Cooperative Game Theory based Formation Control for Tethered Space Net Robot

Yifeng Ma[a], Yizhai Zhang[a], Ya Liu[b], Panfeng Huang[a] and Fan Zhang[a]

[a]*Research Center of Intelligent Robotics, School of Astronautics, Northwestern Polytechnical University, Xi'an, China,*
[b]*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China,*

ARTICLE INFO

Keywords:
Tethered Space Net Robot
Formation control
Differential cooperative games
Learn-based control

ABSTRACT

The traditional control methods for Tethered Space Net Robot (TSNR) typically treat the tension force induced by the net as a disturbance and employ passive suppression for compensation. However, these approaches overlook the intrinsic nature of the net dynamics. When one Maneuverable Unit (MU) maneuvers, it generates a tension force on the net that is transmitted to other MUs. This force not only affects the control accuracy of other MUs but also has a positive effect. In this paper, an active tension force control strategy is proposed to reveal this kind of interaction and maximize its advantage. Specifically, a neutral network estimator is designed to capture the hysteresis relationship in which MUs influence each other by transmitting forces through the net. Furthermore, to achieve the cooperative completion of tasks, a game control framework is established to optimize the control input and tension force collectively. Through prediction and optimization, MUs actively manage their impacts on each other, thereby actively controlling the influence of tension force on the tracking errors of others. Finally, numerical simulations are conducted to showcase the effectiveness of the proposed scheme.

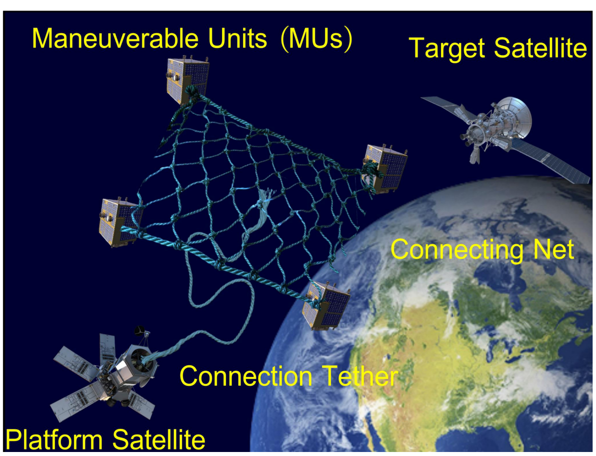

Fig. 1: TSNR system.

1. Introduction

On-orbit services, such as debris removal and the capture of noncooperative targets, are emerging as promising trends in future space development [?]. Over the past few decades, various flexible tethered space systems have been proposed to facilitate capture missions. These systems, such as the Tethered Space Net (TSN) [?], Tethered Space Robot (TSR) [?], Tethered Space Net Robot (TSNR) [?], and other [?], utilize a flexible tether to connect the capturing device and the target. Tethered Space Net Robot (TSNR) emerges as an innovative solution for active space debris capture and removal, as illustrated in Fig.**??**. Its expansive envelope and straightforward capture method render it an appealing choice for such tasks. However, control challenges arise when attempting to capture debris with the flexible and elastic underactuated net.

In recent years, researchers have extensively explored control challenges associated with TSNR for space debris capture tasks. For example, the study in [?] formulated formation-keeping control strategies, employing two artificial potential functions to preserve the TSNR's configuration while adhering to relative distance constraint. Additionally, Zhang et al. [?] introduced a fuzzy-based adaptive super-twisting sliding mode control to estimate and suppress the complex oscillations of the TSNR. Liu et al. [?] designed a capture configuration optimization method and an observer base time-varying formation tracking control algorithm, considering contact dynamics with realistic space debris.

However, prior studies have primarily focused on overarching control objectives, overlooking the importance of individual subjective initiative. Treating the extra tension force caused by other MUs as mere perturbations fails to adequately capture dynamic interactions among MUs. Game theory serves as a fitting analytical method for addressing interaction formation issues within multi-agent systems. Jiang et al. [?] introduced the cooperative error into the cost function to study the consistency of the formation. Additionally, Li et al. [?] explored a distributed game strategy for the formation control of multi-spacecraft, introducing a worst-case Nash strategy against the disturbance defined as a player. The study in [?] investigated the problem of modular robots cooperating in handling a large space structure by using the framework of cooperative game, and designed a coordinated compensatory control mechanism using the principle of model predictive control, which ensures that it is able to track the desired optimal handling trajectory. Liu et al. [?] extended the application of differential graphical

*Corresponding author at: Research Center of Intelligent Robotics, School of Astronautics, Northwestern Polytechnical University; E-mail address: fzhang@nwpu.edu.cn

game control to the coupled non-linear dynamics of the T-MRUAVs system by introducing a sophisticated framework, capturing the collaborative and competitive nature among UAVs to achieve efficient and secure stability control of the system.

Nonetheless, current game theory based methods designed for multi-agent systems are not directly applicable to TSNR. In existing work, the game model typically has a deterministic form, allowing for a visual depiction of the effect of each individual in the multi-agent system on the game state. However, MUs in TSNR interact with each other by transferring tension forces through the net, raising the question of how to describe this implicit and time-delayed relationship. In this work, a neural network based tension force estimator is proposed to estimate and predict this complex relationship among MUs. For treating the tension force as a perturbation, both the fixed-time [**? ?**] or finite-time [**? ?**] state observer can effectively estimate the perturbation, while sensors or camera can be employed to measure the external force. Recently, machine-learning scheme has emerged as attractive approaches for modeling complex dynamics or external forces [**? ? ?**], owing to their learning and fitting ability. However, these methods often struggle to describe this kind of hysteresis relationship between tension force and individual state.

To address the aforementioned problems, a novel cooperative game theory based formation control scheme for TSNR is proposed in this paper. In this scheme, the tension force of the net is no longer considered as a mere perturbation. Instead, forces acting on other MUs generated by one MU's maneuver are concretely characterized. The advantages of this interaction are further explored and exploited, enhancing formation control capabilities. The main contributions of this work are outlined as follows:

1. Tension forces are transmitted among MUs through the net like ocean waves, where these forces are not only influenced by the current state but also by the past states. To effectively depict this kind of hysteresis interaction among MUs, a Deep Neutral Network (DNN) based tension force estimator is proposed in this paper to estimate and predict tension force.
2. To mitigate the negative effects and capitalize on the positive effects of tension force, a game theory based active tension force control framework is established among MUs. Unlike traditional passive compensation formation control methods [**? ?**], in this scheme, one MU can actively manage the impact of tension force on the tracking errors of other MUs, thus being able to cooperate better in the capture task.

The remainders are stated as follows. Section 2 presents the mission and dynamic model of TSNR. In Section 3, a novel game based formation control strategy is proposed. In Section 4, the proposed scheme is validated using numerical simulations and results. Section 5 concludes this paper.

