Aerospace Science and Technology 124 (2022) 107527

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Bi-level game-based reconfigurable control for on-orbit assembly

Yuan Chai, Jianjun Luo *, Mingming Wang

Northwestern Polytechnical University, Xi'an, Shaanxi, China

ARTICLE INFO

Article history:
Received 7 December 2021
Received in revised form 15 February 2022
Accepted 27 March 2022
Available online 31 March 2022
Communicated by Qinglei Hu

Keywords:
Reconfigurable control
On-orbit assembly
Game theory

ABSTRACT

During on-orbit assembly, microsatellites are used to provide control ability for substructures to be assembled. Every time a substructure is assembled to the combination composed of the assembled substructures, the significant variations of the combination are not only in the mass properties, but also in the microsatellites configuration. Accordingly, this paper presents a bi-level game-based reconfigurable control method for microsatellites. At the upper level, taking each substructure with its microsatellites as a subsystem, the interaction among different subsystems is formulated as a graphical game with constraints from system and lower level, where the total control force of each subsystem corresponding to Nash equilibrium is computed in a distributed iteration manner. At the lower level, the interaction among microsatellites of one subsystem is formulated as a cooperative game with constraints from system and upper level, where the searching for Pareto optimal is nested into the iteration of the upper game. The proposed method can benefit both assembly of each substructure for the good universality and control of each microsatellite for the reduced individual burden, which is validated by numerical simulations.

1. Introduction

With the rapid development of space exploration, there is a growing demand in size and complexity of space infrastructures [1]. Limited by the dimension of launch vehicles and the complexity of deployable mechanisms, on-orbit assembly has been an appealing technique to construct such space infrastructures in orbit by autonomous agents. Compared with 'pick and place' form by space robots which needs to launch all of the elements simultaneously, the 'free-flying' form by servicing microsatellites has advantage in flexibility [2]. Servicing microsatellites can be attached to substructures without independent actuators to realize the attitude and orbit control. Notably, every time a substructure is assembled into the original combination composed of all assembled substructures to form a new combination, the significant changes of mass properties and microsatellites configuration will lead to unexpected state deviation. Therefore, different from our previous research which focused on the transportation stage of substructures before assembly operations [3,4], it is desired to design a reconfigurable control accounting for these properties changes for the post-assembly stage of each substructure in this paper.

Reconfigurable control has been widely researched in different scenarios. Reconfiguration in the active fault-tolerant control places emphasis on the actuator fault diagnosis and control recovery to adapt these failures, which does not involve the mass property changes [5]. In the post-capture attitude takeover control scenarios, the reconfigurable control focuses more narrowly on the parameter identification for the new system and control reallocation [6]. However, in our scenario, there is not only the mutation of mass properties but also the additional actuators. Reference [7] studied the reconfigurable control allocation during autonomous assembly. Reference [8] presented an LQR-based cooperative control to balance the fuel usage between modules. Literature [9] proposed a resource aggregated reconfigurable control for the combination consisting of two space modules. Not like these methods of which the involved modules are equipped with independent actuators, the substructure is controlled by the attached microsatellites. Nevertheless, the reconfigurable control of this paper is to coordinate the actuators of servicing microsatellites and realize the stability of the new combination.

To realize the coordination among servicing microsatellites, an intuitive thought is to set up an unified dynamic model for the new combination under the unified coordinate frame, based on which the existing controller can be directly applied to realize the describe the dynamic interaction. By viewing a microsatellite on the new combination as the central processor, some centralized methods which are developed for the control problem of a traditional spacecraft with multiple actuators can be used to conduct control calculation and allocation of all microsatellites [10,11]. As a constrained centralized optimization problem, the scale and number of decision variables of control allocation increase as the number of microsatellites increases. Moreover, the heavy depen-

* Corresponding author.
E-mail address: jjluo@nwpu.edu.cn (J. Luo).

https://doi.org/10.1016/j.ast.2022.107527
1270-9638/

dence on the central node makes the system less fault tolerance. Considering that the servicing microsatellites are equipped with independent processor, distributed control allocation methods can be used to reduce the burden of central node [12]. Without extra constraints, reference [13,14] developed distributed control allocation algorithms to allocate the total required control torques. To further consider the input saturation constrains, reference [15] proposed the distributed constrained control allocation method. However, in this paper, besides the control amplitude constraint, the coupled constraints such as torque constraint to avoid disturbing attitude control during orbit control are also required to be considered. The introduction of these constraints will increase the computational complexity of distributed control allocation. Moreover, the dependence on central node still exists in the information collect and the total required control computation, which lead to low-tolerance.

