

# A Learning-Based Scheme for Safe Deployment of Tethered Space Robot

Ao Jin, Fan Zhang, *Member, IEEE*, Ganghui Shen, *Member, IEEE*, Yifeng Ma, Panfeng Huang, *Senior Member, IEEE*

**Abstract**—This work focuses on the problem of collision avoidance with space debris for large-scale deployment of tethered space robot. To this end, a general scheme that contains offline training and online execution is presented for safe deployment of tethered space robot. Specifically, inspired by contraction theory, a feedback controller is learned from data to guarantee the superior tracking performance in the offline phase. Furthermore, the “tube” where state of tethered space robot would stay within is optimized simultaneously. In the online execution phase, when the space debris are detected, the motion planner generates nominal trajectory by considering safety constraints. Then, in the presence of disturbances, the feedback controller learned offline tracks this nominal trajectory safely without collisions. The proposed scheme allows for the comprehensive utilization of prior knowledge for designing the tracking controller in the offline phase, thereby enhancing the online tracking performance. Finally, the numerical simulations demonstrate effectiveness of the proposed framework.

**Index Terms**—Tethered Space Robot, Tethered System, Learning-Based Control, Collision Avoidance.

## I. INTRODUCTION

The tethered space robot has been widely used for artificial gravity in space, asteroid exploration and other space missions due to its advantages of strong flexibility, low cost and large-scale transportability [1]–[3]. However, the space environment around the Earth is becoming increasingly populated by the space debris. And the population of space debris has reached a critical density in Earth orbits, which significantly threatens the safe operations of spacecraft [4], [5]. Colliding with these space debris would cause extensive damage on the spacecraft. More seriously, collisions with space debris would generate more debris, leading to a chain reaction. Furthermore, when deploying the tethered space robot on a large scale, there is a high possibility that tethered space robot will collide with space debris and colliding with space debris would present a potentially fatal issue for the deployment mission. Thus, the safety of deployment for tethered space robot is an imperative

This research was funded by the National Natural Science Foundation of China under Grant Nos. 62222313, 62327809 and 62173275, and the Fundamental Research Funds for the Central Universities under Grant Nos. D5000230161. (*Corresponding author: Fan Zhang*).

The authors are with Shaanxi Province Innovation Team of Intelligent Robotic Technology, China.

Ao Jin is with Research Center for Intelligent Robotics, School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (E-mail: jinao@mail.nwpu.edu.cn).

Fan Zhang, Ganghui Shen, Yifeng Ma and Panfeng Huang are with Research Center for Intelligent Robotics, School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China (E-mail: fzhang@nwpu.edu.cn, shenganghui@nwpu.edu.cn, yifengma@mail.nwpu.edu.cn, pfhuang@nwpu.edu.cn).

problem expected to be studied for collision avoidance and space debris mitigation.

The task of collision avoidance for spacecraft often contains two parts. One part is the motion planner which has been reported in literature [6]–[8]. Considering the safety constraints, the motion planner can generate a nominal reference trajectory online or offline. Many existed schemes for motion planning could be modified slightly to be suitable for the collision avoidance mission of tethered space robot. The other part is the tracking controller, which tracks the any feasible reference trajectory convergently in the presence of disturbances. Although many studies focus on the stable deployment problem [9]–[16] for tethered space robot, there are a few studies taking the tracking tasks into consideration. Therefore, robust tracking controller design is salient for the safe deployment of tethered space robot.

The contraction theory is a tool for analyzing the nonlinear dynamical system by studying the exponential stability of any two neighbor trajectories [17]. One can design a tracking controller without a specific reference trajectory under the contraction theory framework. The control contraction metric (CCM) [18] synthesis provides a general scheme for designing the tracking controller of nonlinear system by solving a convex optimization problem. The sum-of-squares (SOS) programming has been widely used for solving the convex optimization problem related CCM [19]–[21]. As the neural networks have remarkable ability for approximating the nonlinear functions from data, many learning-based schemes are raised for finding a valid CCM and associated tracking controller [22]–[26] recently. These works parameterize the metric or controller by neural networks and train them with contraction constraints without solving convex optimization problem. In this sense, the learning-based approaches provide an attractive pattern for finding a valid metric and tracking controller.

**Main challenges and motivations:** The collision avoidance with space debris has remained a focal point of space research, especially for large-scale deployment of tethered space robot. As shown in the subsequent section, the collision avoidance problem of tethered space robot is more complicated than other tasks due to the existence of space tether, which engenders more difficulties for the safe deployment. However, a few studies focus collision avoidance problem for tethered space robot. The tracking controller is a crucial component in a collision avoidance task. Compared to other schemes, the CCM synthesis discussed above delivers a more general pipeline for designing tracking controller. However, the CCM tracking controller is usually designed by minimizing the geodesic

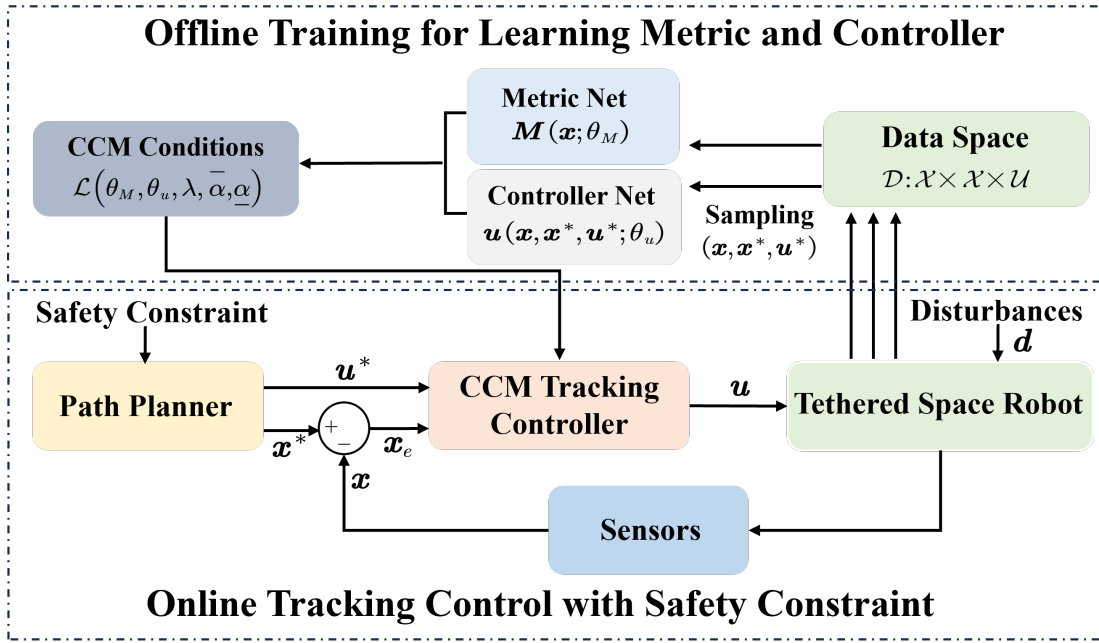


Fig. 1. The illustration of proposed scheme which contains two parts. The first part is learning metric  $M(x, \theta_M)$  and tracking controller  $u(x, x^*, u^*; \theta_u)$  offline with known dynamics. And the second part is tracking the reference trajectory generated from path planner considering the safety constraints with learned metric and tracking controller online.

online, which is not straightforward in many cases. And this optimization-based implementation may not be able to run on the on-board computer of spacecraft in real-time. Thus, some of learning-based schemes [22], [25], [26] parameterize the metric and controller by neural networks together and jointly learn them from data offline. The learned tracking controller is used to track reference trajectory online without further computation of solving optimization problem. However, in these works, the upper bound of distance between trajectories of closed-loop perturbed system and unperturbed system is decided by tuning parameters manually offline, which may be suboptimal and time-consuming. The performance of tracking controller obtained in this way may also not be satisfactory.

Accordingly, a general scheme containing offline training part and online execution part is proposed for deploying tethered space robot to the destination while keep avoiding colliding with space debris. Inspired by contraction theory, a learning-based approach is developed to jointly learn the feedback controller and metric in the offline phase. Moreover, the size of “tube” where the state of tethered space robot would stay within is also optimized offline simultaneously by adding a term to the loss function. In the online phase, considering safety constraints, an optimization-based formulation is given to generate a reference trajectory. And the learned feedback controller will track this reference trajectory robustly online. Apart from tracking the reference trajectory with learned feedback controller directly, a receding-horizon scheme where the reference trajectory is re-generated by considering the local information to reduce tracking cost is also discussed in this work.

#### Contributions:

- 1) A general scheme containing offline training part and

online execution part shown in Fig. 1 is proposed for safe deployment of tethered space robot in the presence of disturbances.

- 2) Inspired by CCM [18], [22], a learning-based approach is developed for finding a valid metric and tracking controller offline simultaneously. The learned tracking controller is used to track nominal trajectory online without additional computation of solving convex problem. Further, a regulation term is considered in loss function to optimize the size of “tube”.
- 3) An optimization-based formulation and collision avoidance constraints for safe deployment of tethered space robot are derived to generate nominal motion plans that can be executed safely online.
- 4) An online path planner scheme in the receding-horizon manner is presented to reduce tracking cost by re-planning the reference trajectory with the local information.

The remainder of this paper is arranged as follows. Section II derives the dynamical equation of tethered space robot and gives a brief introduction of contraction theory. Section III presents the proposed scheme which contains the pattern of learning robust tracking controller and motion planning algorithm. And the theoretical guarantee of learned tracking controller is also given in this section. Section IV illustrates the simulation results of proposed scheme. Section V concludes this paper and discusses some possibilities for further work.

**Notation:** Denote the set of symmetric matrices in  $\mathbb{R}^{n \times n}$  by  $S_n$ , and  $S_n^{>0}$ ,  $S_n^{\geq 0}$  represent the set of symmetric positive definite and semi-positive definite matrices. Given a matrix  $A$ , let  $\tilde{A} = A + A^T$ . The  $\mathcal{T}_x \mathcal{X}$  represents the tangent space of  $\mathcal{X}$  at  $x \in \mathcal{X}$ . Denote the maximum and minimum singular values

of matrix  $\mathbf{A}$  by  $\bar{\sigma}(\mathbf{A})$  and  $\sigma(\mathbf{A})$ , respectively. The  $\|\mathbf{x}\|_A = \sqrt{\mathbf{x}^T \mathbf{A} \mathbf{x}}$  represents the weighted form for matrix  $\mathbf{A} \in S_n^{>0}$ .  $\|\cdot\|_F$  is the Frobenius norm.

## II. PRELIMINARIES AND PROBLEM STATEMENT

### A. Dynamics of Deploying Tethered Space Robot

The platform, tethered space robot (TSR) and space tether constitute the tethered system, which is depicted in Fig. 2. The  $EXYZ$  is the inertial frame, where  $E$  is the center of the Earth. The barycenter of tethered system is denoted as  $O$ . The  $Oxyz$  is the body frame of tethered system, and the direction of axis  $Ox$  is tangential to the motion of tethered system and axis  $Oz$  points toward the center of the Earth  $E$ . The  $Ox_2y_2z_2$  is the body frame of space tether. The angle generated by axis  $Oy$  and axis  $Ox_2$  are defined as in-plane angle  $\vartheta$  and out-plane angle  $\beta$ , respectively. Before deriving the dynamics of deploying tethered space robot, some assumptions are required.

**Assumption 1.** The platform and tethered space robot are mass points. The mass of space tether is neglectable. The orbit of tethered system is circular. The space tether is straight and inelastic during deployment.