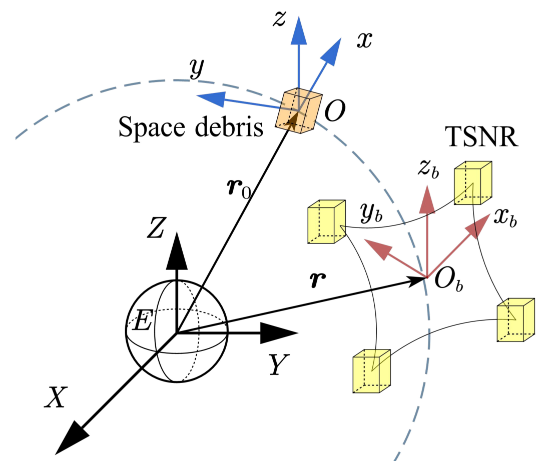

Fig. 2: Coordinate frames of TSNR system.

2. Problem formuation

2.1. Mission description

This paper focuses on the application of TSNR to complete an on-orbit debris removal service to capture space debris or failed satellites. First, the net is folded and four MUs are clustered together. When a target is detected, the four MUs carrying the net are ejected from the platform satellite. Then four MUs manoeuvre to unfold the net and approach the target, maintaining an easy-capture configuration. Finally, four MUs drive the net to envelope the target and complete the capture mission.

2.2. Dynamic model of TSNR

To describe the movements of TSNR and space debris, the following coordinate frames are first introduced (as depicted in Fig. **??**). The Earth-Centered Inertial (ECI) frame, denoted by $EXYZ$, is located at the center of the Earth. The non-inertial Local Vertical Local Horizontal (LVLH) orbit coordinate frame, denoted by $Oxyz$, is located at the target. TSNR only undergoes translation relative to the target, and the relative distance is much smaller than the orbital radius. Furthermore, the body-fixed frame denoted by $O_b x_b y_b z_b$ and located at the center of TSNR, is parallel to the orbital coordinate frame.

The relative dynamics equation between space debris and TSNR in inertial frame can be expressed by the following equation:

$$\begin{cases} \ddot{\boldsymbol{r}}_0 = -\frac{\mu \boldsymbol{r}_0}{r_0^3} \\ \ddot{\boldsymbol{r}}_{ij} = -\frac{\mu \boldsymbol{r}_{ij}}{r_{ij}^3} + \boldsymbol{u}_{ij} + \boldsymbol{T}_{ij} \end{cases} \tag{1}$$

where $\boldsymbol{r}_0$ and $\boldsymbol{r}_{ij}$ denote the position of space debris and any net node of TSNR respectively, with $r_0 = \left\|\boldsymbol{r}_0\right\|$ and $r_{ij} = \left\|\boldsymbol{r}_{ij}\right\|$; μ is the gravitation constant; $\boldsymbol{u}_{ij}$ and $\boldsymbol{T}_{ij}$ represent control input and tension force on node ij. The dynamic model of node ij which expresses the relative motion in

LVLH frame can be written as:

$$\begin{cases} \ddot{x}_{ij} - 2\omega\dot{y}_{ij} - \omega^2 x_{ij} - \dot{\omega}y_{ij} = \frac{2\mu x_{ij}}{r_0^3} + u_{ij}^x + T_{ij}^x \\ \ddot{y}_{ij} + 2\omega\dot{x}_{ij} - \omega^2 y_{ij} + \dot{\omega}x_{ij} = -\frac{\mu y_{ij}}{r_0^3} + u_{ij}^y + T_{ij}^y \\ \ddot{z}_{ij} = -\frac{\mu z_{ij}}{r_0^3} + u_{ij}^z + T_{ij}^z \end{cases} \tag{2}$$

where ω is the orbital frequency of the debris; $\boldsymbol{u}_{ij} = \left[u_{ij}^x, u_{ij}^y, u_{ij}^z\right]^{\mathrm{T}}$ and $\boldsymbol{T}_{ij} = \left[T_{ij}^x, T_{ij}^y, T_{ij}^z\right]^{\mathrm{T}}$. Define the node ij's state as $\boldsymbol{x}_{ij} = \left[x_{ij}, y_{ij}, z_{ij}, \dot{x}_{ij}, \dot{y}_{ij}, \dot{z}_{ij}\right]^{\mathrm{T}}$, and model (**??**) can be rearranged as:

$$\dot{\boldsymbol{x}}_{ij} = \boldsymbol{A}(r_0, \omega, \dot{\omega})\boldsymbol{x}_{ij} + \boldsymbol{B}(\boldsymbol{u}_{ij} + \boldsymbol{T}_{ij}) \tag{3}$$

What makes dynamics of TSNR most different from other space robots is the tension force T_{ij} generated by the flexible net. Inside the net, each mesh edge between two adjacent nodes is simplified as spring-damping model. Therefore, the tension force of node ij is calculated as:

$$\begin{aligned} \boldsymbol{T}_{ij} &= \sum_{kh \in N_{ij}} \boldsymbol{T}_{ij-kh} \\ &= \begin{cases} \left(\frac{EA}{l_n}\Delta l_{ij-kh} + c\dot{l}_{ij-kh}\right)\hat{\boldsymbol{l}}_{ij-kh} & \Delta l_{ij-kh} \geqslant 0 \\ 0 & \Delta l_{ij-kh} < 0 \end{cases} \end{aligned} \tag{4}$$

where N_{ij} is the set of nodes connected to node ij; $\boldsymbol{T}_{ij-kh}$ is the tension force between node ij and kh; l_n and l_{ij-pq} denote the length of mesh edge and actual length of the two nodes; $\Delta l_{ij-kh} = l_{ij-kh} - l_n$ is the elongation of mesh edge; E is Young's modulus of the net material; A represents the horizontal area of mesh edge; c is damping coefficient; $\hat{\boldsymbol{l}}_{ij-kh}$ is unit vector from node ij to kh.

Remark 1. TSNR is a underactuated system, which can only be controlled indirectly by four MUs on the corner of the net. These MU nodes are noted as MU $1-4$, and dynamics of MU i can be obtained as:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{A}(r_0, \omega, \dot{\omega})\boldsymbol{x}_i + \boldsymbol{B}(\boldsymbol{u}_i + \boldsymbol{T}_i) \tag{5}$$

Obviously tension force can have a significant impact on MUs and configuration of the net, so the objective of control is to avoid excessive tension forces, and to maintain microtension for net stabilisation. The control method will be discussed in the next section.