Compared with the above traditional methods, game theory provides a more intelligent tool to make the best of the autonomy of microsatellite and realize the coordination among microsatellites. It is noted that the key of game theory is to find the equilibrium, and the constraints would heavily increase the complexity in solving equilibrium. For the uncooperative game-based method, the traditional way to get the Nash equilibrium strategy is to solve a set of complicated Hamilton-Jacobi equations, which cannot deal with constraints [16]. In order to deal with the input constraints, reference [17,18] employed a non-quadratic function to construct the individual performance functions. However, the input constraint in this paper is not a symmetric constraint because one surface of the microsatellite is connected to the structure. In order to satisfy the state constraints, literature [19] proposed a new barrier Lyapunov function and combined it into zero-sum game to prevent constraint violations. Different from [19] resorting to the techniques in control theory, reference [20] formulated the state constraints from the formation tracking into the individual objective function of each agent to satisfy the formation objective. However, these methods rely on the neural networks to get the Nash equilibrium, whose learning precision heavily depends on the quality and quantity of data. Literature [21] employed the frame of model predictive control (MPC) to deal with constrained pursuit-evasion game, which seems to be a better scheme for constrained uncooperative game for the great performance of MPC in dealing with constraints. For the cooperative game, there are two kinds of research ideas. One is, under the team objective function which is the sum of individual performance function, the methods in uncooperative game are extended to the cooperative game. As in [22,23], the cooperative game was solved by adaptive dynamic programming to get Pareto solution, where the input constraints were reflected in the non-quadratic performance index function. Another is to find a more optimal solution named Pareto optimal solution, which can achieve simultaneous optimization of all players' performance indices. As a widely-used strategy candidate in cooperative games, it is usually got by solving a multiple-objective problem [24]. As analyzed in [3], the introduction of constraints would increase the optimization complexity for this multiple-objective problem.

For cooperative game or uncooperative game, besides the computational complexity increases by the complex constraints, the unified dynamic model used by each microsatellite has to be updated every time the substructures are assembled based on the sudden changed communication topology among microsatellites [25]. Furthermore, it is usually required that the substructures have autonomy to be assembled to form one pattern for one objective, and can be disassembled and reassembled to form another pattern. In other words, it is desired to capture the interaction among subsystems and inside subsystem, respectively. Taking advantage of the coordination among microsatellites in one substructure in transporting stage, a hierarchical structure is employed for reconfigurable control, of which the upper level aims to realize the coordination among different subsystems and the lower level retains the coordination inside one subsystem. It is noted that one subsystem refers to one substructure with attached microsatellites. Compared with the above single-level game-based methods which deals with all constraints directly among all microsatellites, the multiple constraints can be solved respectively in different levels among limited number of subsystems or limited number of microsatellites inside one subsystem, which can reduce the computational complexity. Compared with single level based on unified dynamic model of the new combination, the bi-level game control based on interconnected system can improve the scalability to more complex system.

In light of the foregoing, the reconfigurable control in this scenario with significant changes in mass properties and microsatellites configuration is studied under the hierarchical structure, so as to fully capture the interaction among subsystems and inside subsystem. By resorting to MPC [21] and multiple-objective optimization [24], a bi-level game-based control method for servicing microsatellites is proposed to deal with different kinds of constraints. The main contributions of the paper are as follows:

(1) Different from the existing reconfigurable controllers, the proposed method makes the best of the coordination among servicing microsatellites by means of game theory. To further hold the autonomy of each substructure, a bi-level game structure is proposed to capture the difference in interaction among subsystems and inside subsystem.