Assuming that  $\mathbf{F}_p$  and  $\mathbf{F}_r$  are the force applied to the platform and tethered space robot,  $m_p$  and  $m_r$  are the mass of platform and tethered space robot,  $\mathbf{r}_p$  and  $\mathbf{r}_r$  are the position vector of platform and tethered space robot described in the inertial frame,  $T$  is the tension of space tether ( $T > 0$ ),  $\mathbf{F}$  represents the thrust of tethered space robot. Based on the Newton's second law and above assumptions, the dynamics of deployment of tethered space robot can be described by

$$\frac{\mathbf{F}_p}{m_p} - \frac{\mathbf{F}_r}{m_r} = \mathbf{a}_p - \mathbf{a}_r \quad (1)$$

where the  $\mathbf{a}_p$  and  $\mathbf{a}_r$  are acceleration of the platform and tethered space robot represented in the inertial frame. The  $\mathbf{a}_p$  and  $\mathbf{a}_r$  can be represented by

$$\begin{aligned} \mathbf{a}_p &= -\frac{\mu}{|\mathbf{r}_p|^2} \frac{\mathbf{r}_p}{|\mathbf{r}_p|} + \frac{T}{m_p} \frac{\mathbf{r}_{op}}{|\mathbf{r}_{op}|} \\ \mathbf{a}_r &= -\frac{\mu}{|\mathbf{r}_r|^2} \frac{\mathbf{r}_r}{|\mathbf{r}_r|} + \frac{T}{m_r} \frac{\mathbf{r}_{or}}{|\mathbf{r}_{or}|} + \frac{\mathbf{F}}{m_r} \end{aligned} \quad (2)$$

where  $\mathbf{r}_{op}$  represents the position vector of platform relative to the barycenter of tethered system  $O$  and  $\mathbf{r}_{or}$  has similar definition. The transformation matrix  $\mathbf{C}$  from frame  $Oxyz$  to frame  $Ox_2y_2z_2$  is described by

$$\mathbf{C} = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ \sin \alpha \sin \beta & \cos \beta & \cos \alpha \sin \beta \\ \sin \alpha \cos \beta & -\sin \beta & \cos \alpha \cos \beta \end{bmatrix} \quad (3)$$

The total length of space tether, the distance between barycenter of tethered system and the platform, the distance between barycenter of tethered system and tethered space robot, are denoted by  $l$ ,  $l_p$  and  $l_r$ , respectively. The vector

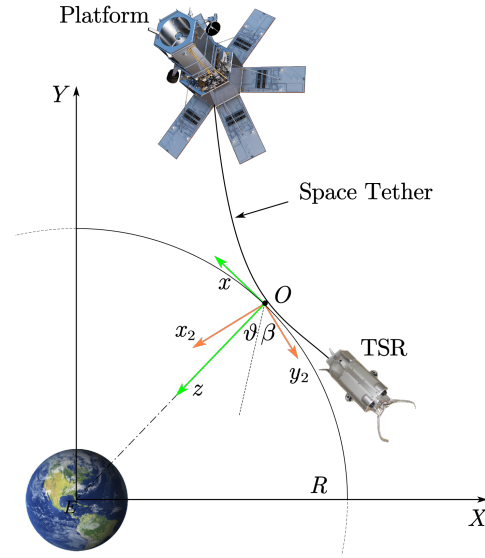


Fig. 2. The definition of tethered system.

$\mathbf{r}_p$ ,  $\mathbf{r}_r$ ,  $\mathbf{r}_{op}$  and  $\mathbf{r}_{or}$  can be represented in the  $Ox_2y_2z_2$  frame by

$$\begin{aligned} \mathbf{r}_p &= \mathbf{r}_o + \mathbf{r}_{po} = \mathbf{C} \begin{bmatrix} 0 & 0 & -R \end{bmatrix}^T + \begin{bmatrix} 0 & 0 & -l_p \end{bmatrix}^T \\ \mathbf{r}_r &= \mathbf{r}_o + \mathbf{r}_{ro} = \mathbf{C} \begin{bmatrix} 0 & 0 & -R \end{bmatrix}^T + \begin{bmatrix} 0 & 0 & -l_r \end{bmatrix}^T \end{aligned} \quad (4)$$

where  $R$  is the radius of orbit. The term  $\mathbf{r}_{op}/|\mathbf{r}_{op}|$ ,  $\mathbf{r}_{or}/|\mathbf{r}_{or}|$  and  $\mathbf{F}$  in (2) can be represented in the frame  $Ox_2y_2z_2$  by

$$\frac{\mathbf{r}_{op}}{|\mathbf{r}_{op}|} = [0, 0, 1]^T, \frac{\mathbf{r}_{or}}{|\mathbf{r}_{or}|} = [0, 0, -1]^T, \mathbf{F} = [F_1, F_2, F_3]^T \quad (5)$$

As  $l_p \ll R$ ,  $l_r \ll R$ , the term  $|\mathbf{r}_p|^{-3}$  and  $|\mathbf{r}_r|^{-3}$  of (2) could be approximated as

$$\begin{aligned} |\mathbf{r}_p|^{-3} &\approx \frac{1}{R^3} (1 - 3 \frac{l_p}{R} \cos \vartheta \cos \beta) \\ |\mathbf{r}_r|^{-3} &\approx \frac{1}{R^3} (1 + 3 \frac{l_r}{R} \cos \vartheta \cos \beta) \end{aligned} \quad (6)$$

Combining the (2)-(6), one can obtain

$$\begin{aligned} \frac{\mathbf{F}_p}{m_p} - \frac{\mathbf{F}_r}{m_r} &= \mathbf{a}_p - \mathbf{a}_r \\ &= \begin{bmatrix} 3\Omega^2 l \sin \vartheta \cos \vartheta \cos \beta - \frac{F_1}{\bar{m}} \\ -3\Omega^2 l \sin \beta \cos \beta \cos^2 \vartheta - \frac{F_2}{\bar{m}} \\ -3\Omega^2 l \cos^2 \vartheta \cos^2 \beta + \Omega^2 l + \frac{T}{\bar{m}} - \frac{F_3}{\bar{m}} \end{bmatrix} \end{aligned} \quad (7)$$

where  $\bar{m} = m_p m_r / (m_p + m_r)$  and  $\Omega$  is the orbit angular velocity. According to the relationship of rotational complex motion of rigid body, the acceleration of platform relative to tethered space robot  $\mathbf{a}_{pr}$  is described by

$$\mathbf{a}_{pr} = \begin{bmatrix} -\vartheta'' \cos \beta l + 2(\vartheta' - \Omega)(\sin \beta \beta' l - l' \cos \beta) \\ l \beta'' + 2\beta' l' + (\vartheta' - \Omega)^2 \sin \beta \cos \beta l \\ -l'' + (\vartheta' - \Omega)^2 \cos^2 \beta l + l \beta'^2 \end{bmatrix} \quad (8)$$

where  $\mathbf{a}_{pr} = \mathbf{a}_p - \mathbf{a}_r$  and  $()'$  represents the derivation with respect to time.

As the tension based control would be applied to the tethered space robot next, the component of thrust along the

tether direction is assumed to be equal to zero for redundancy, i.e.,  $F_3 = 0$ . Combining the (7) and (8), the dimensionless dynamics equation of tethered space robot is given directly as

$$\begin{aligned}\ddot{\vartheta} &= 2(\dot{\vartheta} + 1)[\dot{\beta} \tan \beta - \frac{\dot{\xi}}{\xi}] - 3 \sin \vartheta \cos \vartheta + u_1 \\ \ddot{\xi} &= \xi[\dot{\beta}^2 + (\dot{\vartheta} + 1)^2 \cos^2 \beta + 3 \cos^2 \vartheta \cos^2 \beta - 1] + u_3 \\ \ddot{\beta} &= -2\frac{\dot{\xi}}{\xi}\dot{\beta} - [(\dot{\vartheta} + 1)^2 + 3 \cos^2 \vartheta] \sin \beta \cos \beta + u_2\end{aligned}\quad (9)$$

and

$$\begin{aligned}\xi &= l/l_{\text{total}}, \tau = \Omega t \\ u_1 &= \frac{1}{\Omega^2 L^2 \cos \beta} \frac{F_1}{m_r}, u_2 = -\frac{1}{\Omega^2 L} \frac{T}{\bar{m}}, u_3 = -\frac{1}{\Omega^2 L^2} \frac{F_2}{m_r}\end{aligned}$$

where  $l$  and  $l_{\text{total}}$  are instantaneous and total length of space tether,  $\xi \in (0, 1]$  is the tether deployment ratio,  $\tau$  is the dimensionless form of time  $t$  and the dot ( $\dot{\cdot}$ ) in (9) represents the derivation with respect to the dimensionless time  $\tau$ .

For the in-plane deployment ( $\beta = \dot{\beta} = 0$ ) of tethered space robot, the dynamics can be written as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u} + \mathbf{d} \quad (10)$$

where  $z_1 = \vartheta$ ,  $z_2 = \xi - 1$ ,  $z_3 = \dot{\vartheta}$ ,  $z_4 = \dot{\xi}$ ,  $\mathbf{x} = [z_1, z_2, z_3, z_4]^T$ ,  $\mathbf{u} = [u_1, u_2]^T$  and  $\mathbf{d}$  represents the bounded disturbances, with

$$\begin{aligned}\mathbf{f}(\mathbf{x}) &= \begin{bmatrix} z_3 \\ z_4 \\ -2\frac{z_4}{z_2+1}(z_3+1) - 3 \sin z_1 \cos z_1 \\ (z_2+1)[(z_3+1)^2 + 3 \cos^2 z_1 - 1] \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T\end{aligned}\quad (11)$$

where  $\mathbf{x} \in \mathcal{X} := \{\mathbf{x} \in \mathbb{R}^n \mid -1 < z_2 \leq 0\}$ ,  $\mathbf{u} \in \mathcal{U} := \{\mathbf{u} \in \mathbb{R}^m \mid u_2 \leq 0\}$ ,  $n = 4$ ,  $m = 2$ .

The execution of collision avoidance usually rely on the thrusters which are attached to tethered space robot. Consequently, the derivation of above dynamics considers the thrust produced by thrusters. Note that collision avoidance is active maneuver, which differs from the passive deployment of tethered space robot.

## B. Control Contraction Metrics

The working principle of contraction theory [17] is to analyze the incremental exponential stability (IES) of a system by investigating the evolution of infinitesimal distance between any two neighboring trajectories. Brief introductions of contraction theory are stated as follows.

For an autonomous system  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ ,  $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ , we consider two neighboring trajectories and denote the infinitesimal (virtual) displacement between them by  $\delta_{\mathbf{x}} \in \mathcal{T}_{\mathbf{x}}\mathcal{X}$ . The dynamics of  $\delta_{\mathbf{x}}$  is given by

$$\dot{\delta}_{\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta_{\mathbf{x}} \quad (12)$$

Hence, the dynamics of the squared distance  $\delta_{\mathbf{x}}^T \delta_{\mathbf{x}}$  is derived by

$$\frac{d}{dt} \delta_{\mathbf{x}}^T \delta_{\mathbf{x}} = 2 \delta_{\mathbf{x}}^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta_{\mathbf{x}} \quad (13)$$

If the symmetrical part of the Jacobian  $\partial \mathbf{f} / \partial \mathbf{x}$  is uniformly negative definite, which can be described by

$$\exists \lambda > 0, \forall \mathbf{x} \in \mathcal{X}, \quad \frac{1}{2} \widehat{\frac{\partial \mathbf{f}}{\partial \mathbf{x}}} \preceq -\lambda \mathbf{I} \quad (14)$$

Then, the squared distance  $\delta_{\mathbf{x}}^T \delta_{\mathbf{x}}$  is convergent to zero exponentially at rate  $2\lambda$ . Thus, any of trajectories would converge to a common trajectory. This kind of system is defined to be contracting and  $\lambda$  is called as the contracting rate.

The above observation can be generalized by introducing a control metric  $\mathbf{M}(\mathbf{x}) : \mathcal{X} \rightarrow S_n^{>0}$ , which is a smooth function and is a mapping from  $\mathcal{X}$  to the set of uniformly positive definite symmetric matrices. If the Riemann squared length [17]

$$\mathbf{V}(\mathbf{x}, \delta_{\mathbf{x}}) = \delta_{\mathbf{x}}^T \mathbf{M}(\mathbf{x}) \delta_{\mathbf{x}} \quad (15)$$

satisfies

$$\frac{d}{dt} \mathbf{V}(\mathbf{x}, \delta_{\mathbf{x}}) \preceq -2\lambda \mathbf{V}(\mathbf{x}, \delta_{\mathbf{x}}), \forall \mathbf{x} \in \mathcal{X}, \forall \delta_{\mathbf{x}} \in \mathcal{T}_{\mathbf{x}}\mathcal{X} \quad (16)$$

for some positive constant  $\lambda$ , then the system is contracting at rate  $\lambda$ . The reverse also holds, which could be stated as follows. Any contracting system admits a contraction metric  $\mathbf{M}(\mathbf{x})$  and a positive constant  $\lambda > 0$  such that  $\frac{d}{dt} \mathbf{V}(\mathbf{x}, \delta_{\mathbf{x}}) \preceq 0, \forall \mathbf{x} \in \mathcal{X}, \forall \delta_{\mathbf{x}} \in \mathcal{T}_{\mathbf{x}}\mathcal{X}$ .