2.3. Graph theory

In TSNR system, the directed graph $\mathcal{G}$ represents the network structure facilitating information exchange among MUs. The adjacency matrix $\boldsymbol{\mathcal{A}} = \left[a_{ij}\right] \in \mathbb{R}^{n\times n}$, where $a_{ij} > 0$ if MU i can directly receive infromation from its neighbor MU j, and $a_{ij} = 0$ otherwise. The set of neighbors of MU i is defined as N_i. And the indegree matrix $\boldsymbol{\mathcal{D}} = \mathrm{diag}(d_{ii})$ is fromed by $d_{ii} = \sum_{j=1}^n a_{ij}$. Then the graph Laplacian matrix is defined as $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{A}}$. In addition, for the problem of trajectory tracking, introduce $\boldsymbol{\mathcal{B}} = \mathrm{diag}(b_1, \cdots, b_n) \in \mathbb{R}^{n\times n}$, where $b_i > 0$ if MU i can obtain a desired trajectory.

3. Controller design

In this section, a formation tracking control strategy based on differential cooperative game is designed for TSNR (as depicted in Fig. **??**), including two parts: tension force estimator and game based controller. In each time step, the neutral estimator estimates and predicts tension force in real time based on the states of adjacent MUs. Then MUs optimize their own objective that contains predicted tension force, by communicating and gaming with each other.

3.1. Tension force neutral network estimator

According to dynamics of TSNR (**??**), MU i's tension force T_i is associated with the nodes near MU i, which are also affected by their neighbors. Exploring further out of nodes, it can be found that the four MUs interact with each other by transferring forces through the net. Therefore,T_i is caused by the manoeuvres of adjacent MUs, and can be expressed as:

$$\boldsymbol{T}_i = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{x}_{-i}, \boldsymbol{u}_i, \boldsymbol{u}_{-i}) \tag{6}$$

where $\boldsymbol{x}_{-i}$ and $\boldsymbol{u}_{-i}$ are the states and control inputs of MU i's neighbors; $\boldsymbol{f}$ is a function that represents mapping from state and control to tension force.

This function $\boldsymbol{f}$ is strongly nonlinear, and although an analytical solution can be obtained by net model (**??**), in practice it is unrealistic to obtain the state of each node to solve it. Meanwhile, the function has a very noticeable hysteresis. The vibrations caused by one particular MU will be transmitted to other neighboring MUs through the net like waves. Thus, the present force $T_{i,k}$ is related to the past state, and the present state will produce a new sequence of tension force in the future. Therefore, the function (**??**) can be deduced as :

$$\boldsymbol{T}_i\left(t_k + N_t\Delta t | t_k\right) = \boldsymbol{\Phi}\left(\boldsymbol{X}_i\left(t_k - N_t\Delta t | t_k\right)\right) \tag{7}$$

where N_t is predict horizon of the estimator; $\boldsymbol{T}_i\left(t_k + N_t\Delta t | t_k\right)$ denotes tension force sequence $\left[T_{i,k}^{\mathrm{T}}, T_{i,k+1}^{\mathrm{T}}, \cdots, T_{i,k+N_t-1}^{\mathrm{T}}\right]^{\mathrm{T}}$; $\boldsymbol{X}_i\left(t_k - N_t\Delta t | t_k\right)$ represents state sequence $\left[\boldsymbol{X}_{i,k-N_t+1}^{\mathrm{T}}, \cdots, \boldsymbol{X}_{i,k-1}^{\mathrm{T}}, \boldsymbol{X}_{i,k}^{\mathrm{T}}\right]^{\mathrm{T}}$; the state is defined as $\boldsymbol{X}_{i,k} = \left[\boldsymbol{x}_{i,k}^{\mathrm{T}}, \boldsymbol{x}_{-i,k}^{\mathrm{T}}, \boldsymbol{u}_{i,k}^{\mathrm{T}}, \boldsymbol{u}_{-i,k}^{\mathrm{T}}\right]^{\mathrm{T}}$.

Furthermore, due to DNN's remarkable ability of function approximation, a DNN parameterized by weights $\boldsymbol{\theta} = W^1, \cdots, W^{L+1}$ is raised to approximate the nolinear and time-delayed function $\boldsymbol{\Phi}$:

$$\boldsymbol{\Phi} = W^{L+1}\phi(W^L(\cdots\phi(W^1\boldsymbol{X}_i)\cdots)) \tag{8}$$

where ϕ is the ReLU activation function. To estimate and predict tension force associated with present and past state, loss function L is formulated as:

$$\begin{aligned} L\left(\boldsymbol{T}_i, \hat{\boldsymbol{T}}_i\right) = &\alpha L_e\left(\boldsymbol{T}_i\left(t_k\right), \hat{\boldsymbol{T}}_i\left(t_k\right)\right) \\ &+ \beta L_p\left(\boldsymbol{T}_i\left(t_k + N_t\Delta t | t_k\right), \hat{\boldsymbol{T}}_i\left(t_k + N_t\Delta t | t_k\right)\right) \end{aligned} \tag{9}$$

figure/overview-eps-converted-to.pdf

Fig. 3: System Overview.

where $\hat{\boldsymbol{T}}_i$ is estimate of tension force $\boldsymbol{T}_i$; α and β are positive hyperparameters. Loss function $L\left(\boldsymbol{T}_i, \hat{\boldsymbol{T}}_i\right)$ consists of two parts: estimation L_e and prediction L_p. The purpose of the first part is to estimate exact tension force at every time step, having the following form:

$$L_e = \begin{cases} (1-\gamma)\left\|\boldsymbol{T}_i\left(t_k\right) - \hat{\boldsymbol{T}}_i\left(t_k\right)\right\| & \boldsymbol{T}_i\left(t_k\right) > \hat{\boldsymbol{T}}_i\left(t_k\right) \\ \gamma\left\|\boldsymbol{T}_i\left(t_k\right) - \hat{\boldsymbol{T}}_i\left(t_k\right)\right\| & \boldsymbol{T}_i\left(t_k\right) < \hat{\boldsymbol{T}}_i\left(t_k\right) \end{cases} \tag{10}$$

where γ is a positive constant. Because excessive tension force can lead to net breakage or even system collapse, overestimation of tension force is permissible, while underestimation is not. Therefore γ is set in $(0, 0.5)$ to penalise underestimation.

Next, the second part is aimed to predict the change in tension force over a period of time. This can be expressed as:

$$L_p = \sum_{n=1}^{N_t} \eta_n \left\|\boldsymbol{T}_i\left(t_k + n\Delta t\right) - \hat{\boldsymbol{T}}_i\left(t_k + n\Delta t\right)\right\| \tag{11}$$

where $\eta_n \in (0,1)$ is a decay constant. This part of loss function focuses on forecasting tension force errors over a period time in the future. Due to the temporal delay between MUs' interactions, the current state $\boldsymbol{X}_i(t_k)$ exerts a lesser influence on the force at next step $\boldsymbol{T}_i(t_k + \Delta t)$. Instead, it exerts a more pronounced effect on the force at a slightly later time. So the value of η_n depends on the time.