(2) In the upper level, the interaction among different subsystems is formulated as a graphical game to control the whole system described by an interconnected system, which can improve scalability. At the lower level, the cooperation is taken as granted, which leads to a cooperative game formulation. By transforming the constraints in different levels, the upper level and the lower level are integrated by constraints.

(3) In order to get the equilibrium of the bi-level game, an iteration algorithm is proposed to optimize the performance index function of both upper level and lower level while satisfying different kinds of constraints. Both the periodic control form and simplified quadratic programming (QP) formulation can improve online implementation ability.

The remaining part of this paper is as follows: Section 2 describes the reconfigurable control problem for on-orbit assembly and formulates it as a bi-level game. The detailed game-based controller design is shown in Section 3. In Sec. 4, numerical simulations are presented to demonstrate the effectiveness of the proposed method. Finally, Sec. 5 concludes this paper.

2. Preliminaries

2.1. Problem statement

During on-orbit assembly, servicing microsatellites are fixed to a substructure without independent actuators so as to realize the control, as shown in Fig. 1. A large variation in mass properties is caused every time a substructure is assembled to form a new combination and physically linked to the original combination by connection interface. It is noted that two types of microsatellites are used to respectively produce torque and force. For the first type, microsatellite is equipped with reaction wheels along the inertia principal axis to produce torque. And the second type is equipped with thruster nozzles on each surface except the one connected with the structure to produce force. In this paper, we mainly focus on designing a reconfigurable orbit control by the second type of microsatellites, and the similar design process for

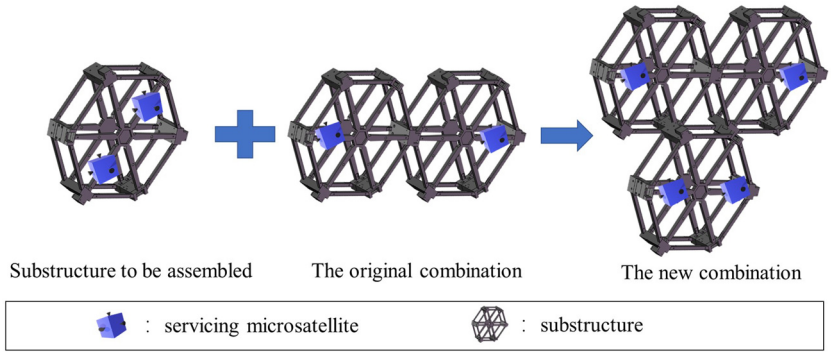


Fig. 1. Sketch of the mission.

attitude control by the first type of microsatellites. Thus, the servicing microsatellite mentioned below refers to the second type of microsatellites. For generality, the number of the substructure to be assembled concurrently is not limited to only one.

It is noted that the structure with microsatellites is termed as subsystem $i \in \mathcal{M}$ and the j-th microsatellite in the subsystem i is termed as $A_j^i \in \mathcal{N}_i$, where $\mathcal{M}$ is a set of subsystems' index in the new combination and $\mathcal{N}_i$ is a set of microsatellites' index in the subsystem i. For the reconfigurable control problem, the following frames are defined:

(1) The Clohessy-Wiltshire-Hill (CWH) frame $O_c x_c y_c z_c$, with the origin at the position of the combination before assembling. x_c axis points radially outward from Earth, y_c axis is in the direction of the orbital velocity direction of the combination, and z_c axis is along the orbit normal completing the triad.

(2) The body frame of each subsystem $O_{bi}x_{bi}y_{bi}z_{bi}, i \in \mathcal{M}$, with the origin at the CoM of each subsystem. The three axes of the frame are defined to be parallel with the inertia principal axes of microsatellites A_1^i.

(3) The body frame of each microsatellite $O_{ij}x_{ij}y_{ij}z_{ij}, i \in \mathcal{M}, j \in \mathcal{N}_i$. Its axes are aligned with the three inertia principal axes of microsatellite A_j^i. Assumed that the surface of A_j^i connected with the structure is vertical to $-z_{ij}$, which means that A_j^i can only generate force along the $\pm x_{ij}$, $\pm y_{ij}$ and $-z_{ij}$ directions.