The above analysis could be extended to the systems with control input, which introduces the concept of control contraction metric (CCM) synthesis. The CCM synthesis analyzes the IES properties of closed-loop system with designed controller. Now consider a control-affine nonlinear system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u} + \mathbf{d} \quad (17)$$

where  $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ ,  $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$ , and  $\mathbf{d} \in \mathcal{D} \subseteq \mathbb{R}^n$  are state, control input and bounded disturbances, respectively.  $\mathcal{X}, \mathcal{U}$  and  $\mathcal{D}$  are compact sets. It is assumed that the  $\mathbf{f}$  and  $\mathbf{B}$  are smooth functions in their domains, the control input of nonlinear system is piece-wise continuous function, and (17) holds with right derivative at discontinuous points.

The variational dynamical equation of (17) ignoring the disturbances is given by

$$\dot{\delta}_{\mathbf{x}} = \mathbf{A}(\mathbf{x}, \mathbf{u}) \delta_{\mathbf{x}} + \mathbf{B}(\mathbf{x}) \delta_{\mathbf{u}} \quad (18)$$

where  $\delta_{\mathbf{x}} \in \mathcal{T}_{\mathbf{x}}\mathcal{X}$ ,  $\delta_{\mathbf{u}} \in \mathcal{T}_{\mathbf{u}}\mathcal{U}$ ,  $\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \sum_{i=1}^m \frac{\partial \mathbf{b}_i}{\partial \mathbf{x}} \mathbf{u}_i$ ,  $\mathbf{b}_i$  is the  $i$  th column of  $\mathbf{B}(\mathbf{x})$  and  $\mathbf{u}_i$  is  $i$  th component of  $\mathbf{u}$ . This work adopts the definition of CCM as follows [20].

**Definition 1.** If there exists a uniformly bounded metric  $\mathbf{M}(\mathbf{x})$ , i.e.,  $\underline{\alpha} \mathbf{I} \prec \mathbf{M}(\mathbf{x}) \prec \bar{\alpha} \mathbf{I}$ , and a differential controller  $\delta_{\mathbf{u}} : \mathcal{T}_{\mathbf{x}}\mathcal{X} \rightarrow \mathcal{T}_{\mathbf{u}}\mathcal{U}$ , such that  $\dot{\mathbf{V}}(\mathbf{x}, \delta_{\mathbf{x}}) \prec 0, \forall \mathbf{x} \in \mathcal{X}, \forall \delta_{\mathbf{x}} \in \mathcal{T}_{\mathbf{x}}\mathcal{X}$ . Then the metric  $\mathbf{M}(\mathbf{x})$  is referred as a CCM of the system.

Given a feedback controller  $\mathbf{u}(t) = \mathbf{k}(\mathbf{x}(t)) + \mathbf{v}(t)$ , if the closed-loop system is contracting with rate  $\lambda$  and a metric  $\mathbf{M}(\mathbf{x})$  for arbitrary continuous signal  $\mathbf{v}(t)$ , then

$$\dot{\mathbf{V}}(\mathbf{x}, \delta_{\mathbf{x}}) < -2\lambda \mathbf{V}(\mathbf{x}, \delta_{\mathbf{x}}) \quad (19)$$

holds for  $x \in \mathcal{X}$ ,  $\delta_x \in \mathcal{T}_x \mathcal{X} \setminus \{0\}$ . The above inequality (19) can be derived as

$$\begin{aligned} \dot{V}(x, \delta_x) &= \frac{d}{dt}(\delta_x^T M(x) \delta_x) \\ &= \delta_x^T [\dot{M} + \overline{M(A+BK)}](x) \delta_x < -2\lambda \delta_x^T M(x) \delta_x \end{aligned} \quad (20)$$

where  $\dot{M} = \sum_{i=1}^n \frac{\partial M}{\partial x_i} \dot{x}$ ,  $K = \frac{\partial k}{\partial x}$ . The above inequality could be equivalent to

$$\dot{M} + \overline{M(A+BK)} + 2\lambda M \prec 0 \quad (21)$$

where  $x \in \mathcal{X}$ ,  $\delta_x \in \mathcal{T}_x \mathcal{X} \setminus \{0\}$ . According to the Definition 1, if one could find the controller  $u$  and metric  $M(x)$  satisfying condition (21), then the system is contracting and  $M(x)$  is a valid CCM. Expanding  $\dot{V}(x, \delta_x)$  further, it gives

$$\begin{aligned} \dot{V}(x, \delta_x) &= \frac{d}{dt}(\delta_x^T M(x) \delta_x) \\ &= \delta_x^T [\partial_f M + \overline{M(\frac{\partial f}{\partial x} + BK)} \\ &\quad + \sum_{i=1}^m (\partial_{b_i} M + \overline{M \frac{\partial b_i}{\partial x}})(k_i + v_i)] \delta_x \end{aligned} \quad (22)$$

The above inequality holds for arbitrary continuous signal  $v(t)$  for a contracting system. Thus, it is required that

$$\partial_{b_i} M + \overline{M \frac{\partial b_i}{\partial x}} = 0, \forall x \in \mathcal{X}, i = 1, \dots, m \quad (23)$$

holds. Then (19) can be written as

$$\delta_x^T [\partial_f M + \overline{M(\frac{\partial f}{\partial x} + BK)} + 2\lambda M] \delta_x < 0 \quad (24)$$

where  $x \in \mathcal{X}$ ,  $\delta_x \in \mathcal{T}_x \mathcal{X} \setminus \{0\}$ . Furthermore, if the system is naturally contracting in directions orthogonal to the span of the control inputs, i.e.,  $\delta_x^T M B = 0$ , the inequality (21) reduces to

$$\delta_x^T [\partial_f M + \overline{M \frac{\partial f}{\partial x}} + 2\lambda M] \delta_x < 0 \quad (25)$$

where  $x \in \mathcal{X}$ ,  $\delta_x \in \mathcal{T}_x \mathcal{X} \setminus \{0\}$ . The above condition could be written as

$$\partial_f M + \overline{M \frac{\partial f}{\partial x}} + 2\lambda M \prec 0 \quad (26)$$

Both condition (21) and conditions (23), (26) are sufficient for searching a valid CCM. But the latter is stronger than former. This means that the conditions (23), (26) can provide more precise optimization guidance for searching a valid CCM than condition (21).

**Lemma 1.** If a uniformly bounded metric  $M(x)$  satisfies the condition (21) or conditions (23) and (26), then the metric  $M(x)$  is a CCM for the system  $\dot{x} = f(x) + B(x)u$  and the contracting rate is  $\lambda$ .

Consider a feedback controller  $u = k(x, x^*) + u^*$  satisfies that  $u = u^*$  when  $x = x^*$  and a reference trajectory  $(x^*, u^*)$  which is a valid trajectory of the closed-loop system. A contracting system has the property that any of trajectories would converge to a common trajectory, which can be stated as follows.

**Proposition 1.** If the condition (21) or conditions (23) and (26) are satisfied with a uniformly bounded metric  $M(x)$ ,

i.e.,  $\alpha I \preceq M(x) \preceq \bar{\alpha} I$  and a feedback tracking controller  $u = k(x, x^*) + u^*$ , then the displacement between actual trajectory and nominal trajectory of unperturbed close-loop system exponentially converges to zero, i.e.

$$\|x(t) - x^*(t)\| \leq \sqrt{\frac{\bar{\alpha}}{\alpha}} e^{-\lambda t} \|x(0) - x^*(0)\| \quad (27)$$

where  $\lambda$  is the contracting rate and the constant  $\sqrt{\bar{\alpha}/\alpha}$  is the overshoot constant.

When the closed-loop system is perturbed with disturbances, there is a similar conclusion for the contracting system.

**Definition 2.** Define  $\Omega : \mathbb{R}^n \rightarrow 2^{\mathbb{R}^n}$  as a mapping s.t.  $x \in \Omega(x)$  and  $\Omega(x)$  is a closed and bounded set for every  $x$ . For a nonlinear system  $\dot{x} = f(x, u)$ ,  $\Omega(\cdot)$  is a robust control invariant (RCI) mapping if there exists a tracking controller  $k(x^*, x)$  s.t.  $x(t_0) \in \Omega(x^*(t_0))$ , then for any realizations of disturbance  $d(t)$ ,  $x(t) \in \Omega(x^*(t))$  for all  $t_0 \leq t \leq t_f$ .

**Lemma 2.** [20] If there is a uniformly bounded metric  $M(x)$ , i.e.,  $\underline{m}I \preceq M(x) \preceq \bar{m}I$  and a feedback controller  $u = k(x, x^*) + u^*$  satisfying the conditions (23) and (26), or condition (21). Then, for the same reference trajectory, the distance between actual trajectory of perturbed closed-loop dynamics and nominal trajectory of unperturbed dynamics is bounded. And

$$\Omega(x^*) := \left\{ x \in \mathcal{X} : \|x - x^*\|_{\underline{M}}^2 \leq \bar{c}^2 \right\} \quad (28)$$

is an ellipsoid robust control invariant (RCI) set of closed-loop system stabilized with  $u = k(x, x^*) + u^*$ , where  $\bar{c} = \sup_{x \in \mathcal{X}} \bar{\sigma}(\Theta(x)) \bar{d}/\lambda$ ,  $M(x) = \Theta(x)^T \Theta(x)$  and  $M(x) \geq \underline{M}$  holds for all  $x \in \mathcal{X}$ .

Note that the RCI set (28) is not the minimal RCI set of closed-loop system stabilized with CCM controller. The ellipsoid RCI (28) is an outer approximation of minimal RCI mapping which has been derived in [20]. Furthermore, according to Lemma 2, the RCI set  $\Omega(\cdot)$  is derived on condition that CCM conditions hold. Therefore, if the MPC is utilized for producing the nominal trajectory, the set  $\mathcal{X} \ominus \Omega$  should be the tightened state constraint in the MPC problem to guarantee that the CCM conditions hold.

### C. Problem Formulation

Differing from other deployment missions that have studied in the literature, this work focuses on the safety of deploying tethered space robot. The safe deployment mission for tethered space robot is formulated as follows.

**The problem of safely deploying:** Considering in-plane dynamics (10) of TSR under Assumption 1, the objective is to design a control law that can deploy the tethered space robot from a initial position  $x_0 = [0, 0, \xi_0 - 1, \dot{\xi}_0]^T \in \mathcal{X}$  to the final position  $x_f = [0, 0, 0, 0]^T \in \mathcal{X}$  while keeping avoid colliding with space debris along the path.

## III. MAIN RESULTS

To solve the safety problem for TSR deployment mission, we propose a general scheme that contains a collision-free path

planner and a robust tracking controller. The proposed scheme is depicted in Fig. 1. The path planner adopts the optimization-based approach considering the safety constraints to produce a reference trajectory. Then a learning-based CCM controller is designed for tracking the planned trajectory in the presence of disturbances.