Finally, based on the predicted force $\hat{\boldsymbol{T}}_i$ obtained from the proposed estimator (**??**), MU i's dynamic can be rearranged as:

$$\dot{\hat{\boldsymbol{x}}}_i = \boldsymbol{A}(r_0, \omega, \dot{\omega})\hat{\boldsymbol{x}}_i + \boldsymbol{B}(\boldsymbol{u}_i + \hat{\boldsymbol{T}}_i) \tag{12}$$

where $\hat{\boldsymbol{x}}$ is the estimated state. Define $\hat{\boldsymbol{u}}_i = \boldsymbol{u}_i + \hat{\boldsymbol{T}}_i$, predicted dynamic of MU can be reformulated as:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\hat{\boldsymbol{u}}_i \tag{13}$$

Remark 2. During the data collection process, multiple trajectories of TSNR are given and the state of each moment $\boldsymbol{X}_i(t_k)$ is recorded. Further the actual tension force $\boldsymbol{T}_i$ can be obtained offline by states information and (**??**). In order to facilitate the analysis of these data, the input and output data were normalised. This was necessary due to the large temporal span of the state, the numerous dimensions of the state, and the considerable disparity between position data and velocity ones.

3.2. Cooperative game based formation controller

The objective of each MU is not only to track its own trajectory, but also to maintain a certain formation scaling, so that the net remains well configured. Define tracking error $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{x}_i^d$, where $\boldsymbol{x}_i^d$ is the desired capture trajectory according to [**?**]. Then, the cooperative consensus error is:

$$\boldsymbol{e}_i = \sum_{j\in N_i} a_{ij}\left(\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\right) + b_i\tilde{\boldsymbol{x}}_i \tag{14}$$

Define global error $\tilde{\boldsymbol{x}} = \left[\tilde{\boldsymbol{x}}_1^{\mathrm{T}}, \tilde{\boldsymbol{x}}_2^{\mathrm{T}}, \cdots, \tilde{\boldsymbol{x}}_n^{\mathrm{T}}\right]^{\mathrm{T}}$, the derivative of (**??**) has the following form:

$$\begin{aligned} \dot{\boldsymbol{e}}_i &= \left(\left(\boldsymbol{\mathcal{L}}_i + \boldsymbol{\mathcal{B}}_i\right) \otimes \boldsymbol{I}\right)\dot{\tilde{\boldsymbol{x}}} \\ &= \left(\left(l_{ii} + b_{ii}\right) \otimes \boldsymbol{I}\right)\dot{\tilde{\boldsymbol{x}}}_i + \sum_{j\in N_i}\left(\left(l_{ij} + b_{ij}\right) \otimes \boldsymbol{I}\right)\dot{\tilde{\boldsymbol{x}}}_j \\ &= \boldsymbol{L}_i\dot{\tilde{\boldsymbol{x}}}_i + \sum_{j\in N_i}\boldsymbol{L}_{ij}\dot{\tilde{\boldsymbol{x}}}_j \\ &= \boldsymbol{L}_i(\boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\hat{\boldsymbol{u}}_i - \dot{\boldsymbol{x}}_i^d) + \sum_{j\in N_i}\boldsymbol{L}_{ij}(\boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{B}\hat{\boldsymbol{u}}_j - \dot{\boldsymbol{x}}_j^d) \end{aligned} \tag{15}$$

where $\boldsymbol{\mathcal{L}}_i$ is the i th row vector of the Laplacian matrix $\boldsymbol{\mathcal{L}}$, i.e., $\boldsymbol{\mathcal{L}}_i = \left[l_{i1}, \cdots, l_{in}\right]$. Similarly, $\boldsymbol{\mathcal{B}}_i = \left[b_{i1}, \cdots, b_{in}\right]$. $\boldsymbol{L}_i = \left(l_{ii} + b_{ii}\right) \otimes \boldsymbol{I}$ and $\boldsymbol{L}_{ij} = \left(l_{ij} + b_{ij}\right) \otimes \boldsymbol{I}$. According to (**??**), apparently there is a cooperative relationship among MUs.

Then, by taking each MU as a game player, consensus error $\boldsymbol{e}_i$ as the game state and $\hat{\boldsymbol{u}}_i$ as the control strategy, a n-person cooperative game is established to design the distributed formation controller for TSNR. The cost function of each player is formulated as:

$$J_i = \int_{t_0}^{t_f}\left(\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{Q}_i\boldsymbol{e}_i + \hat{\boldsymbol{u}}_i^{\mathrm{T}}\boldsymbol{R}_i\hat{\boldsymbol{u}}_i + \sum_{j\in N_i}\hat{\boldsymbol{u}}_j^{\mathrm{T}}\boldsymbol{R}_{ij}\hat{\boldsymbol{u}}_j\right)\mathrm{d}t \tag{16}$$

where t_0 and t_f are the initial and final time. All weighting matrices are symmetric and constant. And $\boldsymbol{Q}_i > 0$, $\boldsymbol{R}_i > 0$ and $\boldsymbol{R}_{ij} \geqslant 0$. In order to suppress tracking error and realize consensus, each MU discovers the optimal control strategy by minimizing its own cost function (**??**). To be distributed, the control strategy $\hat{\boldsymbol{u}}_i$ of MU i only utilizes its individual state information and that of its neighbors during the gaming process.

Definition 1. (Pareto Optimal Solution)[**?**]. In a cooperative game composed by n players, for any two sets of control strategies $\boldsymbol{u}^* = \left[\boldsymbol{u}_1^*, \cdots, \boldsymbol{u}_n^*\right]^{\mathrm{T}}$ and $\boldsymbol{u} = \left[\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n\right]^{\mathrm{T}}$, $\boldsymbol{u}^*$ is said to dominate $\boldsymbol{u}$ in Pareto sense if the following conditions hold:

$$\begin{cases} J_i(\boldsymbol{u}^*) \leqslant J_i(\boldsymbol{u}) & \text{for all } i \in \{1,2,\cdots n\} \\ J_i(\boldsymbol{u}^*) < J_i(\boldsymbol{u}) & \text{for at least one } i \in \{1,2,\cdots n\} \end{cases} \tag{17}$$

$\boldsymbol{u}^*$ is the Pareto optimal solution if there is no other strategy dominates $\boldsymbol{u}^*$. The Pareto optimal solution can be interpreted as a public agreement of all players, where it is impossible to improve one particular player's cost function without causing loss in any other's. It can be obtained by minimizing the following combination of all player's cost functions:

$$J = \sum_{i=1}^{n} \lambda_i J_i \tag{18}$$

where $\lambda_i \in (0,1)$ is the weight on cost function of player i, and $\sum_{i=1}^{n} \lambda_i = 1$.