2.2. Relative orbit dynamic

Each time the substructures are assembled, the state evolution of the combination is driven by microsatellites from both the substructures of the original combination and the new assembled substructures. The robustness of the controller cannot suppress the interference caused by the significant change of the physical properties and the coupled control inputs from the additional subsystems.

To this end, the mass properties are aggregated to form the mass properties of the new combination. Among these mass properties, the mass and the CoM are related to orbit control, of which the aggregation formulas are represented as follows, respectively [9]

$$\begin{aligned} m_{agg} &= m_i + \sum_{k\in\mathcal{M}_{\hat{i}}} m_k \\ {}^i\boldsymbol{r}_g^{ij} &= \frac{1}{m_{agg}} m_i {}^i\boldsymbol{r}_i^{ij} + \frac{1}{m_{agg}} \sum_{k\in\mathcal{M}_{\hat{i}}} m_k {}^i\boldsymbol{r}_k^{ij} \end{aligned} \tag{1}$$

where m_k is the mass of one subsystem and m_{agg} is the mass of the new combination, $\mathcal{M}_{\hat{i}}$ denotes all subsystems except subsystem i. ${}^i\boldsymbol{r}_g^{ij}$ is a vector from the CoM of the combination in the frame of $O_{bi}x_{bi}y_{bi}z_{bi}$ to the CoM of microsatellite A_j^i. ${}^i\boldsymbol{r}_k^{ij}$ is a vector from the CoM of the subsystem k in the frame of $O_{bi}x_{bi}y_{bi}z_{bi}$ to the CoM of microsatellite A_j^i.

Based on the aggregated mass properties, the aggregated relative orbit dynamic model is described from the perspective of each subsystem i, so as to maintain adequate control performance without losing the autonomy of the subsystem

$$\begin{aligned} \dot{\boldsymbol{x}}_i &= \boldsymbol{A}_i\boldsymbol{x}_i + \sum_{j\in\mathcal{N}_i} \boldsymbol{B}_{ij}\boldsymbol{u}_{ij} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{B}_k\boldsymbol{\tau}_k \\ &= \boldsymbol{A}_i\boldsymbol{x}_i + \boldsymbol{B}_i\boldsymbol{\tau}_i + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{B}_k\boldsymbol{\tau}_k, i \in \mathcal{M} \end{aligned} \tag{2}$$

where $\boldsymbol{x}_i = [\boldsymbol{r}_i^{\mathrm{T}}, \dot{\boldsymbol{r}}_i^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{6\times 1}$ is the state vector that can be measurable. $\boldsymbol{r}_i = [x, y, z]^{\mathrm{T}}$ is the position vector and $\dot{\boldsymbol{r}}_i = [\dot{x}, \dot{y}, \dot{z}]^{\mathrm{T}}$ is the velocity vector of subsystem i. And

$$\boldsymbol{A}_i = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{I}_3 \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

$$\boldsymbol{A}_{21} = \begin{bmatrix} 3n_0^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n_0^2 \end{bmatrix} \in \mathbb{R}^{3\times 3},$$

$$\boldsymbol{A}_{22} = \begin{bmatrix} 0 & 2n_0 & 0 \\ -2n_0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

where $\boldsymbol{I}_n$ denotes the identity matrix in $\mathbb{R}^{n\times n}$ and $\boldsymbol{0}_n$ is the all-zero matrix in $\mathbb{R}^{n\times n}$. $n_0 = \sqrt{\mu/r_0^3}$ is the average orbit angular velocity of the desired position, where μ is the gravitational constant of Earth and r_0 is radius of the desired position. Moreover, $\boldsymbol{u}_{ij}$ represents the control force generated by microsatellite A_j^i expressed in $O_{ij}x_{ij}y_{ij}z_{ij}$.

