterized in terms of  $\check{k}_p \in \mathbb{R}^{n_p}$ ,  $\check{k}_d \in \mathbb{R}^{n_d}$ . These functions are chosen such that the following condition is satisfied:

**CONDITION 4** For given sets  $\mathbb{S}_* \subseteq \mathbb{S}^{3 \times 3}$  and  $\mathbb{K}_* \subseteq \mathbb{R}$ , the symmetric functions  $\check{\zeta}_p, \check{\zeta}_d : \mathbb{S}^{3 \times 3} \rightarrow \mathbb{S}^{3 \times 3}$  are monotone increasing w.r.t.  $K$  and bounded in the sense that the minimum  $\check{\zeta}_\tau$  of  $\sigma_{\min}(\check{\zeta}_\tau(K, \check{k}_\tau))$  and the maximum  $\bar{\zeta}_\tau$  of  $\sigma_{\min}(\check{\zeta}_\tau(K, \check{k}_\tau))$  exist over  $K \in \mathbb{S}_*$  and  $\check{k}_\tau \in \mathbb{K}_*$  where  $\sigma_{\min}$  and  $\sigma_{\max}$  corresponds to the minimum and maximum singular values and  $\tau \in \{p, d\}$ . Then

$$\begin{aligned} \check{\zeta}_p \|\mathbf{w}\|^2 &\leq \mathbf{w}^\top \check{\zeta}_p(\check{k}_p, K)\mathbf{w} \leq \bar{\zeta}_p \|\mathbf{w}\|^2 \\ \check{\zeta}_d \|\mathbf{w}\|^2 &\leq \mathbf{w}^\top \check{\zeta}_d(\check{k}_d, K)\mathbf{w} \leq \bar{\zeta}_d \|\mathbf{w}\|^2 \end{aligned}$$

holds for all  $\mathbf{w} \in \mathbb{R}^3$ .

There is a wide class of  $\zeta$  functions such that Condition 4 is satisfied. For instance, if  $\mathbb{S}_*$  and  $\mathbb{K}_*$  are bounded sets, then polynomial functions can be selected for  $\zeta_\sigma$ . Otherwise, they can be chosen as saturated functions such as sigmoid or Gauss functions. Furthermore,  $\check{k}_p$  and  $\check{k}_d$  can be seen as tuning parameters that can be adjusted based on practical experience w.r.t. nominal controller design and do not require extensive tuning procedures.

The implementation of the controller is summarized in Algorithm 1.

## B. Stability Analysis

In order to conduct the stability analysis of the proposed scheme, we take the assumption that  $T_s$  is small enough such that  $\mathbf{x}(t) \approx \mathbf{x}[k]$  for  $t \in (kT_s, (k+1)T_s]$ , which is reasonable in case of 10-50 Hz sampling based attitude control of a satellite. This simplifies our analysis as instead of sampled data related issues we can focus on

---

**Algorithm 1** GP-based online learning control strategy for attitude takeover tasks

---

**Initialization:** Choose  $\mathbf{Q}_0, \mathbf{Q}_d, \boldsymbol{\omega}_0, \check{k}_p, \check{k}_d, \lambda, \mathbf{P}[0], k = 0, T_s, N$ .

**Offline Training Phase:**

- 1: Generate training data set  $\mathcal{D}_N$ ;
- 2: Train GP model to optimise  $\hat{\boldsymbol{\theta}}$  and  $\boldsymbol{\alpha}[0]$ .

**Online Control Phase:**

- 3: **for**  $k = 1, 2, \dots$ , **do**
  - 4: Observe the current  $\tilde{\mathbf{x}}[k]$  and  $\mathbf{y}[k]$ ;
  - 5: Update:
    - 6:  $\mathbf{P}[k] \leftarrow \lambda^{-1}\mathbf{P}[k-1] - \lambda^{-1}\mathbf{L}[k]\mathbf{K}_{M[k]}^\top\mathbf{P}[k-1]$
    - 7:  $\mathbf{L}[k] \leftarrow \mathbf{P}[k]\mathbf{K}_{M[k]}$
    - 8:  $r[k] \leftarrow y[k] - \boldsymbol{\alpha}^\top[k-1]\mathbf{K}_{M[k]}$
    - 9:  $\boldsymbol{\alpha}[k] \leftarrow \boldsymbol{\alpha}[k-1] + \mathbf{L}[k]r[k]$
  - 10: Compute  $\boldsymbol{\mu}_{\Delta,k}$  and  $\boldsymbol{\Sigma}_{\Delta}$  with (27) and (25b);
  - 11: Compute  $\mathbf{u}[k]$  using (41);
  - 12: Apply  $\mathbf{u}(t) = \mathbf{u}[k]$  on the system for  $t \in [kT_s, (k+1)T_s]$
  - 13: **end for**
- 

the interplay between the GP-based adaptive control law and the motion dynamics of the combined spacecraft. For this reason we will treat (41) as a continuous time control law and consider  $\boldsymbol{\alpha}$  varying continuously with time  $t$ .

Assume that the true model uncertainty  $\check{\Delta}(\tilde{\mathbf{x}}(t)) = \mathbf{J}_{c0}^{-1}\mathbf{d}(\tilde{\mathbf{x}}(t))$  at time moment  $t$  can be parameterized as  $\mathbf{d}(\tilde{\mathbf{x}}(t)) = \mathbf{J}_{c0}\check{\boldsymbol{\alpha}}^\top(t)\mathbf{K}_{M(t)}$ , where  $\check{\boldsymbol{\alpha}}(t)$  denotes the true value of the weight for time  $t$ . Define  $\tilde{\boldsymbol{\alpha}}(t) = \boldsymbol{\alpha}(t) - \check{\boldsymbol{\alpha}}(t)$  as the deviation of the weight for the time moment  $t$ . Substituting the controller (41) into (6) yields the closed-loop system:

$$\mathbf{J}_{c0}\dot{\boldsymbol{\omega}}(t) = -\check{\zeta}_p(\check{k}_p, \boldsymbol{\Sigma}_{\Delta}(t))\mathbf{q}_e(t) - \check{\zeta}_d(\check{k}_d, \boldsymbol{\Sigma}_{\Delta}(t))\boldsymbol{\omega}(t) - \mathbf{J}_{c0}\tilde{\boldsymbol{\alpha}}^\top(t)\mathbf{K}_{M(t)}. \quad (42)$$

The following theorem shows boundedness of the closed-loop signals.

**THEOREM 1** Consider that the combined spacecraft (7) which satisfies Conditions 1-3 and Assumption 1. Under the proposed GP-based learning control law (41), satisfying Condition 4, where the unknown function  $\check{\Delta}$  is modeled by a multi-dimensional GP (11) that is recursively online updated according to Algorithm 1,  $\mathbf{Q}_e, \boldsymbol{\omega}$  and  $\tilde{\boldsymbol{\alpha}}$  are guaranteed to be ultimately uniformly bounded.

**PROOF** Consider the radially unbounded Lyapunov function:

$$V_0(t) = \frac{1}{2}\boldsymbol{\omega}^\top(t)\mathbf{J}_{c0}\boldsymbol{\omega}(t) + \int_{t-T_s}^t \tilde{\boldsymbol{\alpha}}^\top(\tau)\tilde{\boldsymbol{\alpha}}(\tau)d\tau. \quad (43)$$

By differentiating (43) along the system trajectories (42),

$$\begin{aligned} \dot{V}_0(t) &= \boldsymbol{\omega}^\top(t)(-\check{\zeta}_p(\check{k}_p, \boldsymbol{\Sigma}_{\Delta}(t))\mathbf{q}_e(t) - \check{\zeta}_d(\check{k}_d, \boldsymbol{\Sigma}_{\Delta}(t))\boldsymbol{\omega}(t) \\ &\quad - \mathbf{J}_{c0}\tilde{\boldsymbol{\alpha}}^\top(t)\mathbf{K}_{M(t)} + \tilde{\boldsymbol{\alpha}}^\top(t)\tilde{\boldsymbol{\alpha}}(t) - \tilde{\boldsymbol{\alpha}}^\top(t-T_s)\tilde{\boldsymbol{\alpha}}(t-T_s)) \\ &\leq -\check{\zeta}_d \|\boldsymbol{\omega}(t)\|^2 - \boldsymbol{\omega}^\top\check{\zeta}_p(\check{k}_p, \boldsymbol{\Sigma}_{\Delta}(t))\mathbf{q}_e(t) \\ &\quad - \boldsymbol{\omega}^\top(t)\mathbf{J}_{c0}\tilde{\boldsymbol{\alpha}}^\top(t)\mathbf{K}_{M(t)} + \tilde{\boldsymbol{\alpha}}^\top(t)\tilde{\boldsymbol{\alpha}}(t) \\ &\quad - \tilde{\boldsymbol{\alpha}}^\top(t-T_s)\tilde{\boldsymbol{\alpha}}(t-T_s). \end{aligned} \quad (44)$$