For the designed game consisting of (**??**) and (**??**), in order to obtain its Pareto optimal solution $\hat{\boldsymbol{u}}^*$, a model predict strategy is proposed. Firstly, game model (**??**) must be discretised and iterated. At t_k, MU's dynamic is discretised as:

$$\boldsymbol{x}_{i,k+1} = \boldsymbol{A}_k \boldsymbol{x}_{i,k} + \boldsymbol{B}_k \hat{\boldsymbol{u}}_{i,k} \tag{19}$$

where $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are discrete dynamic matrices. Then discretised cooperative consensu error yields the following expression:

$$\begin{aligned} \boldsymbol{e}_{i,k+1} &= \boldsymbol{L}_i(\boldsymbol{A}_k \boldsymbol{x}_{i,k} + \boldsymbol{B}_k \hat{\boldsymbol{u}}_{i,k} - \boldsymbol{x}_{i,k+1}^d) \\ &\quad + \sum_{j\in N_i} \boldsymbol{L}_{ij}(\boldsymbol{A}_k \boldsymbol{x}_{j,k} + \boldsymbol{B}_k \hat{\boldsymbol{u}}_{j,k} - \boldsymbol{x}_{j,k+1}^d) \\ &= \hat{\boldsymbol{A}}_{i,k}\boldsymbol{x}_{i,k} + \hat{\boldsymbol{B}}_{i,k}\hat{\boldsymbol{u}}_{i,k} - \hat{\boldsymbol{x}}_{i,k+1}^d \\ &\quad + \sum_{j\in N_i} (\hat{\boldsymbol{A}}_{ij,k}\boldsymbol{x}_{j,k} + \hat{\boldsymbol{B}}_{ij,k}\hat{\boldsymbol{u}}_{j,k} - \hat{\boldsymbol{x}}_{ij,k+1}^d) \end{aligned} \tag{20}$$

where $\hat{\boldsymbol{A}}_{i,k} = \boldsymbol{L}_i \boldsymbol{A}_k$, $\hat{\boldsymbol{A}}_{ij,k} = \boldsymbol{L}_{ij}\boldsymbol{A}_k$, and similarly to other matrices. Further, define predict horizon of the solver as N_p and iterate the discrete dynamics (**??**) in the predict horizon $\left[t_k, t_{k+N_p-1}\right]$. Define $\boldsymbol{E}_{i,k} = \left[\boldsymbol{e}_{i,k+1}^{\mathrm{T}}, \boldsymbol{e}_{i,k+2}^{\mathrm{T}}, \cdots, \boldsymbol{e}_{k+N_p}^{\mathrm{T}}\right]^{\mathrm{T}}$ $\boldsymbol{U}_{i,k} = \left[\hat{\boldsymbol{u}}_{i,k}^{\mathrm{T}}, \hat{\boldsymbol{u}}_{i,k+1}^{\mathrm{T}}, \cdots, \hat{\boldsymbol{u}}_{k+N_p-1}^{\mathrm{T}}\right]^{\mathrm{T}}$ as sequences of error and control strategy, one has:

$$\begin{aligned} \boldsymbol{E}_{i,k} &= \boldsymbol{H}_{i,k}\boldsymbol{x}_{i,k} + \boldsymbol{W}_{i,k}\boldsymbol{U}_{i,k} - \boldsymbol{X}_{i,k}^d \\ &\quad + \sum_{j\in N_i} \boldsymbol{H}_{j,k}\boldsymbol{x}_{j,k} + \boldsymbol{W}_{j,k}\boldsymbol{U}_{j,k} - \boldsymbol{X}_{j,k}^d \\ &= \boldsymbol{H}_{i,k}\boldsymbol{x}_{i,k} + \boldsymbol{W}_{i,k}\boldsymbol{U}_{i,k} + \boldsymbol{S}_{i,k} \end{aligned} \tag{21}$$

where

$$\begin{aligned} \boldsymbol{H}_{i,k} &= \begin{bmatrix} \hat{\boldsymbol{A}}_{i,k}^{\mathrm{T}} & \cdots & \left(\hat{\boldsymbol{A}}_{i,k}^{N_p}\right)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{W}_{i,k} &= \begin{bmatrix} \hat{\boldsymbol{B}}_{i,k} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \hat{\boldsymbol{A}}_{i,k}^{N_p-1}\hat{\boldsymbol{B}}_{i,k} & \cdots & \hat{\boldsymbol{B}}_{i,k} \end{bmatrix} \\ \boldsymbol{X}_{i,k}^d &= \begin{bmatrix} \hat{\boldsymbol{x}}_{i,k}^{\mathrm{T}} & \cdots & \hat{\boldsymbol{x}}_{i,k+N_p}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{S}_{i,k} &= -\boldsymbol{X}_{i,k}^d + \sum_{j\in N_i} (\boldsymbol{H}_{j,k}\boldsymbol{x}_{j,k} + \boldsymbol{W}_{j,k}\boldsymbol{U}_{j,k} - \boldsymbol{X}_{j,k}^d) \end{aligned} \tag{22}$$

Secondly, do the same for the cost function, the discrete form and iterative form are:

$$\begin{aligned} J_i &= \sum_{g=k}^{k+N_p-1} \left(\boldsymbol{e}_{i,g+1}^{\mathrm{T}} \boldsymbol{Q}_i \boldsymbol{e}_{i,g+1} + \hat{\boldsymbol{u}}_{i,g}^{\mathrm{T}} \boldsymbol{R}_i \hat{\boldsymbol{u}}_{i,g} + \sum_{j\in N_i} \hat{\boldsymbol{u}}_{j,g}^{\mathrm{T}} \boldsymbol{R}_{ij} \hat{\boldsymbol{u}}_{j,g} \right) \\ J_{i,k} &= \boldsymbol{E}_{i,k}^{\mathrm{T}} \boldsymbol{Q}_{i,k} \boldsymbol{E}_{i,k} + \boldsymbol{U}_{i,k}^{\mathrm{T}} \boldsymbol{R}_{i,k} \boldsymbol{U}_{i,k} + \sum_{j\in N_i} \boldsymbol{U}_{j,k}^{\mathrm{T}} \boldsymbol{R}_{ij,k} \boldsymbol{U}_{j,k} \end{aligned} \tag{23}$$

where $\boldsymbol{Q}_{i,k} = \boldsymbol{I}_{N_p} \otimes \boldsymbol{Q}_i$, $\boldsymbol{R}_{i,k} = \boldsymbol{I}_{N_p} \otimes \boldsymbol{R}_i$ and $\boldsymbol{R}_{ij,k} = \boldsymbol{I}_{N_p} \otimes \boldsymbol{R}_{ij}$. However, the optimal control strategy of single MU cannot be solved independently due to the coupling in (**??**) and (**??**). In order to achieve distributed control, further decoupling of cost function is required. Substituting (**??**) into (**??**), one has:

$$\begin{aligned} J_{i,k} &= \left(\boldsymbol{x}_{i,k}^{\mathrm{T}}\boldsymbol{H}_{i,k}^{\mathrm{T}} + \boldsymbol{S}_{i,k}^{\mathrm{T}}\right) \boldsymbol{Q}_{i,k} \left(\boldsymbol{H}_{i,k}\boldsymbol{x}_{i,k} + \boldsymbol{S}_{i,k}\right) \\ &\quad + \boldsymbol{U}_{i,k}^{\mathrm{T}} \left(\boldsymbol{R}_{i,k} + \boldsymbol{W}_{i,k}^{\mathrm{T}}\boldsymbol{Q}_{i,k}\boldsymbol{W}_{i,k}\right) \boldsymbol{U}_{i,k} \\ &\quad + 2\left(\boldsymbol{x}_{i,k}^{\mathrm{T}}\boldsymbol{H}_{i,k}^{\mathrm{T}} + \boldsymbol{S}_{i,k}^{\mathrm{T}}\right) \boldsymbol{Q}_{i,k}\boldsymbol{W}_{i,k}\boldsymbol{U}_{i,k} \\ &\quad + \sum_{j\in N_i} \boldsymbol{U}_{j,k}^{\mathrm{T}} \boldsymbol{R}_{ij,k} \boldsymbol{U}_{j,k} \end{aligned} \tag{24}$$

MU i can only optimize its individual control strategy, so terms in (**??**) without $\boldsymbol{U}_{i,k}$ are omitted. Therefore, the cooperative game controller can be reformulated as the following quadratic programming (QP) problem:

$$\min_{\boldsymbol{U}_{i,k}} J_{i,k} = \frac{1}{2}\boldsymbol{U}_{i,k}^{\mathrm{T}} \boldsymbol{q}_{i,k} \boldsymbol{U}_{i,k} + \boldsymbol{p}_{i,k}^{\mathrm{T}} \boldsymbol{U}_{i,k} \tag{25}$$

where

$$\begin{aligned} \boldsymbol{q}_{i,k} &= 2\left(\boldsymbol{R}_{i,k} + \boldsymbol{W}_{i,k}^{T}\boldsymbol{Q}_{i,k}\boldsymbol{W}_{i,k}\right) \\ \boldsymbol{p}_{i,k} &= 2\boldsymbol{W}_{i,k}^{T}\boldsymbol{Q}_{i,k}\left(\boldsymbol{H}_{i,k}\boldsymbol{x}_{i,k} + \boldsymbol{S}_{i,k}\right) \end{aligned} \tag{26}$$

During the gaming process, MU i solve the above QP problem in each time t_k to achieve Pareto optimal, based on neighbor MU's control strategy sequences $\boldsymbol{U}_j$ and predicted state sequences $\boldsymbol{X}_j$. After obtaining its own control strategy sequence $\boldsymbol{U}_i$, the first term $\hat{\boldsymbol{u}}_i^*$ is extracted, and the actual control input is calculated as $\boldsymbol{u}_i^* = \hat{\boldsymbol{u}}_i^* - \hat{\boldsymbol{T}}_i$.

Remark 3. Game model (**??**) includes the predicted tension force $\hat{\boldsymbol{T}}_i$. By introducing it into dynamics, the effect of tension force on the game state $\boldsymbol{e}_i$ can be forecasted. Thus this effect can be reduced by active control. Unlike traditional perturbation estimation methods, this approach does not require bounded assumptions about the perturbations themselves and their derivatives, which are generally unknown and highly conservative.

4. Simulation and results

In this section, numerical simulations are conducted to demonstrate the performance of the proposed cooperative game theory based formation controller. The simulation environment for TSNR is built based on MuJoCo [**?**], which is a high-fidelity open source physics engine considering complex dynamical effects and flexible constraints.

The orbit radius of space debris and TSNR is r_0 = 42164 km. The initial position of target is at $\boldsymbol{x}_0 = [5, 0, 0]^{\mathrm{T}}$, and the initial states of four MUs when leaving the platform are $\boldsymbol{x}_1(t_0) = [-10, 0.75, 0.75, 0, 0, 0]^{\mathrm{T}}$, $\boldsymbol{x}_2(t_0) = [-10, -0.75, 0.75, 0, 0, 0]^{\mathrm{T}}$, $\boldsymbol{x}_3(t_0) = [-10, 0.75, -0.75, 0, 0, 0]^{\mathrm{T}}$ and $\boldsymbol{x}_4(t_0) = [-10, -0.75, -0.75, 0, 0, 0]^{\mathrm{T}}$, where the position and velocity units are m and m/s. Before capture, TSNR is stored in the platform and folded in a square pattern with mesh edge l_m = 0.5 m. Then TSNR is released and the net is gradually expanded as large as possible before contact by the motion of four MUs. To successfully capture the target, the net should close completely after contact and envelope it, so the terminal condition of four MUs are $\boldsymbol{x}_1(t_f) = [5, 0.5, 0.5, 0, 0, 0]^{\mathrm{T}}$, $\boldsymbol{x}_2(t_f) = [5, -0.5, 0.5, 0, 0, 0]^{\mathrm{T}}$, $\boldsymbol{x}_3(t_f) = [5, 0.5, -0.5, 0, 0, 0]^{\mathrm{T}}$ and $\boldsymbol{x}_4(t_f) = [5, -0.5, -0.5, 0, 0, 0]^{\mathrm{T}}$. The other detailed parameters of TSNR is shown in [**?**].

Fig.**??** depicts tension forces acting on each MU obtained by the proposed neural network estimator. It is clear that during approaching phase (0 − 10 s), tension force acting on each MU is quite small or even negligible. This phenomenon is due to the net not being taut enough to generate significant force. When the net is fully deployed (10 − 20 s), tension force increases substantially, enabling each MU to interact with others through this force transmission. After the net makes contact with debris (at 20 s), a collison force is generated. However, the impact of the collision phase is not discussed in this paper for simplicity.

Fig.**??** shows the trajectory tracking errors of four MUs. It can be seen that the tracking error in each direction remains limited to 0.02 m throughout the mission phase. From 0 s to 10 s, tension force is relatively small during releasing phase (as shown in Fig.**??**), and the tracking error steadily decreases as the net expands until unfolding to its maximum size. Subsequently during approaching phase, there is a slight increase in the error occurs due to the rise in tension force. The error further widens during the enclosing phase, attributed to the contact between the net and debris (at 20 s). However, the designed cooperative controller effectively suppresses tracking errors, maintaining them within a narrow range.