Accounting for the changed mass properties, the mass used in $\boldsymbol{B}_{ij}$ is changed to the mass of the new combination m_{agg}

$$\boldsymbol{B}_{ij} = \begin{bmatrix} \boldsymbol{0}_3 \\ \frac{1}{m_{agg}} \boldsymbol{C}_{bi}^c \boldsymbol{C}_{ij}^{bi} \end{bmatrix} \in \mathbb{R}^{6\times 3}$$

where $\boldsymbol{C}_{ij}^{bi}$ is the transition matrix from $O_{ij}x_{ij}y_{ij}z_{ij}$ to $O_{bi}x_{bi}y_{bi}z_{bi}$, and $\boldsymbol{C}_{bi}^c$ is the transition matrix from $O_{bi}x_{bi}y_{bi}z_{bi}$ to $O_c x_c y_c z_c$. $\boldsymbol{\tau}_i$ is the control force produced by all N_i microsatellites of subsystem i expressed in $O_c x_c y_c z_c$

$$\boldsymbol{\tau}_i = \sum_{j\in\mathcal{N}_i} \boldsymbol{C}_{bi}^c \boldsymbol{C}_{ij}^{bi} \boldsymbol{u}_{ij} \tag{3}$$

with control matrix $\boldsymbol{B}_i = [\boldsymbol{0}_3{}^{\mathrm{T}}, 1/m_{agg}\boldsymbol{I}_3^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{6\times 3}$. Accounting for the coupled inputs from other subsystems, the last term in Eq. (2) is added.

Eq. (2) is a typical model pattern of the interconnected system, which will bring convenience to the distributed controller design and introduce the flexibility of each subsystem.

2.3. Bi-level game formulation

The reconfigurable orbit control objective of each microsatellite in the same subsystem is to return to the desired state $\boldsymbol{x}_{id}$ after the assembly of substructures causes the deviation of the combination position, while using little total control effort. Therefore, in accordance with the global objective, the local performance function of each microsatellite A_j^i is defined in the quadratic manner as

$$J_{ij} = \int_{t_0}^{t_f} ((\boldsymbol{x}_i - \boldsymbol{x}_{id})^{\mathrm{T}} \boldsymbol{Q}_i (\boldsymbol{x}_i - \boldsymbol{x}_{id}) + \sum_{j\in\mathcal{N}_i} \boldsymbol{u}_{ij}^{\mathrm{T}} \boldsymbol{R}_{ij} \boldsymbol{u}_{ij} + \sum_{k\in\mathcal{M}_{\hat{i}}} \sum_{j\in\mathcal{N}_k} \boldsymbol{u}_{kj}^{\mathrm{T}} \boldsymbol{R}_{ikj} \boldsymbol{u}_{kj}) dt \tag{4}$$

where $\boldsymbol{Q}_i \in \mathbb{R}^{6\times 6}$, $\boldsymbol{R}_{ij}, \boldsymbol{R}_{ikj} \in \mathbb{R}^{3\times 3}$ are symmetric positive-definite weighting matrices. The first term in the integral function penalizes the transient error state, the second and the third term penalizes the control effort of each microsatellite from the perspective of microsatellite A_j^i. Because of the invariable relative position between different substructures, the objective of each substructure is basically consistent.

In order to achieve the information exchange among subsystems without changing the existing communication edges inside each subsystem, only the communication link between one microsatellite as the supervisor of subsystem i and another supervisor of subsystem j is added. Then, one can get the communication graph of each subsystem

$$\mathcal{G}_i = (\mathcal{N}_i, \mathcal{E}_i, \mathcal{E}_{ik}), i \in \mathcal{M}, k \in \mathcal{M}_{\hat{i}} \tag{5}$$

where $\mathcal{E}_i \in \mathcal{N}_i \times \mathcal{N}_i$ is the set of communication links inside of subsystem i. Any communication link $e_{jl}^i = (A_j^i, A_l^i) \in \mathcal{E}_i$ defines the mutual information exchange between A_j^i and A_l^i, and $e_{jl}^i \Leftrightarrow e_{lj}^i$. The connectivity properties of microsatellites inside subsystem i are guaranteed by the condition that there always exists at least one path between any pair of nodes. The communication link $e_{ab}^{ik} = (A_a^i, A_b^k) \in \mathcal{E}_{ik}$ between A_a^i and A_b^k is added to realize the information exchange between subsystem i and subsystem $k \in \mathcal{M}_{\hat{i}}$, and $e_{ab}^{ik} \Leftrightarrow e_{ba}^{ki}$. It is assumed that if there is communication demand between two nodes, the advanced communication technology can ensure the effective and reliable communications.