Furthermore, from (37), we have:

$$\begin{aligned}\tilde{\alpha}(t) &= \alpha(t - T_s) + \mathbf{L}(t)r(t) - \check{\alpha}(t), \\ &= \tilde{\alpha}(t - T_s) + \mathbf{L}(t)r(t) + \check{\alpha}(t - T_s) - \check{\alpha}(t).\end{aligned}\quad (45)$$

Define  $\Psi(t) = \check{\alpha}(t - T_s) - \check{\alpha}(t)$  and  $\Theta(t) = \mathbf{L}(t)r(t)$ . Then, it follows that:

$$\begin{aligned}\tilde{\alpha}^\top(t)\tilde{\alpha}(t) &= \tilde{\alpha}^\top(t - T_s)\tilde{\alpha}(t - T_s) + 2\tilde{\alpha}^\top(t - T_s)\Theta(t) \\ &\quad + 2\tilde{\alpha}^\top(t - T_s)\Psi(t) + 2\Theta^\top(t)\Psi(t).\end{aligned}\quad (46)$$

Using the special case of Young's inequality  $\mathbf{x}^\top\mathbf{y} \leq \frac{1}{2}\mathbf{x}^\top\mathbf{x} + \frac{1}{2}\mathbf{y}^\top\mathbf{y}$ , the following inequalities can be derived:

$$2\tilde{\alpha}^\top(t - T_s)\Theta(t) \leq \|\tilde{\alpha}(t - T_s)\|^2 + \|\Theta(t)\|^2 \quad (47a)$$

$$2\tilde{\alpha}^\top(t - T_s)\Psi(t) \leq \|\tilde{\alpha}(t - T_s)\|^2 + \|\Psi(t)\|^2 \quad (47b)$$

$$-\omega^\top(t)\zeta_p(\check{k}_p, \Sigma_\Delta(t))\mathbf{q}_e(t) \leq \frac{1}{2}\bar{\zeta}_p^2 + \frac{1}{2}\|\omega(t)\|^2 \quad (47c)$$

$$-\omega^\top(t)\mathbf{J}_{c0}\tilde{\alpha}^\top(t)\mathbf{K}_{M(t)} \leq \frac{1}{2}\lambda_J^2\bar{\sigma}_f^4\|\tilde{\alpha}(t)\|^2 + \frac{1}{2}\|\omega(t)\|^2 \quad (47d)$$

where  $\bar{\sigma}_f = \sup \sigma_{f,j}$ ,  $j \in \mathbb{I}_1^3$ . Substituting the inequalities (47a)-(47d) into (44) leads to:

$$\begin{aligned}\dot{V}_0(t) &\leq -(\zeta_d - 1)\|\omega(t)\|^2 - \gamma\|\tilde{\alpha}(t)\|^2 + \frac{1}{2}\bar{\zeta}_p^2 \\ &\quad + \left(\frac{1}{2}\lambda_J^2\bar{\sigma}_f^4 + 1 + \gamma\right)(3\|\tilde{\alpha}(t - T_s)\|^2 + 3\|\Psi(t)\|^2 + 3\|\Theta(t)\|^2) \\ &\quad - \|\tilde{\alpha}(t - T_s)\|^2,\end{aligned}\quad (48)$$

where  $\gamma$  is a positive constant. Furthermore, according to the definition of  $\Psi$  and  $\Theta$ , it is reasonable to assume that  $\Psi$  and  $\Theta$  are bounded<sup>2</sup>, that is,  $\|\Psi(t)\| \leq \bar{\phi}$  and  $\|\Theta(t)\| \leq \bar{\xi}$ . If  $\zeta_d - 1 > 0$  and  $1 - 3(\frac{1}{2}\lambda_J^2\bar{\sigma}_f^4 + 1 + \gamma) > 0$  are satisfied, then the time derivative of  $V_0$  can be given as:

$$\dot{V}_0(t) \leq -\eta\|\omega(t)\|^2 - \gamma\|\tilde{\alpha}(t)\|^2 + \varpi \quad (49)$$

where  $\eta = \zeta_d - 1$ ,  $\varpi = \frac{1}{2}\bar{\zeta}_p^2 + 3(\frac{1}{2}\lambda_J^2\bar{\sigma}_f^4 + 1 + \gamma)(\bar{\phi}^2 + \bar{\xi}^2)$ . Clearly, equation (49) indicates that  $\omega(t)$  and  $\tilde{\alpha}(t)$  are ultimately uniformly bounded in the set:

$$\Omega_0 = \left\{ (\omega, \tilde{\alpha}) \in \mathbb{R}^3 \times \mathbb{R}^3 \mid \|\omega\|^2 \leq \frac{\varpi}{\eta}, \|\tilde{\alpha}\|^2 \leq \frac{\varpi}{\gamma} \right\} \quad (50)$$

Furthermore, in view of the natural boundedness of the quaternion, one can conclude that the system state  $(\mathbf{Q}_e, \omega)$  is ultimately uniformly bounded. ■

Theorem 1 implies the existence of the ultimate bound of system states  $(\mathbf{Q}_e, \omega)$ . However, because  $\varpi$  is an unknown constant, the specific size of the ultimate bound is hard to know. Next, we will make a step further to quantify the ultimate bound of system states  $(\mathbf{Q}_e, \omega)$  by taking Proposition 1 into consideration.

<sup>2</sup>Generally, the unknown uncertainties will not vary with an infinite rate for on-orbit scenarios, thus  $\|\Psi(t)\|$  has an upper bound. Moreover, according to the definition of  $\Phi$ , if the condition of persistent excitation is satisfied (which is mild for attitude dynamics), then  $\|\Phi(t)\|$  is also bounded.

**THEOREM 2** Consider that the combined spacecraft (7) which satisfies Conditions 1-3 and Assumption 1. Under the proposed GP-based learning control law (41), satisfying Condition 4 and where the unknown function  $\Delta$  is modeled by a multi-dimensional GP (11) that is recursively online updated according to Algorithm 1,  $\mathbf{Q}_e$  and  $\omega$  are guaranteed to be ultimately uniformly bounded with a probability of  $(1 - \delta)^3$ .

**PROOF** Consider the Lyapunov function candidate as:

$$\begin{aligned}V_1 &= (\zeta_p(\check{k}_p, \Sigma_\Delta) + \nu\zeta_d(\check{k}_d, \Sigma_\Delta))((1 - q_{e0})^2 + \mathbf{q}_e^\top\mathbf{q}_e) \\ &\quad + \frac{1}{2}\omega^\top\mathbf{J}_{c0}\omega + \nu\mathbf{q}_e^\top\mathbf{J}_{c0}\omega,\end{aligned}\quad (51)$$

where  $\nu > 0$  is a constant. The proof for the positive-definiteness of  $V_1$  can be found in [40].

By differentiating  $V_1$  and employing the closed-loop system and controller, one can derive:

$$\begin{aligned}\dot{V}_1 &= (\zeta_p(\check{k}_p, \Sigma_\Delta) + \nu\zeta_d(\check{k}_d, \Sigma_\Delta))\mathbf{q}_e^\top\omega \\ &\quad + (\omega + \nu\mathbf{q}_e)^\top(-\zeta_p(\check{k}_p, \Sigma_\Delta)\mathbf{q}_e - \zeta_d(\check{k}_d, \Sigma_\Delta)\omega \\ &\quad - \mathbf{J}_{c0}\mu_{\Delta,k}(\tilde{\mathbf{x}}) + \mathbf{d}(\tilde{\mathbf{x}})) + \frac{1}{2}\nu\omega^\top(q_{e0}\mathbf{I}_3 + \mathbf{q}_e^\times)\mathbf{J}_{c0}\omega\end{aligned}\quad (52)$$