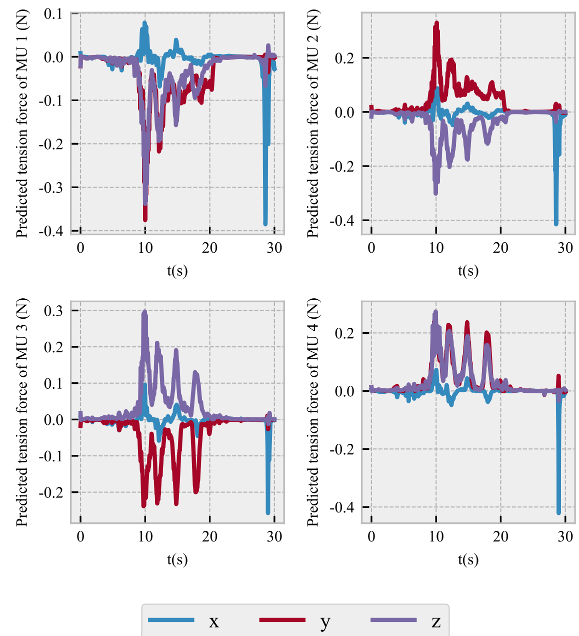

Fig. 4: Predicted tension forces of four MUs.

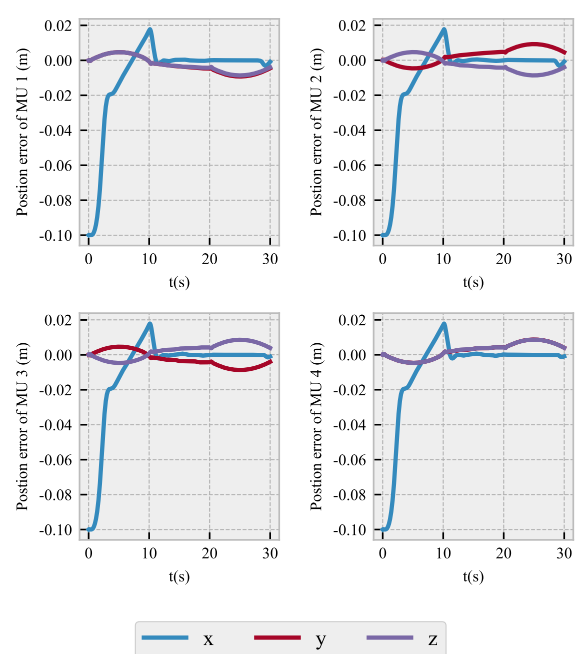

Fig. 5: Position tracking errors of four MUs.

Envelope capture sequence of TSNR is illustrated in Fig.**??**, depicting the action of the designed cooperative formation controller. Upon releasing from platform satellite, four MUs progressively move away to expand the net. As the net reaches its maximum size, MUs drive it closer to space

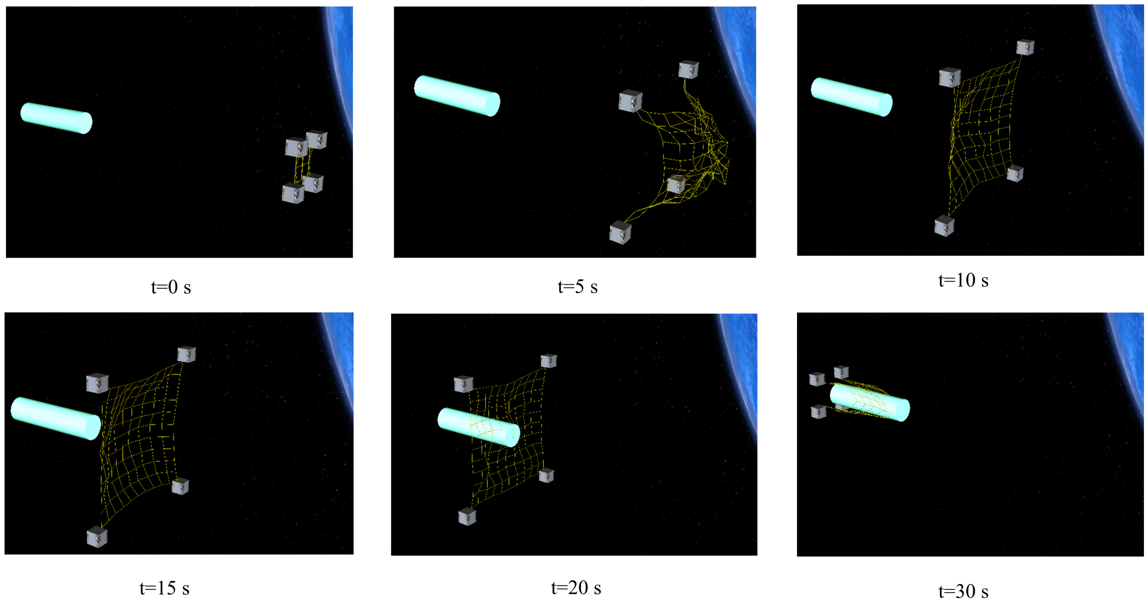

Fig. 6: Envelope capture sequence of TSNR.

debris (depicted as the green cylinder in Fig.**??**). It's evident that in the final moments, the net can completely encircle the target, thereby successfully completing the debris capture mission.

5. Conclusion

This paper proposes a novel strategy to tackle the challenges associated with capturing space debris and tracking a specific trajectory for TSNR. The proposed strategy combines principles from both differential game and machine learning theory, aimed at efficiently completing the capture mission. The conclusions are outlined as follows:

1. A DNN based force estimator is developed to depict the intricate time-delayed interactions among MUs, facilitating the precise estimation and prediction of tension force. And a predictive dynamic model is introduced according to it.
2. Using cooperative game theory, a formation tracking control strategy is proposed to deeply reflect the relationship among MUs. Each MU optimise its cost function in a fully distributed manner to suppress the tracking error and achieve consensus.
3. Numerical simulations demonstrate the effectiveness of the proposed strategy in the mission of capturing space debris utilizing TSNR.

Declaration of competing interest

The authors state that they have no known competing financial or personal interests that could influence the work reported in this publication.

Acknowledgements

This research was funded by the National Natural Science Foundation of China under Grant Nos.62222313, 62173275, 62327809, 62303381, and 62303312.

References

[] Asri, E.G., Zhu, Z.H., 2024. An introductory review of swarm technology for spacecraft on-orbit servicing. Int Journal of Mech Sys Dyn 4, 3–21. URL: `https://onlinelibrary.wiley.com/doi/10.1002/msd2.12098`, doi:`10.1002/msd2.12098`.