The constraints for orbit control include the control torque constraints and control force constraints. On the one hand, in order to avoid disturbing the attitude motion, it requires the resultant moment of all the forces to be zero

$$\sum_{j\in\mathcal{N}_i} {}^i\boldsymbol{r}_g^{ij\times} \boldsymbol{C}_{ij}^{bi} \boldsymbol{u}_{ij} = -\sum_{k\in\mathcal{M}_{\hat{i}}} {}^i\boldsymbol{L}_k \tag{6}$$

where $\boldsymbol{a}^\times$ denotes the skew symmetric matrix of vector $\boldsymbol{a}$. ${}^i\boldsymbol{L}_k$ given by ${}^i\boldsymbol{L}_k = \boldsymbol{C}_{bk}^{bi} \sum_{j\in\mathcal{N}_k} {}^k\boldsymbol{r}_g^{kj\times} \boldsymbol{C}_{kj}^{bk} \boldsymbol{u}_{kj}$ is the control torque generated by subsystem k in the frame of $O_{bi}x_{bi}y_{bi}z_{bi}$. On the other hand, for the sake of matching engineering practice and reducing the number of communications and calculations, the periodic impulsive thrust form in [26] is employed. Compared with continuous thrust form, the periodic impulsive thrust form has advantages in applications simplicity, high fuel efficiency and low computational burden. As shown in Fig. 2, one impulse period which is composed of one impulse action and the sequent free motion, and the whole control process can be regarded as a series of impulse periods. Define the sampling time as Δt, each impulse action lasts for Δt and has the maximum amplitude limitation u_{max}. Each impulse period is $T = n\Delta t$, where n is the sampling times. Taking consideration that microsatellite cannot generate force along the $+z_{ij}$ directions, the control force constraint for the w-th impulse period is formulated as follows

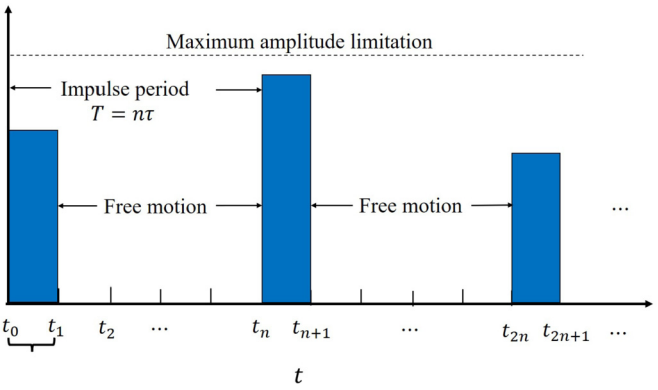


Fig. 2. Impulse diagram.

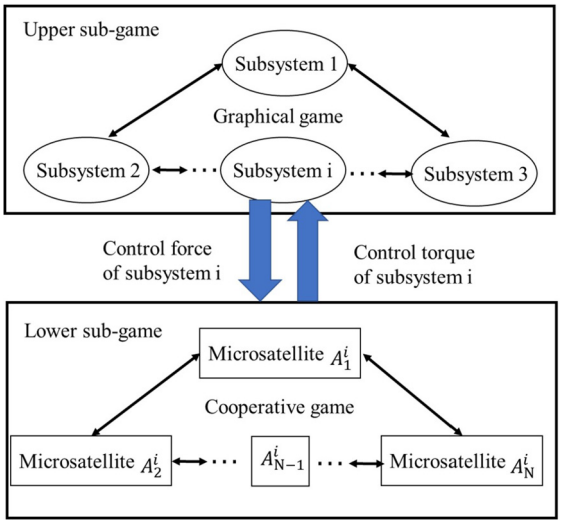


Fig. 3. Framework of bi-level game.