According to Cauchy-Schwartz inequality and  $\|q_{e0}\mathbf{I}_3 + \mathbf{q}_e^\times\| = 1$ :

$$\begin{aligned}\dot{V}_1 &= -\nu\zeta_p(\check{k}_p, \Sigma_\Delta)\|\mathbf{q}_e\|^2 - \zeta_d(\check{k}_d, \Sigma_\Delta)\|\omega\|^2 \\ &\quad + \frac{1}{2}\nu\omega^\top(q_{e0}\mathbf{I}_3 + \mathbf{q}_e^\times)\mathbf{J}_{c0}\omega \\ &\quad + (\omega + \nu\mathbf{q}_e)^\top(\mathbf{d}(\tilde{\mathbf{x}}) - \mathbf{J}_{c0}\mu_{\Delta,k}(\tilde{\mathbf{x}})) \\ &\leq -\nu\zeta_p\|\mathbf{q}_e\|^2 - \lambda_J(\zeta_d - \frac{1}{2}\nu)\|\omega\|^2 \\ &\quad + (\omega + \nu\mathbf{q}_e)^\top(\mathbf{d}(\tilde{\mathbf{x}}) - \mathbf{J}_{c0}\mu_{\Delta,k}(\tilde{\mathbf{x}}))\end{aligned}\quad (53)$$

Further combining Proposition 1 with (53), it can be deduced that:

$$\begin{aligned}\mathbb{P}\{\dot{V}_1 \leq -\lambda_{\min}(\mathbf{M}_l)\|\mathbf{x}\|^2 + \lambda_J(\|\omega\| + \nu\|\mathbf{q}_e\|)\|\beta\|\|\Sigma_\Delta^{1/2}\|\} \\ \geq (1 - \delta)^3\end{aligned}\quad (54)$$

holds for all  $t \in \mathbb{R}_0^+$  where

$$\mathbf{M}_l = \begin{bmatrix} \nu\zeta_p & 0 \\ 0 & \zeta_d - \frac{1}{2}\nu\lambda_J \end{bmatrix}. \quad (55)$$

From the definition of  $V_1$ , we have:

$$\begin{aligned}V_1 &\leq \mathbf{x}^\top\mathbf{M}_s\mathbf{x} \leq \lambda_{\max}(\mathbf{M}_s)\|\mathbf{x}\|^2, \\ \mathbf{M}_s &= \begin{bmatrix} 2(\bar{\zeta}_p + \nu\bar{\zeta}_d) & \frac{1}{2}\nu\lambda_J \\ \frac{1}{2}\nu\lambda_J & \frac{1}{2}\lambda_J \end{bmatrix}\end{aligned}\quad (56)$$

Moreover, we can derive the boundedness of  $\mathbf{q}_e$  and  $\omega$ , which indicates that:

$$\|\mathbf{q}_e\| \leq \sqrt{\frac{V_1}{\bar{\zeta}_p + \nu\bar{\zeta}_d}}, \quad \|\omega\| \leq \sqrt{\frac{2V_1}{\lambda_c}}. \quad (57)$$

Consequently, from (54), (56) and (57), one can readily derive:

$$\mathbb{P}\left\{\dot{V}_1 \leq -\frac{\lambda_{\min}(\mathbf{M}_l)}{\lambda_{\max}(\mathbf{M}_s)}V_1 + \lambda_J \left( \sqrt{\frac{2V_1}{\lambda_c}} + \nu \sqrt{\frac{V_1}{\bar{\zeta}_p + \nu\bar{\zeta}_d}} \right) \varepsilon \right\} \geq (1-\delta)^3, \quad (58)$$

holds for all  $t \in \mathbb{R}_0^+$  where  $\varepsilon \geq \|\beta\| \|\Sigma_\Delta^{1/2}\|$ .

Next, we will take a step further to estimate the ultimately uniform bound for attitude error  $\mathbf{q}_e$  and angular velocity  $\boldsymbol{\omega}$ . Defining  $W = \sqrt{V_1}$ , we have that  $\dot{W} = \dot{V}_1/(2\sqrt{V_1})$  holds when  $V_1 \neq 0$ , that is:

$$\mathbb{P}\left\{\dot{W} \leq -\frac{\lambda_{\min}(\mathbf{M}_l)}{2\lambda_{\max}(\mathbf{M}_s)}W + \frac{\lambda_J\varepsilon}{\sqrt{2}}\vartheta\right\} \geq (1-\delta)^3 \quad (59)$$

with  $\vartheta = \left(\sqrt{\frac{1}{2\lambda_c}} + \nu\sqrt{\frac{1}{2(\bar{\zeta}_p + \nu\bar{\zeta}_d)}}\right)$ . Integrating the inequality in (59) leads to:

$$0 \leq \sqrt{V_1(t)} \leq \left(\sqrt{V_1(0)} - s_1\right)e^{-s_2t} + s_1 \quad (60)$$

where

$$s_1 = \frac{\sqrt{2}\lambda_J\varepsilon\vartheta\lambda_{\max}(\mathbf{M}_s)}{\lambda_{\min}(\mathbf{M}_l)}, \quad s_2 = \frac{\lambda_{\min}(\mathbf{M}_l)}{2\lambda_{\max}(\mathbf{M}_s)}.$$

Thus, when  $t \rightarrow \infty$ , one can derive:

$$\limsup_{t \rightarrow \infty} \sqrt{V_1(t)} \leq s_1 \quad (61)$$

Recalling (57), we can finally compute the bound of  $\mathbf{q}_e$  and  $\boldsymbol{\omega}$  that:

$$\limsup_{t \rightarrow \infty} \|\mathbf{q}_e(t)\| = \frac{s_1}{\sqrt{\bar{\zeta}_p + \nu\bar{\zeta}_d}}, \quad (62a)$$

$$\limsup_{t \rightarrow \infty} \|\boldsymbol{\omega}(t)\| = \sqrt{\frac{2}{\lambda_c}}s_1. \quad (62b)$$

Furthermore, due to  $2(\bar{\zeta}_p + \nu\bar{\zeta}_d)(1 - q_{e0}) \leq V_1$ , we have:

$$\begin{aligned} \limsup_{t \rightarrow \infty} \|q_{e0}(t)\| &\geq 1 - \frac{\lim_{t \rightarrow \infty} \sup V_1(t)}{2(\bar{\zeta}_p + \nu\bar{\zeta}_d)} \\ &\geq 1 - \frac{s_1^2}{2(\bar{\zeta}_p + \nu\bar{\zeta}_d)}. \end{aligned} \quad (63)$$

Then,

$$\begin{aligned} \limsup_{t \rightarrow \infty} \|\mathbf{q}_e(t)\| &= \limsup_{t \rightarrow \infty} \sqrt{1 - q_{e0}^2} \\ &\leq \sqrt{1 - \left(1 - \frac{s_1^2}{2(\bar{\zeta}_p + \nu\bar{\zeta}_d)}\right)^2}. \end{aligned} \quad (64)$$

Comparing (64) and (62a), it can be concluded that the upper bound for  $\mathbf{q}_e$  is further reduced. Hence, the system states  $\mathbf{q}_e$  and  $\boldsymbol{\omega}$  ultimately converge to the compact set  $\Omega_1$  and  $\Omega_2$  within the probability  $(1-\delta)^3$ :

$$\begin{aligned} \Omega_1 &= \left\{ \mathbf{Q}_e \in \mathbb{Q}^3 \mid \|\mathbf{q}_e\| \leq \sqrt{1 - \left(1 - \frac{s_1^2}{2(\bar{\zeta}_p + \nu\bar{\zeta}_d)}\right)^2} \right\} \\ \Omega_2 &= \left\{ \boldsymbol{\omega} \in \mathbb{R}^3 \mid \|\boldsymbol{\omega}\| \leq \sqrt{\frac{2}{\lambda_c}}s_1 \right\} \end{aligned}$$

This completes the proof.  $\blacksquare$

The block diagram of the closed-loop control system is shown in Fig. 3.

**REMARK 5** Compared to the existing attitude takeover controllers for the combined spacecraft, such as NN-based compensation [16] and adaptive control in [13], the advantages of the proposed GP-based online learning controller are as follows:

1) As a nonparametric modeling approach, the predictive outputs of GP are inherently probabilistic. The GP variance quantifies the confidence level of the predicted uncertainty, while the mean corresponds to the estimated uncertainty. Both the mean and the variance are incorporated into the control scheme, where the latter effectively improves the robustness of the algorithm. To the best of the authors' knowledge, existing results for the uncertainty quantification of NN mainly include Monte Carlo methods, dropout, and Bayesian ANNs. These approaches are generally complex and computationally too demanding to be applied in a real implementation on a spacecraft.

2) In contrast to the manual time-consuming tuning process for the hyperparameters of the NN-based adaptive control scheme, the hyperparameters of the GP scheme, such as  $\boldsymbol{\theta}_j = \text{vec}(\sigma_{\epsilon,j}^2, \sigma_{\tilde{\epsilon},j}^2, \lambda_{j,1}, \dots, \lambda_{j,9})$ , are automatically tuned by marginalized likelihood based optimization, while the remaining parameters of the adaptive feedback controller, such as  $k_p$  and  $k_d$ , can be designed by practical experience with PD control design for such systems with known nominal inertia matrix  $\mathbf{J}_{c0}$  and does not require extensive tuning procedures.