[] Benvenuto, R., Salvi, S., Lavagna, M., 2015. Dynamics analysis and GNC design of flexible systems for space debris active removal. Acta Astronaut 110, 247–265. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0094576515000296`, doi:`10.1016/j.actaastro.2015.01.014`.

[] Zhang, Y., Huang, P., Meng, Z., Liu, Z., 2019. Precise angles-only navigation for noncooperative proximity operation with application to tethered space robot. IEEE Trans. Contr. Syst. Technol. 27, 1139–1150. URL: `https://ieeexplore.ieee.org/document/8267496/`, doi:`10.1109/TCST.2018.2790400`.

[] Zhang, F., Huang, P., 2017. Releasing dynamics and stability control of maneuverable tethered space net. IEEE/ASME Trans. Mechatron. 22, 983–993. URL: `http://ieeexplore.ieee.org/document/7742375/`, doi:`10.1109/TMECH.2016.2628052`.

[] Wen, H., Zhu, Z.H., Jin, D., Hu, H., 2016. Constrained tension control of a tethered space-tug system with only length measurement. Acta Astronautica 119, 110–117. URL: `https://www.sciencedirect.com/science/article/pii/S0094576515004075`, doi:`10.1016/j.actaastro.2015.11.011`.

[] Liu, Y., Zhang, F., Huang, P., 2020. Coordinated control for constrained multiple spacecraft system. IEEE Trans. Aerosp. Electron. Syst. 56, 1189–1201. URL: `https://ieeexplore.ieee.org/document/8760549/`, doi:`10.1109/TAES.2019.2926612`.

[] Zhang, F., Huang, P., 2021. Fuzzy-based adaptive super-twisting sliding-mode control for a maneuverable tethered space net robot. IEEE Trans. Fuzzy Syst. 29, 1739–1749. URL: `https://ieeexplore.ieee.org/document/9057415/`, doi:`10.1109/TFUZZ.2020.2985325`.

[] Liu, Y., Ma, Z., Zhang, F., Huang, P., 2023. Time-Varying Formation Planning and Scaling Control for Tethered Space Net Robot. IEEE Transactions on Aerospace and Electronic Systems , 1–12doi:10.1109/TAES.2023.3276742.

[] Jiang, L., Gonzalez, F., McFadyen, A., 2020. Cooperative Game Theory based Multi-UAV Consensus-based Formation Control, in: 2020 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, Athens, Greece. pp. 93–99. URL: https://ieeexplore.ieee.org/document/9213841/, doi:10.1109/ICUAS48674.2020.9213841.

[] Li, J., Chen, S., Li, C., Wang, F., 2021. Distributed Game Strategy for Formation Flying of Multiple Spacecraft With Disturbance Rejection. IEEE Trans. Aerosp. Electron. Syst. 57, 119–128. URL: https://ieeexplore.ieee.org/document/9146203/, doi:10.1109/TAES.2020.3010593.

[] Han, N., Luo, J., Zong, L., 2022. Cooperative Game Method for On-Orbit Substructure Transportation Using Modular Robots. IEEE Trans. Aerosp. Electron. Syst. 58, 1161–1175. URL: https://ieeexplore.ieee.org/document/9536397/, doi:10.1109/TAES.2021.3111141.

[] Liu, Y., Ma, Z., Zhang, F., Huang, P., Lu, Y., Chang, H., 2024. Coordinated transportation of tethered multi-rotor UAVs based on differential graphical games. Aerospace Science and Technology 148, 109078. URL: https://linkinghub.elsevier.com/retrieve/pii/S1270963824002116, doi:10.1016/j.ast.2024.109078.

[] Liu, Y., Zhang, F., Huang, P., Lu, Y., 2021. Fixed-time consensus tracking for second-order multiagent systems under disturbance. IEEE Trans. Syst. Man Cybern, Syst. 51, 4883–4894. URL: https://ieeexplore.ieee.org/document/8870232/, doi:10.1109/TSMC.2019.2944392.

[] Ning, B., Jin, J., Zheng, J., Man, Z., 2018. Finite-time and fixed-time leader-following consensus for multi-agent systems with discontinuous inherent dynamics. Int J Control 91, 1259–1270. URL: https://www.tandfonline.com/doi/full/10.1080/00207179.2017.1313453, doi:10.1080/00207179.2017.1313453.

[] Li, S., Liu, C., Sun, Z., 2021. Finite-time distributed hierarchical control for satellite cluster with collision avoidance. Aerosp Sci Technol 114, 106750. URL: https://linkinghub.elsevier.com/retrieve/pii/S1270963821002601, doi:10.1016/j.ast.2021.106750.

[] Hua, Y., Dong, X., Han, L., Li, Q., Ren, Z., 2019. Finite-time time-varying formation tracking for high-order multiagent systems with mismatched disturbances. IEEE Trans. Syst. Man Cybern, Syst. , 1–9URL: https://ieeexplore.ieee.org/document/8466101/, doi:10.1109/TSMC.2018.2867548.

[] Shi, G., Shi, X., O'Connell, M., Yu, R., Azizzadenesheli, K., Anandkumar, A., Yue, Y., Chung, S.J., 2019. Neural Lander: Stable Drone Landing Control Using Learned Dynamics, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, Montreal, QC, Canada. pp. 9784–9790. URL: https://ieeexplore.ieee.org/document/8794351/, doi:10.1109/ICRA.2019.8794351.

[] Torrente, G., Kaufmann, E., Fohn, P., Scaramuzza, D., 2021. Data-Driven MPC for Quadrotors. IEEE Robot. Autom. Lett. 6, 3769–3776. URL: https://ieeexplore.ieee.org/document/9361343/, doi:10.1109/LRA.2021.3061307.

[] O'Connell, M., Shi, G., Shi, X., Azizzadenesheli, K., Anandkumar, A., Yue, Y., Chung, S.J., 2022. Neural-Fly enables rapid learning for agile flight in strong winds. Sci. Robot. 7, eabm6597. URL: https://www.science.org/doi/10.1126/scirobotics.abm6597, doi:10.1126/scirobotics.abm6597.

[] Farina, M., Amato, P., 2002. On the optimal solution definition for many-criteria optimization problems, in: 2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings. NAFIPS-FLINT 2002 (Cat. No. 02TH8622), IEEE, New Orleans, LA, USA. pp. 233–238. URL: http://ieeexplore.ieee.org/document/1018061/, doi:10.1109/NAFIPS.2002.1018061.

[] Todorov, E., Erez, T., Tassa, Y., 2012. MuJoCo: A physics engine for model-based control, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Vilamoura-Algarve, Portugal. pp. 5026–5033. URL: http://ieeexplore.ieee.org/document/6386109/, doi:10.1109/IROS.2012.6386109.