$$\begin{aligned} &-u_{\max}\mathbf{1}_3 \le \boldsymbol{u}_{ij}(t) \le u_{\max}\mathbf{1}_3^{\bar{3}}, \forall t \in [t_{wn}, t_{wn+1}) \\ &\boldsymbol{u}_{ij}(t) = \mathbf{0}_{3\times 1}, \forall t \in [t_{wn+1}, t_{(w+1)n}) \end{aligned} \tag{7}$$

where $\mathbf{1}_n$ represents an n-dimensional vector with all elements equal to zero, and $\mathbf{1}_n^{\bar{m}}$ denotes the vector $\mathbf{1}_n$ with the m-th element replaced by zero.

In this paper, our goal is to design a reconfigurable controller for microsatellites to achieve the state recovery when substructures are assembled to the combination. That is, considering the system described by Eq. (2) and the control constraints by Eq. (6) and Eq. (7) under the communication graph $\mathcal{G}_i$, it is desired to select control input $\boldsymbol{u}_{ij}$ of each microsatellite to reach the desired state. In order to take the interests of both subsystems and microsatellites, a bi-level game-based control method is designed to depict the interactions.

Fig. 3 shows the analytical framework of the proposed bi-level game-based control. The upper level is among subsystems. All subsystems, including the subsystems assembled in this time and the subsystems in the original combination, coordinate with each other

in a non-cooperative manner to determine control force $\boldsymbol{\tau}_i$ so that the state can be closed to the desired state using minimum total control effort. The lower level is among microsatellites of one subsystem. They coordinate with each other in a cooperative manner to decide control force $\boldsymbol{u}_{ij}$ in response to $\boldsymbol{\tau}_i$ got in the higher level so that $\boldsymbol{\tau}_i$ can be totally allocated while minimizing the energy consumption.

The interaction between upper level and the lower level is reflected in the constraints. The control force constraints of each microsatellite at the lower level will lead to the control force constraints of each subsystem at the upper level. The computed control force of each subsystem at the upper level will impose the coupled control constraints among microsatellites. It is noticed that control torque constraint involves the upper level and the lower level, which is imposed in the lower level and negotiates with other subsystems at the upper level.

3. Bi-level game-based control design

In this section, based on the formulation of the lower sub-game and upper sub-game, a bi-level game-based control algorithm is designed to compute the game equilibrium online.

3.1. Cooperative game in the lower level

3.1.1. Formulation of cooperative game among microsatellites

Given the total control force of one subsystem from the upper level, the aim of the lower level is to get the optimal control allocation scheme which can minimize the total energy consumption of microsatellites in this subsystem. Accordingly, the coordination behavior of microsatellites inside of each subsystem can be formulated as a cooperative game [27]

(1) Players: Microsatellites in the $\mathcal{N}_i$ are taken as game participants in the cooperative game. The communication graph $\mathcal{G}_i$ guarantees that there exists at least one path between any microsatellite $A_j^i \in \mathcal{N}_i$.

(2) Strategies: Each microsatellite $A_j^i \in \mathcal{N}_i$ selects its control force $\boldsymbol{u}_{ij}$ to optimize an energy-related performance index function. The strategy set of one microsatellite is within the space of the following three constraints. The first constraint is from microsatellite itself, which is formulated as Eq. (7). Secondly, the control torques constraints Eq. (6) is directly imposed to the lower level, where ${}^i\boldsymbol{L}_k$ is obtained from information exchange of the upper level. Lastly, the complex dynamics constraints of Eq. (2) is relaxed to the following linear constraints

$$\sum_{j\in\mathcal{N}_i} \boldsymbol{u}_{ij} = \boldsymbol{\tau}_i \tag{8}$$

where $\boldsymbol{\tau}_i$ is got from upper level.