3) As a data-driven adaptive approach, the proposed GP-based learning controller only needs an initial "rough" model of the unknown dynamics that is learnt with only a small amount of data and needs no extensive exploration of the entire state space before safe execution of the mission. Subsequently, the initial GP model is updated online as new operational data becomes available to ensure continuous adaption and performance improvement of the controller.

## V. Simulation Results

In this section, simulation results under various on-orbit scenarios are presented to illustrate the effectiveness of our proposed control strategy.

### A. Simulation Platform

A SimMechanics-based, high-fidelity simulator has been developed for the combined spacecraft to accurately characterize its attitude motion under the considered model uncertainties and target maneuvering, and act as the physical system and *data-generator* for the proposed online learning control strategy. As shown in Fig. 4, the system consists of two components, namely the servicer and the target. The servicer, simplified as a cubesat, is equipped with two robotic arms on each side for capturing the

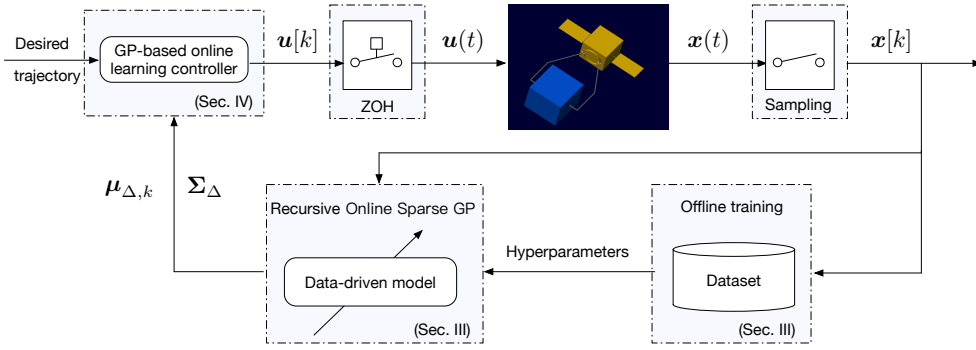


Fig. 3: Block diagram of the proposed control strategy.

TABLE I: Physical parameters of the combined spacecraft

		Parameter	Value
Servicer	Body	Size (m)	$2 \times 2 \times 2$
		Inertia ( $\text{kg} \cdot \text{m}^2$ )	$\text{diag}(405, 405, 405)$
		Mass (kg)	1080
	Robotic Arm	Length per link (m)	0.73, 1, 2
		Mass per link (kg)	6
		Inertia per link ( $\text{kg} \cdot \text{m}^2$ )	$\text{diag}(0.03, 0.73, 0.73)$
Target	Size (m)	$2 \times 2 \times 2$	
	Body	Inertia ( $\text{kg} \cdot \text{m}^2$ )	$\text{diag}(36.8, 37.5, 36.8)$
		Mass (kg)	75
	Others	Size of solar panels (m)	$1.5 \times 0.8 \times 0.01$
Size of docking ring (m)		0.423/0.443 (I/O)	

target. Each arm is composed of three segments, and once capture is completed, the joints are locked in a specific configuration. The target is also modeled as a cube with solar panels on both sides and a capture interface in the form of a docking ring. For the simplicity of the study, each component in the system is considered as a rigid body without flexibility.

The simulation platform can be seen as a black-box system, i.e. the attitude motion equations in the analytical form are “packaged” within Simulink, and only the I/O ports are used for the proposed adaptive control algorithm, just as it would be the case for a real spacecraft. The physical parameters considered in the simulator are given in Table. I, which are mainly based on [41]. Note that appropriate modifications have been done to make this scenario more suitable for implementation in Matlab/SimMechanics.

The studied simulation scenario is as follows. The target is a partially malfunctioning satellite working in a sun-oriented mode. Thus, after being captured by the servicer, the target attempts to perform active attitude maneuvers throughout the takeover task, i.e., when the servicer applies a control torque to the target that leads to a deviation from its initial attitude orientation, the target will generate a competitive torque against the control torque to keep its attitude. The unknown model uncertainties considered in this example consist of two parts, the unknown dynamics corresponding to additional robot arms,

flexible solar panels, etc., and the additional dynamics caused by the target-generated torque. The active attitude control law for the target is chosen to be of the PD form:

$$\mathbf{u}_t = -\mathbf{K}_{\text{pt}}\mathbf{q}_{\text{et}} - \mathbf{K}_{\text{dt}}\boldsymbol{\omega}_t \quad (65)$$

where the subscript “t” denotes the target-related variables,  $\mathbf{K}_{\text{pt}} = 0.02\mathbf{J}_t$ ,  $\mathbf{K}_{\text{dt}} = 0.05\mathbf{J}_t$ ,  $\mathbf{q}_{\text{et}} = \mathbf{q}_{\text{dt}}^{-1} \otimes \mathbf{q}_t$ ,  $\mathbf{q}_{\text{dt}}$  is the vector part of initial quaternion of the target relative to the inertial coordinate system  $\mathcal{F}_I$ . Note that (65) is part of the black-box simulation  $\mathcal{F}_I$  and it is not known by the servicer.

## B. Adaptive Control Under Unknown Model Uncertainties

After docking, the initial Euler angle for the combined spacecraft is  $[15 \ 5 \ -20]^\circ$ , and the initial angular velocity is  $\boldsymbol{\omega}_0 = [0.01 \ 0.02 \ -0.01]^\top$  rad/s. In this simulation scenario, the attitude orientation of the combined spacecraft is forced to converge to  $\mathbf{Q}_d = [1 \ 0 \ 0]^\top$ , which corresponds to an attitude stabilization task. Generally, there are two options to construct the training data set  $\mathcal{D}_N$ : Either just using the transient data of the closed-loop system or applying excitation torque on the spacecraft. In this paper, only the transient data regarding orientation change with a baseline PD controller is collected, which represents a low-profile scenario. That is, the training data is collected during the control process in the first 50s of the simulation. With a sampling frequency of 10 Hz, the size of the collected training set is  $N = 500$ . Considering the sensor-based measurement of  $\mathbf{q}_e$  and  $\boldsymbol{\omega}$ , the training outputs are corrupted by a Gaussian white noise  $\epsilon[k]$  with  $\epsilon[k] \sim \mathcal{N}(0, 0.05)$ . The feedback gain for the baseline controller follows  $\mathbf{K}_{\text{p}0} = 0.1\mathbf{J}_{\text{c}0}$ ,  $\mathbf{K}_{\text{d}0} = 0.3\mathbf{J}_{\text{c}0}$ , where the nominal inertia matrix of the combined spacecraft is selected as  $\mathbf{J}_{\text{c}0} = \text{diag}(600, 450, 600)$   $\text{kg} \cdot \text{m}^2$ . It is worth mentioning that the value of  $\mathbf{J}_{\text{c}0}$  is simply selected based on the known inertia matrix of the servicer.

For the obtained data set  $\mathcal{D}_N$ , both a sparse and a standard GP are trained, of which the hyperparameters of both are optimized by the conjugated gradient descent algorithm. Specifically, in the case of ROSGP, the inducing points are initialized randomly inside  $\mathcal{D}_N$  with the size of  $M = 50$  and the initial  $\boldsymbol{\alpha}[0]$  is computed according

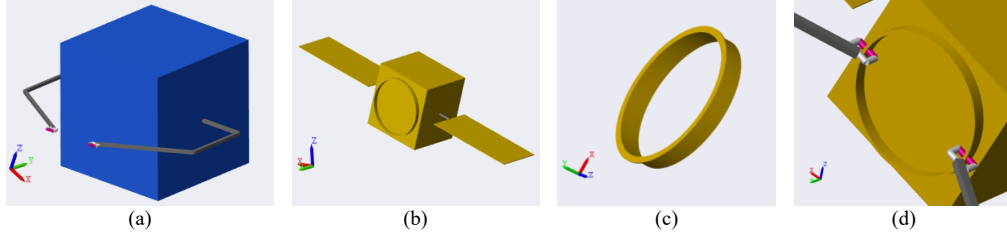


Fig. 4: Parts the combined spacecraft assembly: (a) Servicer spacecraft, (b) Target spacecraft, (c) Docking ring on the target, (d) Capture point.

to (25a). The initial value of  $P$  is set to  $P[0] = 10^2 I_M$ . Then, at  $t = 50$ s, the controller is updated to (41) where the recursive update routine is activated. The feedback gain functions  $\zeta_p(k_p, \Sigma_\Delta)$  and  $\zeta_d(k_d, \Sigma_\Delta)$  are selected as linear functions of the variance:  $\zeta_p(k_p, \Sigma_\Delta) = J_{c0}(k_p + 0.1\Sigma_\Delta^{1/2})$  and  $\zeta_d(k_d, \Sigma_\Delta) = J_{c0}(k_d + 0.2\Sigma_\Delta^{1/2})$  with  $k_p = 0.02$  and  $k_d = 0.05$ . It can be easily verified that  $\zeta_p(\cdot)$  and  $\zeta_d(\cdot)$  satisfy Condition 4 because the GP predictive variance is bounded by  $\sup k_{**j}$  for  $j \in \mathbb{I}_1^3$ . Furthermore, the selection of parameters  $k_p$  and  $k_d$  is based on practical experience with PD control design for such systems with known nominal inertia matrix  $J_{c0}$ , and does not require extensive tuning procedures.