#### A. Learning Robust Tracking Controller Using Control Contraction Metrics

As shown in Lemma 1, if one could find a uniformly bounded metric  $M(x)$  confirming the condition (21) for a nonlinear system, a feedback controller  $u$  can be built for tracking any given reference trajectory exponentially. In this work, a learning-based scheme is proposed to jointly learn the control contraction metric and the feedback controller with known dynamics. The metric and controller are represented by two deep neural networks

$$M(x; \theta_M) = W_M^{L+1} a(W_M^L (\dots a(W_M^1 \cdot x))) \quad (29)$$

$$k(x, x^*; \theta_u) = W_u^{L+1} a(W_u^L (\dots a(W_u^1 \cdot (x, x^*)))) \quad (30)$$

where  $a$  is the nonlinear activation function and  $\theta_M = W_M^1, \dots, W_M^{L+1}$ ,  $\theta_u = W_u^1, \dots, W_u^{L+1}$  are the parameters of associated neural networks. The controller  $u(x, x^*, u^*; \theta_u)$  is given by

$$u(x, x^*, u^*; \theta_u) = u^* + k(x, x^*; \theta_u) \quad (31)$$

The architecture of corporately learning control contraction metric and feedback controller is depicted in the Fig. 3. The objective is to find the metric  $M(x)$  and controller  $u$  satisfying the condition (21). Thus, the control contraction metric risk is given as

$$L_{\text{ccm}}(\theta_M, \theta_u) = \mathbb{E}_{(x, x^*, u^*) \sim \rho(\mathcal{D})} \mathcal{G}(-\phi(x, x^*, u^*; \theta_M, \theta_u) - \sigma) \quad (32)$$

where  $\phi(x, x^*, u^*; \theta_M, \theta_u)$  represents the LHS (Left-Hand Side) of (21),  $\mathbb{E}$  means the value of expectation,  $\rho(\mathcal{D})$  denotes the uniform distribution over the sampling data space  $\mathcal{D} := \mathcal{X} \times \mathcal{X} \times \mathcal{U}$ ,  $\sigma$  is a small positive constant and the auxiliary function  $\mathcal{G}$  is penalized for non-positive definiteness, i.e.,  $\mathcal{G}(A) = 0$  if and only if  $A \succeq 0$ . Therefore,  $L_{\text{ccm}}(\theta_M, \theta_u) = 0$  indicates that the condition (21) holds over the domain with the control contraction metric  $M(x; \theta_M)$  and the controller  $u(x, x^*, u^*; \theta_u)$ .

However, it may lead to poor performance learning the metric and feedback controller by minimizing (32) alone [22]. As shown in Lemma 1, the conditions (23) and (26) are also sufficient for searching a CCM candidate. Therefore, imposing constraints (23) and (26) can help for optimization during training neural network. And by introducing a dual metric  $W(x) = M(x)^{-1}$  and the change of variable  $\eta = M(x)\delta_x$  [18], the two conditions could be rewritten as

$$B_\perp^T [-\partial_f W + \widehat{\frac{\partial f}{\partial x} W} + 2\lambda W] B_\perp \prec 0 \quad (33)$$

$$B_\perp^T [\partial_{b_i} W - \widehat{\frac{\partial b_i}{\partial x} W}] B_\perp = 0, i = 1, \dots, m \quad (34)$$

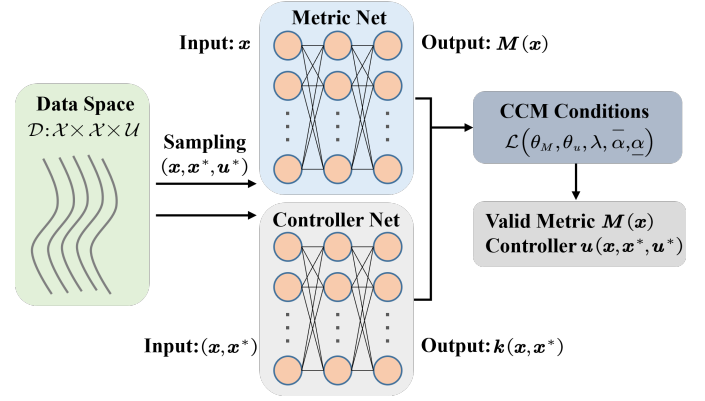


Fig. 3. The schematic of learning control contraction metric and tracking controller simultaneously.

where  $B_\perp$  satisfies  $B_\perp^T B = 0$ . By imposing constraints (33) and (34), the risk functions are raised by

$$L_{C_1}(\theta_M) = \mathbb{E}_{(x, x^*, u^*) \sim \rho(\mathcal{D})} \mathcal{G}(-C_1(x; \theta_M)) \quad (35)$$

$$L_{C_2}(\theta_M) = \mathbb{E}_{(x, x^*, u^*) \sim \rho(\mathcal{D})} \|C_2\|_F \quad (36)$$

where  $C_1(x; \theta_M)$  represents the LHS of (33),  $C_2 := (C_2^1, \dots, C_2^i, \dots, C_2^m)$ ,  $C_2^i$  denotes the LHS of (34).

Since the metric  $M(x)$  is required to be uniformly bounded, it should be imposed constraint to bound the condition number of metric. Construct the metric  $M(x)$  as

$$M(x; \theta_M) = \underline{\alpha} I + m(x; \theta_M)^T m(x; \theta_M) \quad (37)$$

where  $m(x; \theta_M) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$  is a neural network which is parameterized with  $\theta_M$ . This formulation guarantees that  $M(x; \theta_M) \succeq \underline{\alpha} I$ ,  $\forall x \in \mathcal{X}$ . Thus, the smallest condition number of metric is lower bounded by  $\underline{\alpha}$ . The largest condition number risk function could be developed as

$$L_M(\theta_M) = \mathbb{E}_{(x, x^*, u^*) \sim \rho(\mathcal{D})} \mathcal{G}(\bar{\alpha} I - M(x; \theta_M)) \quad (38)$$

The upper bound of normalized distance within the RCI set defined in Lemma 2 is bounded by

$$\sup_{x \in \Omega(x^*)} \frac{\|x - x^*\|^2}{d^2} = \frac{\bar{\sigma}(\Theta(x))^2}{\lambda^2 \underline{\alpha}^2} \leq \frac{1}{\lambda^2} \cdot \frac{\bar{\alpha}}{\underline{\alpha}} \quad (39)$$

It is seen that the size of RCI set is related to the contracting rate  $\lambda$  and the overshoot constant  $\sqrt{\bar{\alpha}/\underline{\alpha}}$ . In addition, as shown in the numerical simulation section later, there is a trade-off between the contracting rate  $\lambda$  and overshoot constant  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  on the control performance. Therefore, the selecting of two parameters contracting rate  $\lambda$  and overshoot constant  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  is crucial to the control performance. But the relationship between two parameters and control performance is complicated and it will take a lot of trial and error to choose an acceptable but possibly not optimal set of parameters. Thus, instead of tuning  $\lambda$  and  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  by hand, we developed a term

$$L_{\text{RCI}}(\theta_M, \lambda, \bar{\alpha}, \underline{\alpha}) = \frac{1}{\lambda^2} \cdot \frac{\bar{\alpha}}{\underline{\alpha}} \quad (40)$$

and considered this term as a part of loss function trying to find two parameters  $\lambda$  and  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  optimally and automatically.

Therefore, for purpose of training the neural network with the sampling data over the data space  $\mathcal{D}$ , the following empirical risk function is given by

$$\begin{aligned} \mathcal{L}(\theta_M, \theta_u, \lambda, \bar{\alpha}, \underline{\alpha}) = & \\ \frac{1}{N} \sum_{i=1}^N [L_{\text{ccm}}(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{u}_i^*; \theta_M, \theta_u) + L_{C_1}(\mathbf{x}_i; \theta_M) & \quad (41) \\ + L_{C_2}(\mathbf{x}_i; \theta_M) + L_M(\mathbf{x}_i; \theta_M)] + \zeta L_{\text{RCI}}(\theta_M, \lambda, \bar{\alpha}, \underline{\alpha}) \end{aligned}$$

where  $(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{u}_i^*)_{i=1}^N$  contains samples distributed uniformly over the data space  $\mathcal{D}$  and  $\zeta$  is a user-defined parameter.

There are several implementations of auxiliary function  $\mathcal{G}$ . In this work, it is implemented as follows: Given a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and sample  $K$  points  $\{\mathbf{p}_i \in \mathbb{R}^n | \|\mathbf{p}_i\| = 1\}_{i=1}^K$ , the function  $\mathcal{G}$  for penalizing non-positive definiteness is defined as  $\mathcal{G}(\mathbf{A}) = \frac{1}{K} \sum_{i=1}^K \max\{0, \mathbf{p}_i^T \mathbf{A} \mathbf{p}_i\}$ .

Note that in the most cases, the learning-based CCM controller which is trained without  $L_{\text{RCI}}$  in (41) has a good tracking performance. The parameters  $\lambda, \bar{\alpha}, \underline{\alpha}$  need to be tuned manually in this setting. Thus, this motivates us to impose  $L_{\text{RCI}}$  in (41), which is conducive to tune  $\lambda, \bar{\alpha}, \underline{\alpha}$  automatically and minimize the size of RCI set outer approximation. While imposing  $L_{\text{RCI}}$  in (41) may lead bad guidance for searching a valid CCM and feedback controller. To circumvent this issue, this work adopts updating the weights of  $\lambda, \bar{\alpha}, \underline{\alpha}$  much slower than updating those of  $\mathbf{M}(\mathbf{x}; \theta_M)$  and  $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*; \theta_u)$ . More details of implementation will be discussed later.

**Remark 1.** As shown in [27], if  $\mathbf{B}(\mathbf{x})$  has the sparse representation as

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} \mathbf{O}_{(n-m) \times m} \\ \mathbf{b}(\mathbf{x}) \end{bmatrix} \quad (42)$$

where  $\mathbf{b}(\mathbf{x}) \in \mathbb{R}^{m \times m}$  is an invertible matrix, the condition (34) would be satisfied automatically if the upper-left  $(n-m) \times (n-m)$  block of the dual metric  $\mathbf{W}(\mathbf{x})$  is not a function of the last  $m$  components of  $\mathbf{x}$ . Hence, one could remove the  $L_{C_2}$  from the empirical risk function (41) by utilizing the above property if the  $\mathbf{B}(\mathbf{x})$  of a dynamical system has the sparse representation as (42).

### B. Theoretical Guarantees of Learning-Based Robust Tracking Controller

The condition (21) should be satisfied for all  $(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*) \in \mathcal{D}$  to guarantee that the learned metric  $\mathbf{M}(\mathbf{x}; \theta_M)$  is a valid CCM. Thus, a theoretical analysis of learning-based tracking controller is given in this section.

**Lemma 3.** For a Lipschitz continuous function  $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}$ , with Lipschitz constant  $L_f$ , discretizing the domain  $\mathcal{X}$  into a grid such that the distance between any two grid points is less than  $\mu$ , if  $\mathbf{f}(\mathbf{x}_i) < -L_f \mu$  holds for all grid points, then  $\mathbf{f}(\mathbf{x}) < 0$  is satisfied for all  $\mathbf{x} \in \mathcal{X}$ .

*Proof.* The function  $\mathbf{f}$  is Lipschitz continuous with Lipschitz constant  $L_f$ , then one has

$$\begin{aligned} |\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_i)| &< L_f \mu \\ \Rightarrow -L_f \mu + \mathbf{f}(\mathbf{x}_i) &< \mathbf{f}(\mathbf{x}) < L_f \mu + \mathbf{f}(\mathbf{x}_i) \end{aligned} \quad (43)$$

where  $\mathbf{x} \in \mathcal{X}$  and  $\mathbf{x}_i$  is the closest grid point to  $\mathbf{x}$ . And  $\mathbf{f}(\mathbf{x}_i) < -L_f \mu$  holds for all grid points, the following inequality holds

$$\mathbf{f}(\mathbf{x}) < L_f \mu + \mathbf{f}(\mathbf{x}_i) < 0 \quad (44)$$

which implies that if  $\mathbf{f}(\mathbf{x}_i) < -L_f \mu$  holds for all grid points, then  $\mathbf{f}(\mathbf{x}) < 0$  holds for all  $\mathbf{x} \in \mathcal{X}$ .  $\square$

The following proposition shows that the largest eigenvalue function of the LHS of (21) has a Lipschitz constant if learned metric and tracking controller has Lipschitz constants. Note that  $\mathbf{M}, \mathbf{A}, \mathbf{B}$  and  $\mathbf{K}$  are functions of  $\mathbf{x}, \mathbf{x}^*$  and  $\mathbf{u}^*$ .

**Proposition 2.** [22] Suppose  $L_{\dot{\mathbf{M}}}, L_A, L_B, L_M$  and  $L_K$  are the Lipschitz constant of  $\dot{\mathbf{M}}, \mathbf{A}, \mathbf{B}, \mathbf{M}$  and  $\mathbf{K}$ , respectively, and 2-norms of  $L_A, L_B, L_M$  and  $L_K$  are bounded by  $N_A, N_B, N_M$  and  $N_K$ , respectively. Then, the largest eigenvalue function  $\lambda_{\max}(\dot{\mathbf{M}} + \mathbf{M}(\mathbf{A} + \mathbf{B}\mathbf{K}) + 2\lambda\mathbf{M})$  has a Lipschitz constant  $L_{\dot{\mathbf{M}}} + 2(N_M L_A + N_A L_M + N_M N_B L_K + N_B N_K L_M + N_M N_K L_B + \lambda L_M)$ .