The proposed recursive online sparse GP-based learning controller is compared to:

- 1) Baseline PD controller: This is the initial controller to generate the training data set during the first 50s, and it is kept fixed after  $t = 50$ s.
- 2) Standard GP-based controller: This controller keeps the same structure and parameters as the proposed controller (41), but it is based only on an initial trained GP without the recursive online update strategy.

The trajectories of the system states, control inputs, together with the feedback gain are depicted in Figs. 5-6, where the green line indicates that the controller updates at 50s. Simulation results show that all three controllers succeed in achieving the attitude stabilization task for the combined spacecraft. Also, as shown in Fig. 6, the feedback gain of baseline PD remains constant during the entire task, and the two GP-based controllers can adapt its gain with the predictive variance. It is worth mentioning that the ROSGP performs the best in terms of final pointing error, because the unknown function  $\check{\Delta}(\tilde{x})$  in the combined spacecraft dynamics is real-time compensated by the predictive mean  $\mu_\Delta(\tilde{x})$  of the trained GP model and further completed by the ROSGP algorithm with online adaption. However, it is more cautious hence its settling time is slightly slower which can be seen in terms of the slower convergence of the angular velocity. Additionally, while the baseline PD controller remains the largest steady-state error of  $q_e$  without the GP compensation, it seems to achieve the smallest steady-state error of  $\omega$ , nonetheless, at the expense of high-gain feedback. This can also be

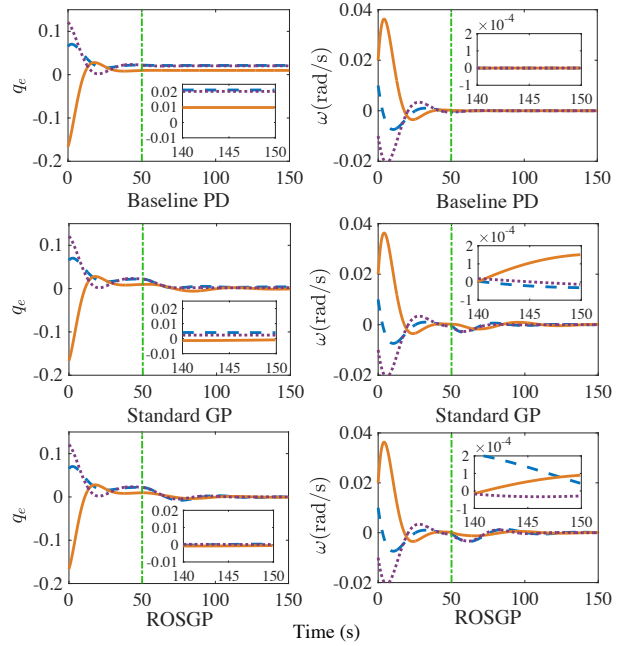


Fig. 5: Trajectories of  $q_e$  and  $\omega$  under the considered controllers. The lines in each subplot describe the three components of these signals. It can be observed that the ROSGP performs the best in terms of the steady-error of  $q_e$  due to its capability of online GP compensation, while the baseline PD controller achieves the smallest steady-state error in terms of  $\omega$  at the expense of high-gain feedback.

seen in Fig. 7, where the Pareto-front-like figure shows the relationship between the normalized feedback gain  $\sqrt{\|\zeta_p(\cdot)\|^2 + \|\zeta_d(\cdot)\|^2}$  and system error  $\sqrt{\|q_e\|^2 + \|\omega\|^2}$  of the two controllers in 30 simulation cases. The color indicates the control efforts  $\int_{50}^{150} \|u\| dt$  during the task. It is clear that the proposed ROSGP-based controller requires a relatively modest feedback gain at the same level of system error.

The absolute value of the two GP estimation errors  $\|\mu_{\Delta,j} - \check{\Delta}_j\|$  is depicted in Fig. 8. One can see that there exists a significant estimation error between the predictive mean of standard GP and true function due to the initial data set  $\mathcal{D}_N$  collected in the first 50s is not sufficient to describe the whole state space. In contrast, the model accuracy is adaptively enhanced by the proposed ROSGP,

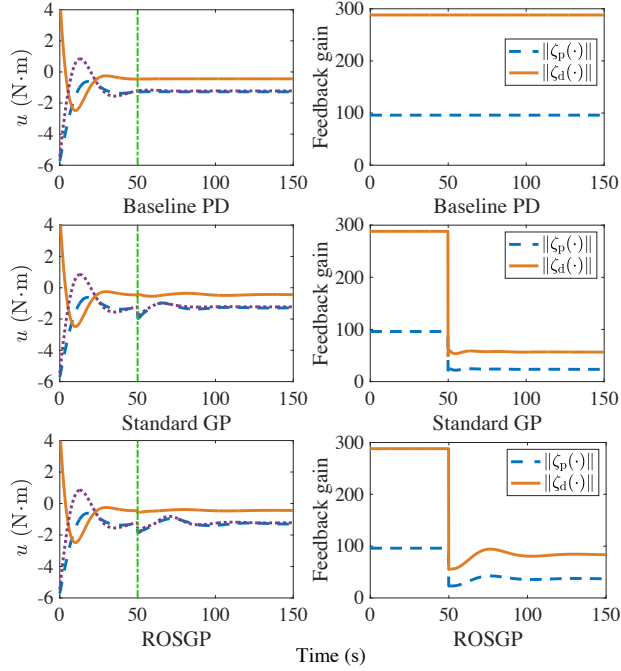


Fig. 6: Control inputs  $u$  and the norm of feedback gain under the considered controllers. It can be observed that the high-gain feedback of the baseline PD controller remains fixed throughout the entire task, while the GP-based controllers can adaptively adjust the feedback gain according to the GP predictive variance, keeping it at a low level while ensuring pointing accuracy.

and the full compensation for the unknown function  $\check{\Delta}(\tilde{x})$  is achieved.

### C. Adaptive Control Under Re-maneuver Scenario

During the on-orbit takeover control tasks, the combined spacecraft often needs to perform additional new tasks after attitude stabilization, such as maneuvering to a new attitude orientation for awaiting further missions. This new attitude orientation may be far from the initial training data set  $\mathcal{D}_N$ , which poses a significant challenge for the GP-based learning control strategy. Therefore, it is essential to verify the generalization performance and control effectiveness of the proposed GP-based online learning control strategy in untrained areas.

Consider the following scenario: On the basis of the effective attitude stabilization in the first 150s, the combined spacecraft is required to re-maneuver to a new attitude orientation  $\mathbf{Q}_d = [0.899 \ -0.30 \ 0.20 \ -0.10]^T$ . Meanwhile, to further illustrate the superior performance of the proposed ROSGP-based control strategy, rougher conditions are considered in this scenario. In addition to larger uncertainties resulting from the re-orientation outside the trained area and competitive torque of the target maneuver, large time-varying external disturbances that may result from the movement of an extra robotic arm, fuel sloshing, and flexible vibrations of solar panels, etc., are applied

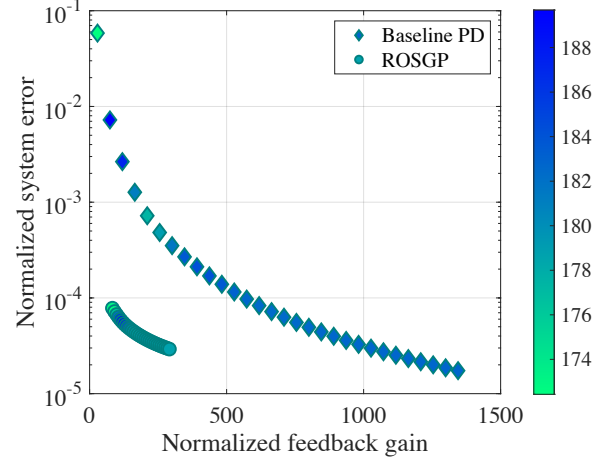


Fig. 7: A Pareto-front-like comparison of the normalized feedback gain and system error between the baseline PD and proposed ROSGP-based controllers. The color indicates the control efforts during the task. It is apparent that at the same level of system error, the ROSGP-based controller requires a relatively small feedback gain.

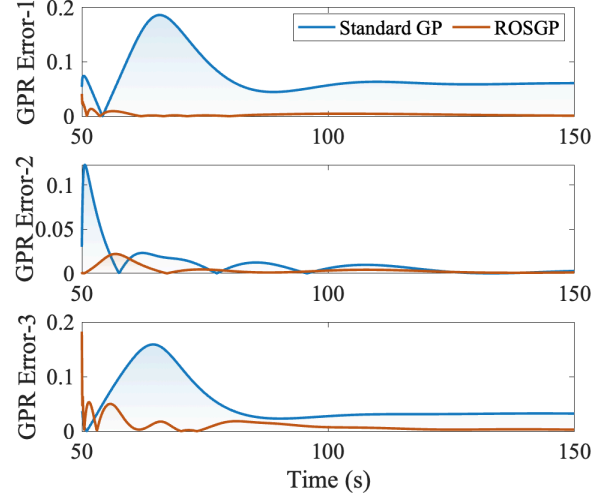


Fig. 8: Absolute value of the GP estimation error  $\|\mu_{\Delta,j} - \check{\Delta}_j\|$  for each dimension  $j$  evaluated over time under the stabilization scenario.

to the combined spacecraft from 150s, and can be modeled as  $\tau_d(t) = [0.5 \sin 0.1t, -\sin 0.15t, 1.5 \sin(-0.15t + 1.5)]^T \text{N} \cdot \text{m}$ .

To show the advantages of the proposed GP-based learning control strategy, the NN-based adaptive control scheme (denoted by ANN) in [16] is applied in the considered scenario. In this ANN approach, radial basis functions (RBF) are chosen as the activation functions to estimate and compensate the unknown uncertainties. The NN structure  $\hat{\Delta} = \hat{\Theta}^T \Phi_\rho$  used to estimate the unknown dynamics  $\check{\Delta}$  contains 10 neurons with centers  $c_i \in \mathbb{R}^6$  ( $i \in \mathbb{I}_1^{10}$ ) randomly distributed in  $[-2, 2]^6$  and widths  $\sigma_i = 2$ . Each element of the initial NN weights  $\hat{\Theta}(0)$  is

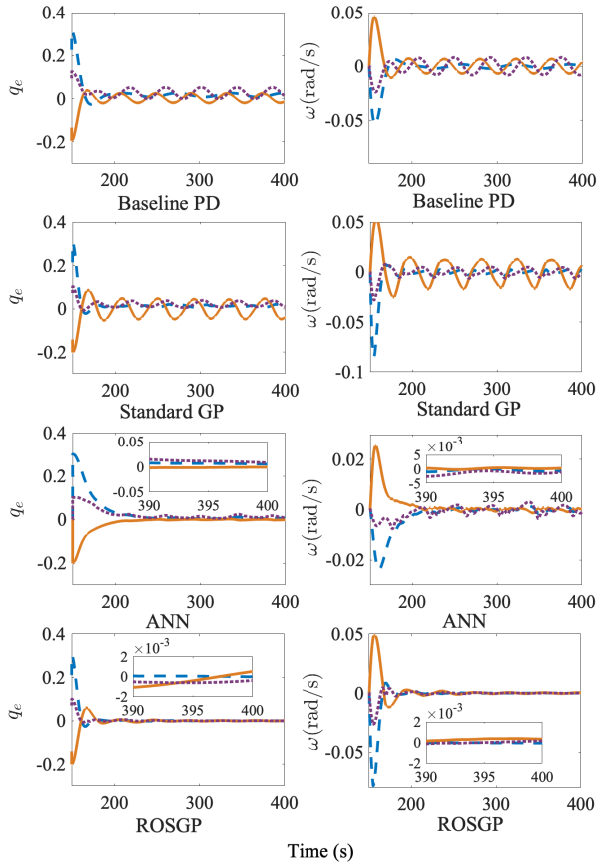


Fig. 9: Trajectories of  $q_e$  and  $\omega$  under the considered controllers and additional torque disturbances. Due to the torque disturbances, only ROSGP can achieve reasonably low steady-state error.

chosen between  $\pm 0.5$  randomly. The adaptive law for the NN weight is given by:  $\dot{\hat{\Theta}} = F_\rho \Phi_\rho r^\top - k_\rho F_\rho \|\hat{r}\| \hat{\Theta}$  where  $r = \omega + \epsilon q_e$ . The parameters for the ANN controller are selected as  $F_\rho = 200I_3$ ,  $k_\rho = 1$ ,  $\Lambda = 0.1I_3$ ,  $\epsilon = 0.1$ . It should be emphasized that the choice of all the aforementioned parameters in the ANN method is required to be tuned manually, for which there are no comprehensive tuning rules. For a fair comparison, the previously used control scenario was used where the GP-based feedforward has been substituted by the ANN compensation term.

The trajectories of the system states, control inputs, together with the feedback gain under these settings are depicted in Figs. 9-10. It can be seen that under the additional disturbances, both the dynamic response and steady-state error of the baseline PD and standard GP-based schemes are obviously unsatisfactory and fail to sufficiently well perform the attitude re-orientation task, resulting in an oscillating behavior around the equilibrium points. The reason is that the high-gain feedback of the baseline PD controller is not capable of dealing with such large uncertainties anymore, which also cannot be precisely captured by the offline-trained standard GP model. The performance of the proposed ROSGP-

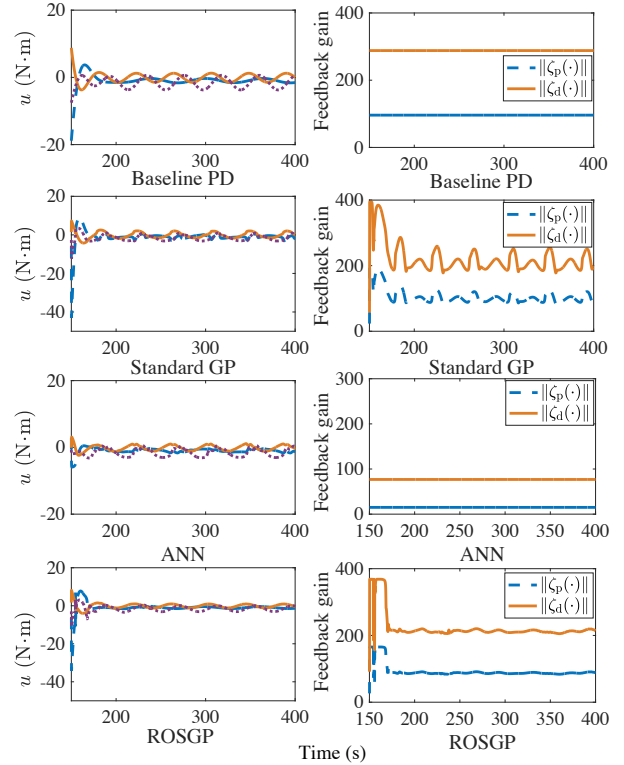


Fig. 10: Control inputs  $u$  and the norm of the feedback gain under the considered controllers. One can observe that the feedback gain of GP-based controllers can vary according to the predictive variance of GP in the untrained areas.