**Theorem 1.** If the condition (21) holds for all discrete samples over the domain and the largest eigenvalue function of the LHS of (21) has a Lipschitz constant, then the learned metric  $\mathbf{M}(\mathbf{x}; \theta_M)$  is a valid CCM that admits a tracking controller  $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*; \theta_u)$ .

*Proof.* Lemma 3 shows that for a Lipschitz continuous function  $\mathbf{f}$ , if  $\mathbf{f}(\mathbf{x}_i) < -L_f \mu$  holds for all discrete samples where the  $L_f$  is Lipschitz constant and  $\mu$  is the distance between discrete samples, then  $\mathbf{f}(\mathbf{x}) < 0$  holds over the domain. Thus, when the condition (21) holds for all discrete samples over the domain and the largest eigenvalue function of the LHS of (21) is Lipschitz continuous, then the learned metric  $\mathbf{M}(\mathbf{x}; \theta_M)$  is a valid CCM. Due to the properties of contraction system, the feedback controller  $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*; \theta_u)$  could track any feasible trajectory. This completes proof.  $\square$

As Theorem 1 shows, the satisfactory of condition (21) is guaranteed by combining the discrete samples (Lemma 3) and Lipschitz constant of LHS of (21). The Lipschitz constant of deep neural network could be estimated by using many methods, such as [28]–[30].

**Remark 2.** According to Lemma 3, one should formulate the empirical risk function  $L_{\text{ccm}}(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{u}_i^*; \theta_M, \theta_u)$  by adding a small positive constant  $\sigma$

$$L_{\text{ccm}} = \mathbb{E}_{(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*) \sim \rho(\mathcal{D})} \mathcal{G}(-\phi(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*; \theta_M, \theta_u) - \sigma) \quad (45)$$

which corresponds to the (32). The above formulation guarantees that the  $\lambda_{\max}(\dot{\mathbf{M}} + \mathbf{M}(\mathbf{A} + \mathbf{B}\mathbf{K}) + 2\lambda\mathbf{M}) < -L_\mu$  would hold over the domain when  $L_{\text{ccm}} = 0$ .

### C. Path Planning for Avoiding Space Debris

The objective of path planner is to generate a collision-free reference trajectory for tracking when the deployment mission is ongoing, which is described in Fig. 4. This work utilizes an optimization-based approach for path planning.

Denote the nominal trajectory generated by path planner for the unperturbed dynamics of (10) with tightened constraints as

$(\mathbf{x}^*, \mathbf{u}^*)$ . Consider the RCI set (28) derived in Sec.II-B, the tightened state and control constraints are given as

$$\begin{aligned} \mathbf{x}^* \in \bar{\mathcal{X}} = \mathcal{X} \ominus \Omega, \mathbf{u}^* \in \bar{\mathcal{U}} = \{\bar{\mathbf{u}} \in \mathcal{U} | \bar{\mathbf{u}} + \mathbf{k}(\mathbf{x}, \mathbf{x}^*) \in \mathcal{U}, \\ \mathbf{x} \in \Omega(\mathbf{x}^*), \forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{x}^* \in \bar{\mathcal{X}}\} \end{aligned} \quad (46)$$

It is assumed that the sensors on tethered space robot could obtain the relative position of space debris during deployment to form a circle where the all detected space debris is located in. The schematic of tethered space robot deployment is shown in Fig. 4. Before deriving the formulation of path planner, the safety constraints of avoiding space debris for tethered space robot deployment are deduced as follows.

The collision-free task of tethered space robot deployment is distinct from the other tasks. As shown in Fig. 4, except for the tethered space robot, the space tether should not collide with the space debris either. This means that both tethered space robot and the space tether are desired to avoid colliding with space debris during deployment. Since it is assumed that space tether is straight and all detected space debris is located in a circle, the collision checking of space tether with space debris could be regarded as a problem of intersectionality between a straight line and a circle. Note that the position of space tether is constantly changing, the space tether is of variable length, and tethered space robot is considered as a mass point. Thus, the safety constraints of avoiding colliding with space debris could be written as

$$\mathcal{W} := \{\mathbf{x} \in \mathbb{R}^4 \mid D_p(\mathbf{x}) > r, d_p(\mathbf{x}) > r\} \quad (47)$$

where  $D_p(\mathbf{x})$  denotes the distance between point  $p$  and the space tether,  $d_p(\mathbf{x})$  represents the distance between point  $p$  and tethered space robot. The point  $p$  is the center of circle where all detected space debris is located in.

The above safety constraints state that distance between the point  $p$  and space tether, distance between the point  $p$  and tethered space robot are both greater than the radius of circle, which indicates that neither the space tether or tethered space robot would collide with space debris. Considering the safety of tethered space robot deployment in the presence of disturbances, the safety margin [31] (the light blue area in Fig. 4) is introduced. Then, safety constraint (47) could be rewritten as

$$\mathcal{W}_h := \{\mathbf{x} \in \mathbb{R}^4 \mid D_p(\mathbf{x}) > r + h, d_p(\mathbf{x}) > r + h\} \quad (48)$$

where  $h$  is the size of safety margin as shown in Fig. 4.

The definitions of  $D_p(\mathbf{x})$  and  $d_p(\mathbf{x})$  are depicted in Fig. 5. The tethered space robot is assumed as a mass point and the safety constraints (47) and (48) do not take the size of tethered space robot into consideration in this work. It is obviously that the  $d_p(\mathbf{x})$  is always greater than or equal to  $D_p(\mathbf{x})$ . Therefore, in this setting, the safety constraints (47) and (48) could be reduced to  $\mathcal{W} := \{\mathbf{x} \in \mathbb{R}^4 \mid D_p(\mathbf{x}) > r\}$ ,  $\mathcal{W}_h := \{\mathbf{x} \in \mathbb{R}^4 \mid D_p(\mathbf{x}) > r + h\}$ , respectively. If one would like to consider the size of tethered space robot in the collision avoidance problem, the safety constraints (47) and (48) can be modified slightly by  $\mathcal{W} := \{\mathbf{x} \in \mathbb{R}^4 \mid D_p(\mathbf{x}) > r + r_{\text{TSR}}, d_p(\mathbf{x}) > r + r_{\text{TSR}}\}$ ,  $\mathcal{W}_h := \{\mathbf{x} \in \mathbb{R}^4 \mid D_p(\mathbf{x}) >$

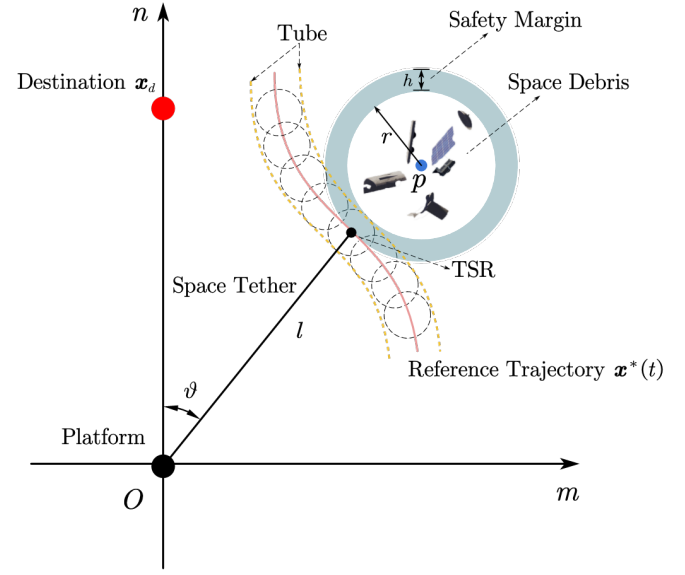


Fig. 4. The illustration of TSR safe deployment. The radius and center of the circle where all detected space debris is located in are denoted by  $r$  and  $p$ , respectively.

$r + r_{\text{TSR}} + h, d_p(\mathbf{x}) > r + r_{\text{TSR}} + h\}$ , where the  $r_{\text{TSR}}$  is the size of tethered space robot.

Next, the formulation of path planner is given as follows. There are many schemes for motion planning [6]. As the optimization-based approach could handle the constraints and optimize the performance index simultaneously, this work adopts it to generate reference trajectory. The formulation of path planner for nominal dynamics considering the safety constraints is given as follows.

$$\begin{aligned} \min \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t)) dt \\ \text{s.t. } \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u} \\ \mathbf{x}(0) = \mathbf{x}_0, \mathbf{x}(t_f) = \mathbf{x}_f \\ \mathbf{x}(t) \in \bar{\mathcal{X}} \cap \mathcal{W}_h, \mathbf{u}(t) \in \bar{\mathcal{U}} \quad \forall t \in [t_0, t_f] \end{aligned} \quad (49)$$

where  $g(\cdot)$  is the cost function,  $\mathbf{x}_0$  and  $\mathbf{x}_f$  denote the initial and terminal state of deployment, respectively. Note that the safety margin is considered here. As discussed above, the formulation (49) would result a safely nominal motion plans  $(\mathbf{x}^*, \mathbf{u}^*)$  for tracking.

#### D. Safe Deployment for Tethered Space Robot

With the motion plan  $(\mathbf{x}^*, \mathbf{u}^*)$  obtained from path planner, one could adopt the learning-based tracking controller  $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*; \theta_u)$  directly until the tethered space robot arrives the destination. If the on-board computer could provide sufficient computational resources, a receding-horizon scheme where the reference trajectory is locally re-generated over a short time horizon can also be considered. One of benefits of this receding-horizon scheme is that it could further reduce

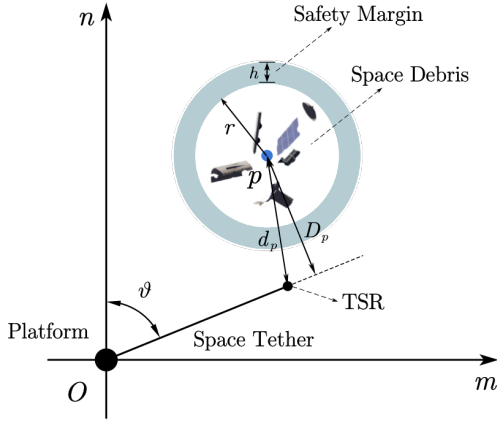


Fig. 5. The definition of  $D_p(x)$  and  $d_p(x)$

the tracking cost as the local information is taken into account. The implementation of it is stated as follows.

Given current state  $\mathbf{x}(t_i)$  of tethered space robot, a reference path  $(\mathbf{x}^*, \mathbf{u}^*)$  and the RCI set (28), solving the following MPC problem at time instant  $t_i$

$$P(\bar{\mathbf{x}}, t_i) : V^0(\bar{\mathbf{x}}, t_i) = \min_{\bar{\mathbf{u}}(\cdot), \bar{\mathbf{x}}(t_i)} \{V(\bar{\mathbf{x}}(t_i), \bar{\mathbf{u}}(t_i))\} \quad (50a)$$

$$V(\bar{\mathbf{x}}(t_i), \bar{\mathbf{u}}(t_i)) = \int_{t_i}^{t_i+T} l(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)) dt \quad (50b)$$

$$\text{s.t. } \dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}) + \mathbf{B}\bar{\mathbf{u}} \quad (50c)$$

$$\mathbf{x}(t_i) \in \Omega(\bar{\mathbf{x}}(t_i)) \quad (50d)$$

$$\bar{\mathbf{x}}(t) \in \bar{\mathcal{X}} \cap \mathcal{W}_h, \bar{\mathbf{u}}(t) \in \bar{\mathcal{U}} \quad \forall t \in [t_i, t_i + T] \quad (50e)$$

$$\bar{\mathbf{x}}(t_i + T) = \mathbf{x}^*(t_i + T) \quad (50f)$$

where  $\bar{\mathbf{x}}(t)$  and  $\bar{\mathbf{u}}(t)$  are the predicted state and control input,  $T$  denotes the predictive horizon and  $l(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t))$  is the stage cost which is generally defined by

$$l(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)) = \frac{1}{2} [\bar{\mathbf{x}}(t)^T \mathbf{Q} \bar{\mathbf{x}}(t) + \bar{\mathbf{u}}(t)^T \mathbf{R} \bar{\mathbf{u}}(t)] \quad (51)$$

where  $\mathbf{Q}$  and  $\mathbf{R}$  are positive definite matrices.