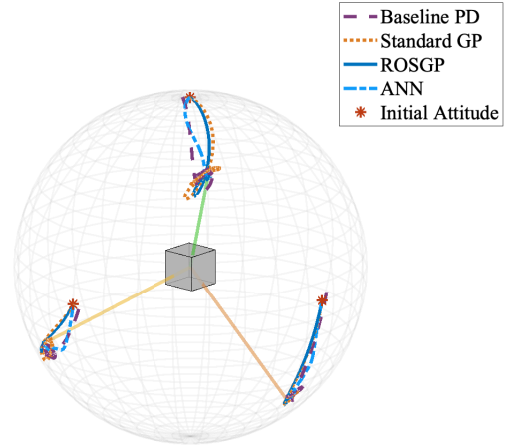


Fig. 11: 3-D trajectories of the spacecraft under the considered controllers.

based controller is significantly better compared to the baselines, mainly resulting from the online adaptation of the GP compensation strategy. Particularly, compared with ANN method, the system states converge faster under the proposed ROSGP-based controller because of the adaptive feedback gains with GP predictive variance. A 3D illustration of the trajectories of the axes of  $\mathcal{F}_B$  is shown in Fig. 11, where the gray cube in the center represents the combined spacecraft. One can observe that the trajectories under the baseline PD controller show an

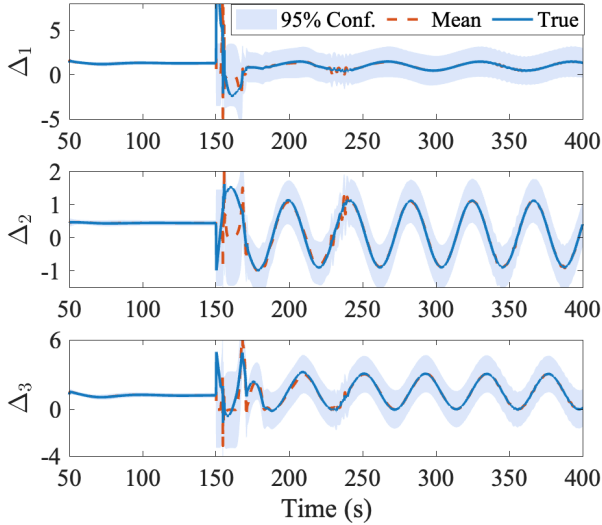


Fig. 12: Value of the true uncertain dynamics (given by blue) for each angular acceleration, the mean of the learnt ROSGP (given by red), and the 95% confidence interval (given by shaded blue) evaluated at the true state values at time  $t$  when re-maneuvers to the untrained area.

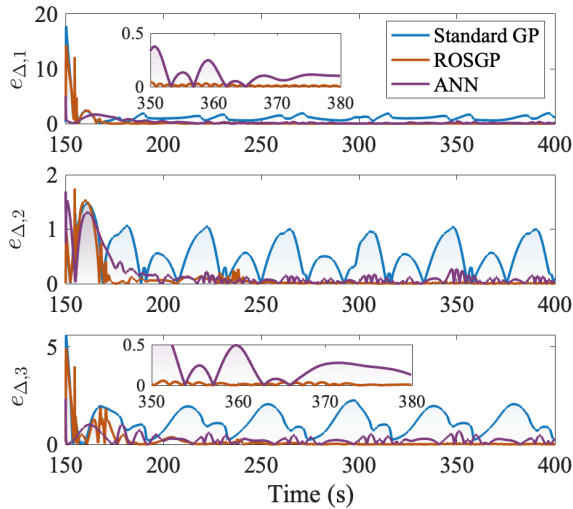


Fig. 13: Absolute value of the estimation error  $e_{\Delta,j}$  for each dimension  $j$  under the considered controllers evaluated over time under the re-orientation scenario.

obvious limit cycle behavior around the desired orientation. This phenomenon can be further demonstrated by Fig. 12.

As shown in Fig. 12, the predictive variance (depicted by a shaded area of 95% confidence interval) keeps at a low level from 50s to 150s during the stabilization task. Next, the predictive variance increases significantly from 150s when the combined spacecraft re-maneuvers to an area outside the training data set. This makes it so that the feedback gain of the two GP-based learning controllers appropriately increases to further mitigate the suddenly appearing time-varying disturbance.

Figure 13 shows the absolute value of the estimation error  $e_{\Delta,j} = \|\mu_{\Delta,j} - \hat{\Delta}_j\|$  of the two GPs and  $e_{\Delta,j} = \|\hat{\Delta}_j - \Delta_j\|$  of the ANN starting from 150s, i.e., the re-maneuver task begins. It can be observed that despite the presence of target attitude maneuvers, external time-varying disturbances, and unlearned dynamics, the ROSGP still has the capability of online learning of complicated time-varying uncertainties and compensating for them. Therefore, despite realizing an attitude re-orientation in a previously unseen area of the state space, the set point error can always be maintained at a low level with superior transient performance. It should be noted that the ANN method results in a comparable estimation error w.r.t. the proposed ROSGP, but this comes at the cost of a significant manual tuning of the hyperparameters of the ANN scheme.

In addition, the computation time for a 400s long simulation (on a MacOS Monterey, 10-core M1 Pro, 16GB RAM) under different controllers is illustrated as follows. Particularly, the time costs of the standard GP and ROSGP-based controller during model training is 5.59s and 2.71s while during task execution of a 400s long trajectory. The average computation time of the standard GP, ROSGP, and ANN-based controller per control cycle is 16.36ms, 3.31ms, and 3.99ms (which are realizable under the sampling time  $T_s = 100\text{ms}$ ), respectively. This shows that the computational load of the proposed control strategy is quite small, and does not require extensive parameter tuning procedures. Thus, the results are consistent with the expected performance of the control system design, indicating that the proposed strategy can efficiently, yet accurately learn the unknown function online, and achieve high-precision control of the attitude takeover task.

#### D. Monte Carlo Simulation

Finally, Monte Carlo simulation is conducted to comprehensively analyze the effectiveness and generalization of the proposed control strategy under different physical parameter values and user chosen controller parameters. Table. II presents the range of the selected random parameters in the Monte Carlo simulation.

TABLE II: Randomized parameter ranges in the Monte Carlo study.

Randomized Parameters	Range
$m_t$ , kg	[50,100]
$J_t$ , kg·m <sup>2</sup>	diag[50 ± 50, 50 ± 50, 50 ± 50]
$Q_0$	$\mathbb{Q}^3$
$\omega_0$ , rad/s	$[-0.1,0.1] \times [-0.1,0.1] \times [-0.1,0.1]$
Baseline $K_p$	$[0.08, 0.25]J_{c0}$
Baseline $K_d$	$[0.28, 0.55]J_{c0}$
$P[0]$	$[1, 1000]I_M$
$K_{pt}$	$[0.01, 0.3]J_t$
$K_{dt}$	$[0.02, 0.8]J_t$
Amplitude of $\tau_d(t)$	$[-0.5,0.5] \times [-0.5,0.5] \times [-0.5,0.5]$

Figure 14 presents the distributions of the steady-state error and *mean square error* (MSE) of  $q_e$  and  $\omega$  from 500 Monte Carlo simulations, respectively, where each



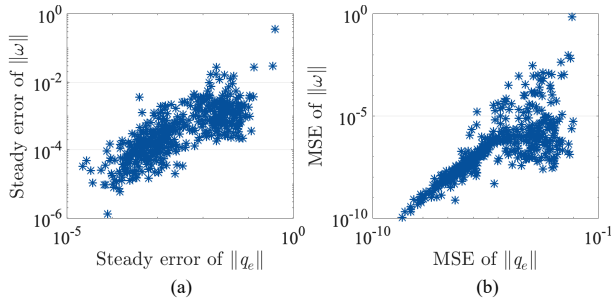


Fig. 14: Distributions of (a) steady-state error and (b) MSE of  $q_e$  and  $\omega$ .

“\*” represents a simulation case under random conditions. It can be observed that the overwhelming majority of the simulation cases correspond to good performance, where the steady-state error of the attitude quaternion is less than  $1 \times 10^{-2}$  and that of attitude angular velocity is less than  $1 \times 10^{-2}$  rad/s. Additionally, the MSE of attitude quaternion is less than  $1 \times 10^{-4}$ , and that of attitude angular velocity is less than  $1 \times 10^{-4}$  rad/s. The results illustrate the generalization performance of the proposed ROSGP-based controller.

In summary, the simulation results show that the proposed GP-based online learning control strategy presented in (41) and Algorithm 1 is highly effective and its implementation is feasible for OOS attitude takeover tasks even in the presence of target attitude maneuverability.

## VI. CONCLUSION

This paper presents an effective GP-based online learning control strategy for attitude takeover control of noncooperative targets with attitude maneuverability. A novel recursive online sparse GP algorithm is introduced to ensure the successive online learning of the unknown dynamics, while the computational load is kept low, making the approach well applicable in resource-constrained onboard scenarios. The proposed method has probabilistic guarantees for a user-defined bound of the pointing error. The introduced approach provides new perspectives into the attitude controller design of spacecrafts with unknown dynamics, especially in cases where the moment of inertia cannot be identified. The properties and effectiveness of our proposed strategy have been analyzed and demonstrated by numerical simulations based on a high-fidelity simulator.

## REFERENCES

- [1] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich  
A review of space robotics technologies for on-orbit servicing  
*Prog. Aerosp. Sci.*, vol. 68, pp. 1–26, 2014.
- [2] D. Pinard, S. Reynaud, P. Delpy, and S. E. Strandmoe  
Accurate and autonomous navigation for the ATV  
*Aerosp. Sci. Technol.*, vol. 11, no. 6, pp. 490–498, 2007.
- [3] H. Liu, Z. Li, Y. Liu, M. Jin, F. Ni, and Y. Liu

- Key technologies of TianGong-2 robotic hand and its on-orbit experiments  
*Sci. Sin. Technol.*, vol. 48, no. 12, pp. 1313–1320, 2018.
- [4] E. Bergmann, B. K. Walker, and D. R. Levy  
Mass property estimation for control of asymmetrical satellites  
*J. Guid., Control, Dyn.*, vol. 10, no. 5, pp. 483–491, 1987.
- [5] Y. Murotsu, K. Senda, M. Ozaki, and S. Tsujio  
Parameter identification of unknown object handled by free-flying space robot  
*J. Guid., Control, Dyn.*, vol. 17, no. 3, pp. 488–494, 1994.
- [6] O. Ma, H. Dang, and K. Pham  
On-orbit identification of inertia properties of spacecraft using a robotic arm  
*J. Guid., Control, Dyn.*, vol. 31, no. 6, pp. 1761–1771, 2008.
- [7] O.-O. Christidi-Loumpasefski and E. Papadopoulos  
On the parameter identification of free-flying space manipulator systems  
*Robot. Auton. Syst.*, vol. 160, p. 104310, 2023.
- [8] O.-O. Christidi-Loumpasefski, G. Rekleitis, E. Papadopoulos, and F. Ankersen  
On system identification of space manipulator systems including their fuel sloshing effects  
*IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2446–2453, 2023.
- [9] Q. Meng, J. Liang, and O. Ma  
Identification of all the inertial parameters of a non-cooperative object in orbit  
*Aerosp. Sci. Technol.*, vol. 91, pp. 571–582, 2019.
- [10] W. Chu, S. Wu, Z. Wu, and Y. Wang  
Least square based ensemble deep learning for inertia tensor identification of combined spacecraft  
*Aerosp. Sci. Technol.*, vol. 106, p. 106189, 2020.
- [11] P. Huang, D. Wang, Z. Meng, F. Zhang, and J. Guo  
Adaptive postcapture backstepping control for tumbling tethered space robot-target combination  
*J. Guid., Control, Dyn.*, vol. 39, no. 1, pp. 150–156, 2015.
- [12] B. Zhang, B. Liang, Z. Wang, Y. Mi, Y. Zhang, and Z. Chen  
Coordinated stabilization for space robot after capturing a noncooperative target with large inertia  
*Acta Astronaut.*, vol. 134, pp. 75–84, 2017.
- [13] G. Kang, J. Wu, C. Jin, and X. Chen  
Adaptive controller design for satellite attached by non-cooperative object  
*Chin. J. Aeronaut.*, vol. 33, no. 3, pp. 1006–1015, 2020.
- [14] C. Liu, X. Yue, and Z. Yang  
Are nonfragile controllers always better than fragile controllers in attitude control performance of post-capture flexible spacecraft?  
*Aerosp. Sci. Technol.*, vol. 118, p. 107053, 2021.
- [15] H. Leeghim, Y. Choi, and H. Bang  
Adaptive attitude control of spacecraft using neural networks  
*Acta Astronaut.*, vol. 64, no. 7, pp. 778–786, 2009.
- [16] P. Huang, D. Wang, Z. Meng, F. Zhang, and Z. Liu  
Impact dynamic modeling and adaptive target capturing control for tethered space robots with uncertainties  
*IEEE/ASME Trans. Mechatronics*, vol. 21, no. 5, pp. 2260–2271, 2016.
- [17] K. Ning, B. Wu, and C. Xu  
Event-triggered adaptive fuzzy attitude takeover control of spacecraft  
*Adv. Space Res.*, vol. 67, no. 6, pp. 1761–1772, 2021.
- [18] C. Wei, J. Luo, H. Dai, and J. Yuan  
Adaptive model-free constrained control of postcapture flexible spacecraft: a Euler-Lagrange approach  
*J. Vib. Contr.*, vol. 24, no. 20, pp. 4885–4903, 2018.
- [19] J. Luo, Z. Yin, C. Wei, and J. Yuan  
Low-complexity prescribed performance control for spacecraft attitude stabilization and tracking  
*Aerosp. Sci. Technol.*, vol. 74, pp. 173–183, 2018.
- [20] Y. Fan and W. Jing

- Inertia-free appointed-time prescribed performance tracking control for space manipulator  
*Aerosp. Sci. Technol.*, p. 106896, 2021.
- [21] C. E. Rasmussen  
Gaussian processes in machine learning  
In *Advanced Lectures on Machine Learning*. Berlin, Germany: Springer, 2004, pp. 63–71.
- [22] T. Beckers, D. Kulić, and S. Hirche  
Stable Gaussian process based tracking control of Euler–Lagrange systems  
*Automatica*, vol. 103, pp. 390–397, 2019.
- [23] Y. Liu and R. Tóth  
Learning based model predictive control for quadcopters with dual Gaussian process  
In *Proc. IEEE Conf. Decis. Control (CDC)*. Austin, TX, USA, December 14–17, 2021, pp. 1515–1521.
- [24] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger  
Learning-based model predictive control for autonomous racing  
*IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3363–3370, 2019.
- [25] L. Csató and M. Opper  
Sparse on-line Gaussian processes  
*Neural Comput.*, vol. 14, no. 3, pp. 641–668, 2002.
- [26] R. C. Grande, G. Chowdhary, and J. P. How  
Experimental validation of bayesian nonparametric adaptive control using Gaussian processes  
*J. Aerosp. Inform. Syst.*, vol. 11, no. 9, pp. 565–578, 2014.
- [27] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela  
Bayesian nonparametric adaptive control using Gaussian processes  
*IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 537–550, 2014.
- [28] D. I. Ignatyev, H.-S. Shin, and A. Tsourdos  
Sparse online Gaussian process adaptation for incremental backstepping flight control  
*Aerosp. Sci. Technol.*, vol. 136, p. 108157, 2023.
- [29] D. Petelin and J. Kocijan  
Control system with evolving Gaussian process models  
In *Proc. IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)*, 2011, pp. 178–184.
- [30] J. Kocijan  
*Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.
- [31] M. Maiworm, D. Limon, and R. Findeisen  
Online learning-based model predictive control with gaussian process models and stability guarantees  
*Int. J. Robust Nonlinear Control*, vol. 31, no. 18, pp. 8785–8812, 2021.
- [32] G. Ma, Y. Liu, Y. Lyu, and P. Wang  
Learning-based attitude takeover control for noncooperative space targets with unknown dynamics  
In *Proc. IEEE Chin. Control Conf. (CCC)*. Shanghai, China, July, 2021, pp. 2233–2238.
- [33] B. Wie  
*Space Vehicle Dynamics and Control*. Reston, VA, USA: AIAA Education Series, 2008.
- [34] Z. Chen and B. Wang  
How priors of initial hyperparameters affect Gaussian process regression models  
*Neurocomputing*, vol. 275, pp. 1702–1710, 2018.
- [35] R. L. Burden, J. D. Faires, and A. M. Burden  
*Numerical analysis*. 10th ed. Boston, MA, USA: Cengage Learning, 2014.
- [36] E. Snelson and Z. Ghahramani  
Sparse Gaussian processes using pseudo-inputs  
*Adv. Neural Inf. Process. Syst.*, vol. 18, pp. 1257–1264, 2005.
- [37] J. Umlauf, L. Pöhler, and S. Hirche  
An uncertainty-based control Lyapunov approach for control-affine systems modeled by Gaussian process  
*IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 483–488, 2018.
- [38] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger  
Information-theoretic regret bounds for Gaussian process optimization in the bandit setting  
*IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.
- [39] J. Umlauf and S. Hirche  
Feedback linearization based on Gaussian processes with event-triggered online learning  
*IEEE Trans. Automat. Control*, vol. 65, no. 10, pp. 4154–4169, 2019.
- [40] H. Gui and A. H. de Ruiter  
Robustness analysis and performance tuning for the quaternion proportional–derivative attitude controller  
*J. Guid., Control, Dyn.*, vol. 41, no. 10, pp. 2308–2317, 2018.
- [41] P. Huang, M. Wang, Z. Meng, F. Zhang, Z. Liu, and H. Chang  
Reconfigurable spacecraft attitude takeover control in post-capture of target by space manipulators  
*J. Franklin Inst.*, vol. 353, no. 9, pp. 1985–2008, 2016.