The terminal set (50f) is raised to ensure the recursive feasibility of the MPC problem (50), which is guaranteed by the following lemma.

**Lemma 4.** If the MPC problem is feasible at the initial time-step  $t_0$ , the problem (50) is recursively feasible for all  $t_i$ .

*Proof.* See V-A □

The above MPC problem provides a receding-horizon pattern for re-planning the reference trajectory locally online, which should be considered as a local re-optimization planner. Note that the initial state  $\bar{\mathbf{x}}$  of unperturbed dynamics is also an optimization variable. The pipeline of this receding-horizon path planner with learning-based tracking controller is provided in **Algorithm 1**.

### Algorithm 1 Safe deployment for tethered space robot

#### Offline

- 1: Input: Unperturbed dynamics, state constraint  $\mathcal{X}$  and control constraint  $\mathcal{U}$ .
- 2: Output: Metric  $M(\mathbf{x}; \theta_M)$ , contraction rate  $\lambda$ , tracking controller  $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*; \theta_u)$  and RCI set  $\Omega$ .

#### Online

- 3: Input: Initial state  $\mathbf{x}_0$ , terminal state  $\mathbf{x}_f$ , tightened state constraint  $\bar{\mathcal{X}}$  and control constraint  $\bar{\mathcal{U}}$ .
- 4: Compute the nominal reference trajectory  $(\mathbf{x}^*, \mathbf{u}^*) \in \bar{\mathcal{X}} \times \bar{\mathcal{U}}$  with path planner.
- 5: Initialization:  $t_{\text{optimization}} = 0$
- 6: **while** The tethered space robot does not reach target point  $\mathbf{x}_f$  **do**
- 7:   **if** New space debris is detected by sensors **then**
- 8:     Re-plan the nominal reference trajectory  $(\mathbf{x}^*, \mathbf{u}^*)$  by solving optimization problem (49).
- 9:   **else**
- 10:     **if**  $t - t_{\text{optimization}} = \delta$  **then**
- 11:       Get  $(\mathbf{x}^*(\cdot; \mathbf{x}(t), t), \mathbf{u}^*(\cdot; \mathbf{u}(t), t))$  by solving (50).
- 12:       Compute the controller output  $\mathbf{u}(t) = \mathbf{u}^*(t) + \mathbf{k}(\mathbf{x}(t), \mathbf{x}^*(t); \theta_u)$ .
- 13:       Apply  $\mathbf{u}(t)$  to the tethered space robot.
- 14:       Update  $t_{\text{optimization}} = t$ .
- 15:     **end if**
- 16:   **end if**
- 17: **end while**

**Theorem 2.** Consider the dynamics (10) of tethered space robot, if there are metric  $M(\mathbf{x})$  and controller  $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*)$  satisfying the condition (21) for some constants  $\lambda$ ,  $\bar{\alpha}$  and  $\underline{\alpha}$  over the domain, then the tethered space robot can be deployed to the destination in the presence of bounded disturbances while avoid colliding with all detected space debris.

*Proof.* As there exist metric  $M(\mathbf{x})$  and controller  $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*)$  satisfying the CCM condition (21) over the domain, Lemma 2 shows that the controller  $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*)$  could track any feasible trajectory and the distance between trajectory of perturbed closed-loop dynamics and nominal trajectory of unperturbed dynamics is bounded. Thus, with the controller  $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*)$ , tethered space robot can track the nominal trajectory generated by path planner which considers safety constraints. This completes proof. □

**Remark 3.** As stated above, a valid CCM admits a feedback controller that could track any feasible trajectory and Theorem 2 does not rely on any specific information across different tasks. Therefore, the proposed scheme is also applicable to collision avoidance tasks of other spacecrafts if one can find a valid CCM over the domain.

## IV. SIMULATION RESULTS

In this section, the influences of term  $L_{C_1}$ ,  $L_{C_2}$  and  $L_{RCI}$  on the performance are discussed first. Then the optimal

contracting rate  $\lambda$  and overshoot constant  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  are given by considering  $L_{RCI}$  in (41). Last, the proposed framework is applied to deploy the tethered space robot to the destination while keeping avoid colliding with any space debris.

### A. Implementation Details

The metric  $M(x; \theta_M)$  is modeled as  $M(x; \theta_M) = \underline{\alpha}I + m(x; \theta_M)^T m(x; \theta_M)$  where the  $m(x; \theta_M)$  is a 2-layer networks and each hidden layer contains 128 neurons. In this way, the metric  $M(x; \theta_M)$  is a symmetric matrix satisfying  $M(x; \theta_M) \succeq \underline{\alpha}I$  for all  $x$  and  $\theta_M$ . The controller is designed by  $u(x, x^*, u^*; \theta_u) = u^* + w_2 \cdot \tanh(w_1 \cdot (x - x^*))$ . The weights  $w_1 = w_1(x, x^*; \theta_{u1})$  and  $w_2 = w_2(x, x^*; \theta_{u2})$  are modeled by two 2-layer neural network and each hidden layer contains 128 neurons ( $\theta_M = \{\theta_{u1} + \theta_{u2}\}$ ). The activation functions in the neural networks are all set to hyperbolic tangent function ( $\tanh$ ). The training set contains 130K samples over the data space  $\mathcal{D}$ . The neural networks are trained with batch size of 512 and with the Adam optimizer. All training in this work is implemented with Pytorch [32]. The nonlinear optimization tool CasADi [33] is employed to solve the path planning problem (49) and MPC problem (50) for re-planning path locally.

The offline training is conducted on a Linux Laptop equipped with Intel Core i9-12900H, 32G RAM and a NVIDIA RTX 3060 GPU. The online tracking control and planning algorithm are running on an embedding computer equipped with Intel Core i5-1135G7 and 16GB RAM.

### B. Evaluation of Learned Robust Tracking Controller

In this part, the impacts of the term  $L_{C1}(\theta_M)$ ,  $L_{C2}(\theta_M)$  and  $L_{RCI}(\theta_M, \lambda, \bar{\alpha}, \underline{\alpha})$  on the performance are investigated. The reference trajectory for evaluating the learned robust tracking controller is generated by solving the problem (49) but the safety constraint  $\mathcal{W}_h$  is not considered. The initial and terminal state are set as  $x_0 = [0, 0, -0.99, 1]^T$  and  $x_f = [0, 0, 0, 0]^T$ . The ended time  $t_f$  is equal to 15 rad. The cost function  $g(\cdot)$  in problem (49) is defined as  $g(x(t), u(x)) = 1/2 \cdot [x(t)^T Q x(t) + u(t)^T R u(t)]$  where  $Q = I \in \mathbb{R}^{4 \times 4}$ ,  $R = 1/2 \cdot I \in \mathbb{R}^{2 \times 2}$ . When evaluating of the learned tracking controller, the disturbance in (10) is stochastic noise and the upper bound of it is assumed to be 0.01, i.e.,  $\bar{d} = 0.01$ .

First, we demonstrate that imposing the constraints (33) and (34) is instrumental in finding a valid metric  $M(x; \theta_M)$  that satisfies the CCM condition strictly. The term  $L_{RCI}(\theta_M, \lambda, \bar{\alpha}, \underline{\alpha})$  is not considered in (41) during training neural network and  $\underline{\alpha}, \bar{\alpha}, \lambda$  are chosen as 0.1, 10, 0.5, respectively. The neural networks are trained for 18 epochs. The probability of learned metric  $M(x; \theta_M)$  satisfying CCM condition (21) over the domain with and without constraints (33) and (34), is shown in Fig. 6. It is obvious that the metric trained without constraints (33) and (34) will lead to unsatisfactory performance of a high probability. This makes it not applicable to obtain a valid tracking controller. And in our experiments, training without constraints (33) and (34) of more epochs does not improve the performance significantly. This may be explained as the optimization problem of finding

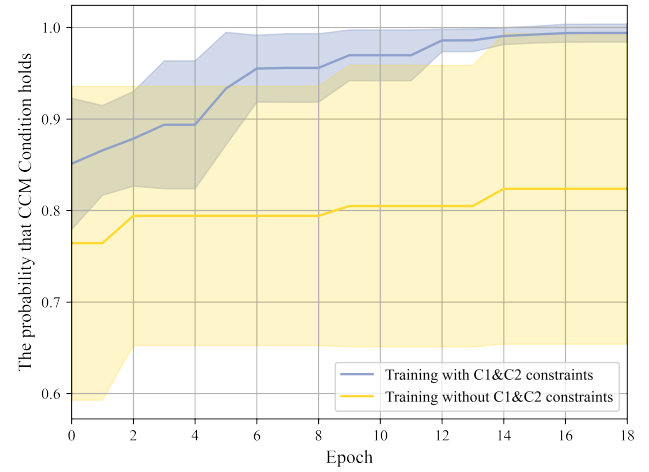


Fig. 6. The probability of learned metric that CCM condition holds. (The closer probability approaches 1, the better.)

a valid metric falling into the local optima sometimes. As shown in Fig. 6, the probability of learned metric  $M(x; \theta_M)$  satisfying CCM condition (21) over the domain with constraints (33) and (34) is very close to 1. In this sense, imposing the constraints (33) and (34) into (41) could provide more guidance for optimization to obtain a valid metric. Note that this does not mean that it is impossible to learn a valid metric with constraint (21) only. However, as shown in this work, learning metric without constraints (33) and (34) will lead unsatisfactory results of a high probability. Thus, in this work, the constraints (33) and (34) are considered for learning a valid metric.

The effects of two parameters contracting rate  $\lambda$  and overshoot constant  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  on tracking performance are investigated here. The tracking results of space tether length with different values of  $\lambda$  and  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  are plotted in Fig. 7. It is clear that the rate of convergence of tracking controller is fastest when  $\lambda = 5$  and the rate of convergence is slowest when  $\lambda = 2.5$ . This shows that the larger of contracting rate  $\lambda$  results the faster convergence to the reference trajectory. However, to guarantee the CCM condition holds, the increase in the value of  $\lambda$  will be accompanied by an increase in the value of  $\sqrt{\bar{\alpha}/\underline{\alpha}}$ , which is verified in Fig. 7. Furthermore, the increase in value of overshoot constant  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  may bring an overshoot as shown in Fig. 7. It is intuitional that a larger overshoot constant  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  results a more obvious overshoot. **Overall**, a larger contracting rate  $\lambda$  enables faster convergence but also leads a larger overshoot constant  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  such that the CCM condition can be held over the domain. There is a trade-off between the contracting rate  $\lambda$  and overshoot constant  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  on the control performance.

To avoid tuning parameters  $\lambda$  and  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  by hand, a term  $L_{RCI}(\theta_M, \lambda, \bar{\alpha}, \underline{\alpha})$  is developed and considered as a part of loss function (41) trying to find  $\lambda$  and  $\sqrt{\bar{\alpha}/\underline{\alpha}}$  optimally and automatically. The more detailed elaboration of considering term  $L_{RCI}(\theta_M, \lambda, \bar{\alpha}, \underline{\alpha})$  in loss function (41) is stated here. Since the metric was constructed by using (37), for the sake of convenience, the lower bound  $\underline{\alpha}$  of metric  $M(x; \theta)$  is

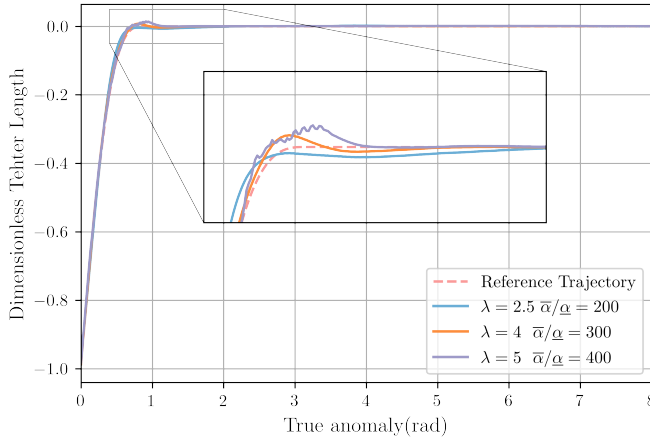


Fig. 7. The trajectories tracked by learned controllers with different values of  $\lambda$  and  $\bar{\alpha}$  ( $\underline{\alpha} = 0.1$ ).

set to a constant during training the neural networks, i.e.,  $\underline{\alpha} = 0.1$  such that condition  $M(x; \theta) \succeq \underline{\alpha} I$  holds for every point over the data space. The user-defined parameter  $\zeta$  is set to  $1 \times 10^{-2}$ , i.e.,  $\zeta = 1 \times 10^{-2}$ . The contraction rate  $\lambda$  and upper bound of metric  $\bar{\alpha}$  are also represented by neural networks for optimization. The neural networks are trained for 400 epochs. As mentioned before, imposing  $L_{RCI}$  may lead inaccurate guidance for searching a valid CCM especially at the beginning of training. To alleviate this issue, we update the weights of  $\lambda$  and  $\bar{\alpha}$  every 10 epochs, which is much slower than updating weights of metric  $M(x; \theta_M)$  and controller  $u(x, x^*, u^*; \theta_u)$ . The optimal value of  $\lambda$  is found by the neural networks which equals to 2.6748 and the upper bound  $\bar{\alpha}$  of metric is 21.3451, i.e.,  $\lambda_{\text{optimal}} = 2.6748$ ,  $\bar{\alpha} = 21.3451$ ,  $\sqrt{\bar{\alpha}/\underline{\alpha}} = 14.6100$ .

In the offline phase, the CCM tracking controller is learned from data such that there is no need to solve a optimization problem of minimizing the geodesic in the online execution phase, which is a superiority of proposed scheme compared to others CCM-based methods. Moreover, the size of “tube” is also optimized for improving the online tracking performance in the presence of disturbances. In this sense, by comprehensively utilizing the prior information of collision avoidance task offline, the learned tracking controller of proposed scheme is ready for online execution for the safe deployment of tethered space robot.

### C. Safe Deployment with Learned Tracking Controller

This part shows that the proposed Algorithm 1 could deploy the tethered space robot from initial position to destination safely. The physical conditions of deployment mission are presented here: the total length of tether is 2000 m, the equivalent mass  $\bar{m}$  is set to 10 kg and the orbit angular velocity is 0.00117 rad/s. The coordinate of in-plane deployment is shown Fig. 8. The initial position where the deployment mission starts is  $(0, -2000)$  m and the destination is  $(0, 0)$  m. The center of circle where all space debris are located in is assumed to be at point  $(300, -1200)$  m, the radius of this circle is set to be 80 m and the size of safety margin is set to 40 m, i.e.,  $h = 40$  m.

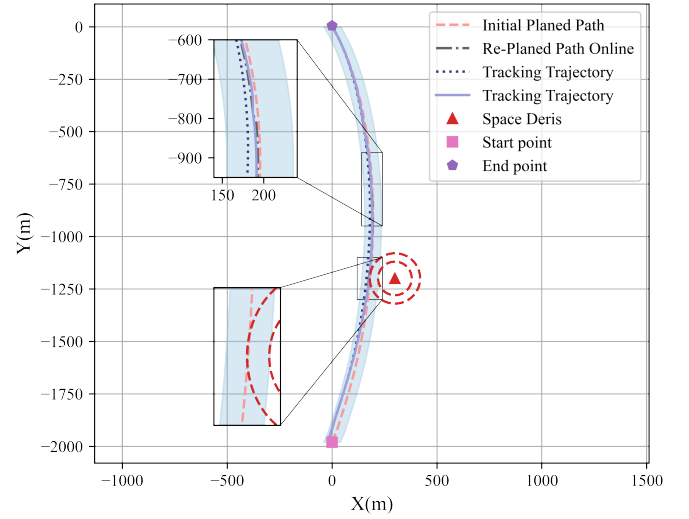


Fig. 8. The trajectories of tethered space robot with different strategies. The red upper triangle is the center of the circle containing space debris. The area between the two red circles is the safety margin as shown in Fig.4. The orange dashed line is the initial planned path by solving the problem (49). The black dash-dot line denotes the re-planned path by solving the problem (50). The black dotted line represents the trajectory by adopting learned tracking controller directly. The lavender solid line is the trajectory by implementing Algorithm 1.

The reference trajectory is generated by solving problem (49). The contraction rate  $\lambda$ , lower bound of metric  $\underline{\alpha}$  and upper bound of metric  $\bar{\alpha}$  are set 2.6784, 0.1, 21.3451, respectively. The disturbance in dynamics (10) is stochastic noise and upper bound of it is set to 0.01, i.e.,  $\bar{d} = 0.01$ . The predictive horizon  $T$  is chosen to be 0.3 s and the MPC problem (50) is solved every 0.15 s, i.e.,  $\delta = 0.15$  s. The positive definite matrices in stage cost (51) are set to  $Q = I \in \mathbb{R}^{4 \times 4}$ ,  $R = 1/2 \cdot I \in \mathbb{R}^{2 \times 2}$ .

The tracking results of adopting the learned tracking controller directly and implementing the Algorithm 1 are depicted in Fig. 8 and Fig. 9. Note that the tethered space robot is assumed to be a mass point in this work. There are three observations. **First**, the path planner considers the constraint (48) into optimization problem such that the generated motion plan can be reliably executed with the learned tracking controller. As stated before, if the size of RCI set is smaller than that of the safety margin for  $t > 0$ , then the actual trajectory of space tethered robot would not collide with any detected space debris. In this sense, with the optimized “tube” offline, one could adjust the size of safety margin of constraint (48) to generate a more safe motion plan. **Further**, utilizing the learned tracking controller directly and Algorithm 1 can both deploy the tethered space robot without any collisions with space debris. The cost of Algorithm 1 and implementing the learned robust tracking controller directly are  $1.234 \times 10^4$  and  $2.896 \times 10^4$ , respectively. The cost of Algorithm 1 is decreased by 57.38% than that of implementing the learned robust tracking controller directly. This shows that the tracking performance of Algorithm 1 is better than that of implementing the learned robust tracking controller directly. However, this requests sufficient computational resources. If there is no additional computation resource to implement the Algorithm

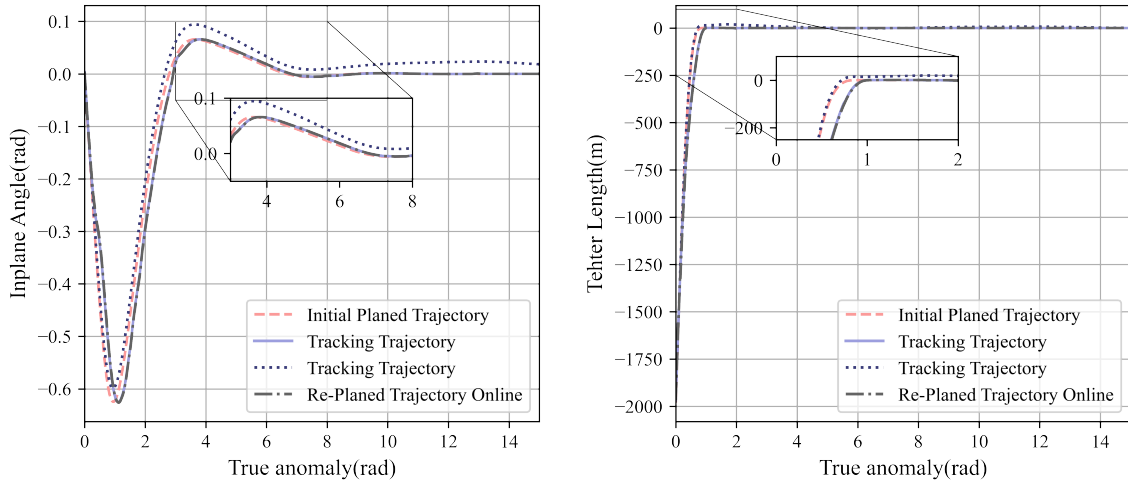


Fig. 9. The inplane angle and tether length trajectories of tethered space robot with different strategies. The red upper triangle is the center of the circle containing space debris. The orange dashed line is the initial planned path by solving the problem (49). The black dash-dot line denotes the re-planned path by solving the problem (50). The black dotted line represents the trajectory by adopting learned tracking controller directly. The lavender solid line is the trajectory by implementing Algorithm 1.

1 in real-time, adopting the learned tracking controller directly is a middle ground since it also provides the convergence and safety guarantees. **Last**, the local information (e.g., realized disturbances) is taken into account when solving the MPC problem (50) such that re-planned path is more applicable than the nominal motion plan. The invariant “tube” (28) would lead conservativeness compared to variant “tube” but it does not need to compute tightened state and control constraint sets online, which could save computational resource for other on-board tasks.

As stated above, the proposed scheme gives two implementations of online execution part. And the offline training part of proposed scheme admits a tracking controller that could execute a nominal motion plan robustly. Thus, this means that it can combine any other motion planner for collision avoidance tasks, which demonstrates the flexibility of proposed scheme.

## V. CONCLUSION

This paper proposes a general scheme to deploy the tethered space robot to destination while keep avoiding colliding with space debris. The proposed scheme decomposes the collision avoidance problem into offline and online components, which offers flexibility for diverse space tasks. In the offline phase, inspired by contraction theory, the robust tracking controller is learned from data. Additionally, the “tube” where the tethered space robot would stay within is also optimized for enhancing the online tracking performance. In the online phase, the motion planner generates the nominal trajectory by considering safety constraints, and then the learned tracking controller executes this motion plan robustly. It is promising to state that the proposed scheme can evolve many other collision avoidance methods by combining different motion planner, which is very flexible. And the proposed scheme could be easily extended to other space related collision avoidance tasks. Future work includes hardware experiments and the derivation of dynamical

“tube” to mitigate the conservativeness of re-planned nominal reference.

## APPENDIX

### A. Proof of Lemma 4

Suppose the solution of problem (50) at time-step  $t_i$  is denoted by  $\bar{\mathbf{u}}^*(\cdot; \mathbf{x}(t_i), t_i) : [t_i, t_i + T] \rightarrow \bar{\mathcal{U}}$ . Apply this control sequence to the nominal system  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u}$ , the following trajectory satisfies

$$\begin{aligned} \bar{\mathbf{x}}^*(\tau; \mathbf{x}(t_i), t_i) &\in \bar{\mathcal{X}} \cap \mathcal{W}_h \quad \tau \in [t_i, t_i + T] \\ \bar{\mathbf{x}}^*(t_i + T; \mathbf{x}(t_i), t_i) &= \mathbf{x}^*(t_i + T) \end{aligned}$$

Because of the RCI set property, the actual state  $\mathbf{x}(t_{i+1})$  of perturbed system at time-step  $t_{i+1} = t_i + \delta$  is guaranteed to lie within the set  $\Omega(\bar{\mathbf{x}}^*(t_{i+1}; \mathbf{x}(t_i), t_i))$  with the tracking controller. Consider the following solution of problem (50) at time-step  $t_{i+1}$

$$\bar{\mathbf{u}}(\tau) = \begin{cases} \mathbf{u}^*(\tau; \mathbf{x}(t_i)) & \tau \in [t_i + \delta, t_i + T] \\ \mathbf{u}^*(\tau) & \tau \in [t_i + T, t_i + \delta + T] \end{cases} \quad (52)$$

The control sequence (52) is a concatenation of the tail section of the last solution with the nominal solution. Thus, under above control law, the trajectory of nominal system at  $[t_i + \delta, t_i + T]$  is equal to the corresponding segment trajectory  $\bar{\mathbf{x}}^*(\cdot; \mathbf{x}(t_i), t_i)$ , i.e.

$$\bar{\mathbf{x}}(\tau; \mathbf{x}(t_i + \delta), t_i + \delta) = \bar{\mathbf{x}}^*(\tau; \mathbf{x}(t_i), t_i) \quad \tau \in [t_i + \delta, t_i + T] \quad (53)$$

It is obvious that this segment of trajectory is satisfied with the constraint of problem (50) and the endpoint of this segment satisfies (terminal constraint (50f))

$$\bar{\mathbf{x}}(t_i + T; \mathbf{x}(t_i + \delta), t_i + \delta) = \mathbf{x}^*(t_i + T) \quad (54)$$

The trajectory of nominal system at  $[t_i + t, t_i + \delta + T]$  is the nominal path plan  $\mathbf{x}^*$ . Therefore, the solution (52) is feasible at time-step  $t_{i+1}$ , which proves the recursive feasibility of problem (50).

### B. Detailed results of training neural networks

We conducted two series of experiments in the simulation section.

1) In the first series of experiments for demonstrating benefits of imposing constraints (33) and (34), there are two cases: “Training C1&C2 constraints” and “Training without C1&C2 constraints”. The loss function of the first case is defined by

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N [L_{\text{ccm}}(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{u}_i^*; \theta_M, \theta_u) + L_{C_1}(\mathbf{x}_i; \theta_M) + L_{C_2}(\mathbf{x}_i; \theta_M) + L_M(\mathbf{x}_i; \theta_M)]$$

and the loss function of second case is defined by

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N [L_{\text{ccm}}(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{u}_i^*; \theta_M, \theta_u) + L_M(\mathbf{x}_i; \theta_M)]$$

Note that since the matrix  $\mathbf{B}$  of (10) has sparse representation (42), the term  $L_{C_2}$  is always equal to zero. For each case, we conducted the training for five times. The value of loss function is plotted in Fig. 10(a). It shows that training without constraints (33) and (34) is not stable and the value of loss function does not converge to zero. After training for 18 epochs, the average probability of 5 trained metrics  $\mathbf{M}(\mathbf{x}; \theta_M)$  satisfying CCM condition over the domain is around 0.82 and the variance is very large (see yellow curve and region in Fig. 6). To address this problem, we imposing constraints (33) and (34) into loss function  $\mathcal{L} = \frac{1}{N} \sum_{i=1}^N [L_{\text{ccm}} + L_{C_1} + L_{C_2} + L_M]$ . In this case, after training 18 epochs, the average probability of 5 trained metrics  $\mathbf{M}(\mathbf{x}; \theta_M)$  satisfying CCM condition over the domain is around 0.996 and the variance is very small (see blue curve and region in Fig. 6).

The above results indicate that imposing constraints (33) and (34) is helpful for obtaining a valid metric. It is promising to state that, in the first case (“Training with C1&C2 constraints”), the probability of trained metric  $\mathbf{M}(\mathbf{x}; \theta_M)$  satisfying CCM condition over the domain is not strictly equal to 1 but very close to 1. This means the inequality (21) does not hold for all points in domain. However, the learned controller can still track the reference trajectories convergently (see the results of Fig. 7 - Fig. 9). Therefore, for the trained metric whose probability satisfying CCM conditions over the domain is very close to 1 but not strictly equal to 1, we still regard this metric as a valid metric.

2) In the second series of experiments, the term  $L_{\text{RCI}}$  is considered as a part of loss function. And we also conducted the training for five times. The value of loss function is plotted in Fig. 10(b). Note that the value of loss function does not converge to zero as the term  $L_{\text{RCI}}$  does not equal to zero.

### REFERENCES

- [1] M. P. Cartmell and D. J. McKenzie, “A review of space tether research,” *Progress in Aerospace Sciences*, vol. 44, no. 1, pp. 1–21, Jan. 2008.
- [2] P. Huang, F. Zhang, L. Chen, Z. Meng, Y. Zhang, Z. Liu, and Y. Hu, “A review of space tether in new applications,” *Nonlinear Dynamics*, vol. 94, no. 1, pp. 1–19, Oct. 2018.
- [3] J. Kang and Z. H. Zhu, “Passivity-based model predictive control for tethered despin of massive space objects by small space tug,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 2, pp. 1239–1248, Apr. 2023.

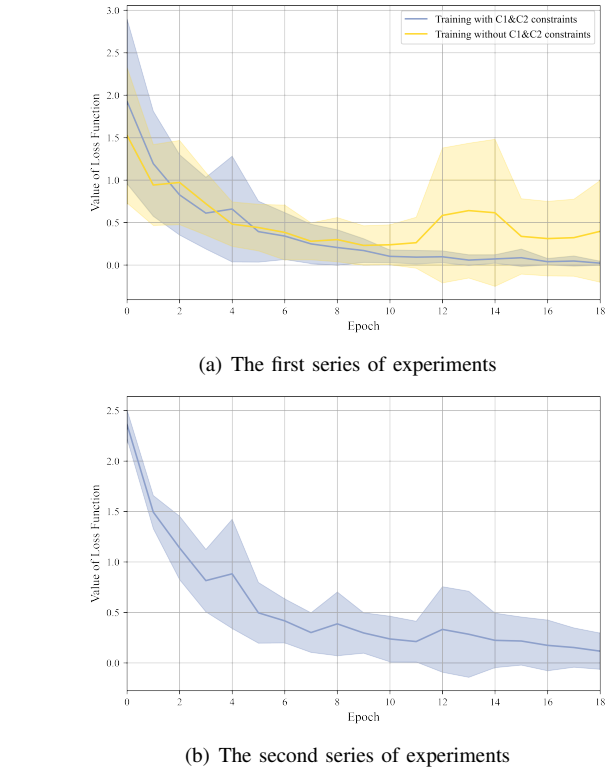


Fig. 10. The value of loss function.

- [4] “Space debris by the numbers.” [Online]. Available: [https://www.esa.int/Space\\_Safety/Space\\_Debris/Space\\_debris\\_by\\_the\\_numbers](https://www.esa.int/Space_Safety/Space_Debris/Space_debris_by_the_numbers)
- [5] S. Patnala and A. Abdin, “Spacecraft collision avoidance: Data management, risk assessment, decision planning models and algorithms,” in *Space Data Management*, A. Cortesi, Ed. Singapore: Springer Nature, 2024, pp. 15–45.
- [6] D. González, J. Pérez, V. Milanés, and F. Nashashibi, “A review of motion planning techniques for automated vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.
- [7] F. Celani and D. Lucarelli, “Spacecraft attitude motion planning using gradient-based optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 1, pp. 140–145, 2020.
- [8] E. Frazzoli, “Quasi-random algorithms for real-time spacecraft motion planning and coordination,” *Acta Astronautica*, vol. 53, no. 4, pp. 485–495, Aug. 2003.
- [9] G. Sun and Z. H. Zhu, “Fractional-order tension control law for deployment of space tether system,” *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, pp. 2057–2062, Nov. 2014.
- [10] H. Wen, Z. H. Zhu, D. Jin, and H. Hu, “Tension control of space tether via online quasi-linearization iterations,” *Advances in Space Research*, vol. 57, no. 3, pp. 754–763, Feb. 2016.
- [11] P. Huang, F. Zhang, J. Cai, D. Wang, Z. Meng, and J. Guo, “Dexterous tethered space robot: Design, measurement, control, and experiment,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 3, pp. 1452–1468, Jun. 2017.
- [12] S. Xu, G. Sun, Z. Ma, and X. Li, “Fractional-order fuzzy sliding mode control for the deployment of tethered satellite system under input saturation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 2, pp. 747–756, Apr. 2019.
- [13] Z. Ma and P. Huang, “Discrete-time sliding mode control for deployment of tethered space robot with only length and angle measurement,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 585–596, Feb. 2020.
- [14] X. Li, G. Sun, and C. Xue, “Fractional-order deployment control of space tethered satellite via adaptive super-twisting sliding mode,” *Aerospace Science and Technology*, vol. 121, p. 107390, Feb. 2022.
- [15] X. Li, G. Sun, Z. Kuang, and S. Han, “Nonlinear predictive optimization for deploying space tethered satellite via discrete-time fractional-order

sliding mode,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 5, pp. 4517–4526, Oct. 2022.

- [16] Z. Ma, X. Duan, X. Liu, Y. Zhang, and P. Huang, “Human-assisted regulation of deployment of tethered space robot via feasibility condition optimization and fast logarithmic sliding mode,” *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–15, 2024.
- [17] WINFRIED. Lohmiller and J.-J. E. Slotine, “On contraction analysis for non-linear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, Jun. 1998.
- [18] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [19] L. Wei, R. McCloy, and J. Bao, “Control contraction metric synthesis for discrete-time nonlinear systems,” *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 661–666, Jan. 2021.
- [20] S. Singh, B. Landry, A. Majumdar, J.-J. Slotine, and M. Pavone, “Robust feedback motion planning via contraction theory,” *The International Journal of Robotics Research*, vol. 42, no. 9, pp. 655–688, Aug. 2023.
- [21] I. R. Manchester and J.-J. E. Slotine, “Output-feedback control of nonlinear systems using control contraction metrics and convex optimization,” in *2014 4th Australian Control Conference (AUCC)*, Nov. 2014, pp. 215–220.
- [22] D. Sun, S. Jha, and C. Fan, “Learning certified control using contraction metric,” in *Proceedings of the 2020 Conference on Robot Learning*. PMLR, Oct. 2021, pp. 1519–1539.
- [23] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 211–216, Jan. 2021.
- [24] N. Rezazadeh, M. Kolarich, S. S. Kia, and N. Mehr, “Learning contraction policies from offline data,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2905–2912, Apr. 2022.
- [25] S. M. Richards, J.-J. Slotine, N. Azizan, and M. Pavone, “Learning control-oriented dynamical structure from data,” in *Proceedings of the 40th International Conference on Machine Learning*. PMLR, Jul. 2023, pp. 29 051–29 062.
- [26] S. Zololi, S. Janny, and M. Giaccagli, “Deep learning-based output tracking via regulation and contraction theory,” *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 8111–8116, Jan. 2023.
- [27] S. Singh, S. M. Richards, V. Sindhwani, J.-J. E. Slotine, and M. Pavone, “Learning stabilizable nonlinear dynamics with contraction-based regularization,” *The International Journal of Robotics Research*, vol. 40, no. 10–11, pp. 1123–1150, Sep. 2021.
- [28] A. Virmaux and K. Scaman, “Lipschitz regularity of deep neural networks: Analysis and efficient estimation,” in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.
- [29] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, “Efficient and accurate estimation of lipschitz constants for deep neural networks,” in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.
- [30] F. Latorre, P. Rolland, and V. Cevher, “Lipschitz constant estimation of neural networks via sparse polynomial optimization,” in *International Conference on Learning Representations*, Sep. 2019.
- [31] X. Shao, B. Zhang, J. G. Romero, B. Fan, Q. Hu, and D. Navarro-Alarcon, “Collision-free navigation of wheeled mobile robots: An integrated path planning and tube-following control approach,” Dec. 2023.
- [32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.
- [33] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “Casadi: A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, Mar. 2019.



**Ao Jin** received the B.S. degree in Automation from Northwestern Polytechnical University in 2021. He is currently pursuing the Ph.D degree in School of Automation, Northwestern Polytechnical University. His research interests include the data-driven control, learning-based control and tethered system.



**Fan Zhang** (S'16–M'18) received the M.S. and Ph.D. in Navigation, Guidance and Control, from School of Astronautics, Northwestern Polytechnical University in 2012 and 2017, respectively, and received B.S. in Detection, Guidance and Control Technology in 2009. She is currently an associate research professor in School of Astronautics, Northwestern Polytechnical University. Her research focuses on the dynamic and control of tethered robotics, intelligent control, and constrained multi-agents system.



**Ganghui Shen** (S'16–M'18) received the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2020. He was a visiting scholar with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2019. He is currently an Associate Professor with the School of Astronautics, Northwestern Polytechnical University. His research interests include tethered spacecraft control, sliding mode control, and constrained control.



**Yifeng Ma** received a B.S. in Aerospace Engineering from Northwestern Polytechnical University, China in 2022. He is pursuing an M.S. degree in Aeronautical and Astronautical Science and Technology with the School of Astronautics, at Northwestern Polytechnical University. His research interests include robotics, game theory, formation control, learning-based control and multi-agent systems.



**Panfeng Huang** (S'01–M'05–SM'17) received the B.S. degree in test and measurement technology and the M.S. degree in navigation guidance and control from Northwestern Polytechnical University, Xi'an, China, in 1998 and 2001, respectively, and the Ph.D. degree in automation and robotics from The Chinese University of Hong Kong, Hong Kong, in 2005. He is currently a Professor with the School of Astronautics, Northwestern Polytechnical University, and the National Key Laboratory of Aerospace Flight Dynamics, and the Director of the Research Center

for Intelligent Robotics, Northwestern Polytechnical University. His current research interests include tethered space robotics, intelligent control, machine vision, and space debris removal.