(3) Payoff: In order to reduce the total energy consumption, the individual performance index function for each microsatellite is defined as

$$C_{ij} = \frac{1}{2}\boldsymbol{u}_{ij}^{\mathrm{T}}\boldsymbol{u}_{ij} \tag{9}$$

In the cooperative game, multiple microsatellites in this subsystem act collectively to minimize individual performance index function simultaneously, which is a multiple-objective problem

$$minC_i = \{C_{ij}\}_{j\in\mathcal{N}_i} \tag{10}$$

subject to the above three constraints. One typical solution to the cooperative game is the Pareto optimal solution, which is defined as follows.

Definition 1 *(Pareto optimal solution [28]).* In a cooperative game with N_i players, for any two sets of admissible strategy $\boldsymbol{u}_i = \{\boldsymbol{u}_{i1}, \boldsymbol{u}_{i2}, \cdots, \boldsymbol{u}_{iN_i}\}$ and $\boldsymbol{v}_i = \{\boldsymbol{v}_{i1}, \boldsymbol{v}_{i2}, \cdots, \boldsymbol{v}_{iN_i}\}$, $\boldsymbol{u}_i$ dominate $\boldsymbol{v}$ if and only if the following conditions hold

$$\begin{cases} C_{ij}(\boldsymbol{u}_i) \le C_{ij}(\boldsymbol{v}_i), & \text{for all } j \in \mathcal{N}_i \\ C_{il}(\boldsymbol{u}_i) < C_{il}(\boldsymbol{v}_i), & \text{for at least one } l \in \mathcal{N}_i \end{cases} \tag{11}$$

where $\boldsymbol{u}_i^*$ is a Pareto optimal solution if there is no $\boldsymbol{u}_i$ in the strategy set such that $\boldsymbol{u}_i$ dominates $\boldsymbol{u}_i^*$.

According to [29], the Pareto optimal solution has a simple characterization depending on a set of control parameters. That is if the individual performance index function is convex, Pareto optimal solution $\boldsymbol{u}_i^*$ is obtained as

$$\boldsymbol{u}_i^* \in arg\min_{\boldsymbol{u}_i} \sum_{j\in\mathcal{N}_i} \alpha_{ij} C_{ij}(\boldsymbol{u}_i) \tag{12}$$

where α_{ij} satisfies $\alpha_{ij} > 0$ and $\sum_{j\in\mathcal{N}_i}\alpha_{ij} = 1$. In accordance with the objective of balancing the fuel consumption of each microsatellite, α_{ij} is defined to satisfy that the microsatellite with less remained fuel can generate smaller control force

$$\alpha_{ij} = \frac{e^{k_\alpha/m_{ij}}}{\sum\limits_{j\in\mathcal{N}_i} e^{k_\alpha/m_{ij}}},\ j \in \mathcal{N}_i \tag{13}$$

where $k_\alpha > 0$ is an adjustable parameter and m_{ij} is the mass of the remained fuel of microsatellite A_j^i. By increasing k_α, the control force of the microsatellite with less remained fuel has to generate is further decreased. The convex combination of all players' performance index function is denoted as

$$C_i = \frac{1}{2}\sum_{j\in\mathcal{N}_i} \alpha_{ij}\boldsymbol{u}_{ij}^{\mathrm{T}}\boldsymbol{u}_{ij} \tag{14}$$

Therefore, the cooperative game among $\mathcal{N}_i$ microsatellites can be formulated as follows

$$\min_{\boldsymbol{u}_{ij}} C_i = \frac{1}{2}\sum_{j\in\mathcal{N}_i} \alpha_{ij}\boldsymbol{u}_{ij}^{\mathrm{T}}\boldsymbol{u}_{ij}$$
$$s.t.\begin{cases} -u_{\max}\boldsymbol{1}_3 \le \boldsymbol{u}_{ij}(t) \le \boldsymbol{u}_{\max}\boldsymbol{1}_3,\ \forall t \in [t_{wn}, t_{wn+1}) \\ \boldsymbol{u}_{ij}(t) = \boldsymbol{0}_{3\times 1},\ \forall t \in [t_{wn+1}, t_{(w+1)n}) \\ \sum_{j\in\mathcal{N}_i} {}^i\boldsymbol{r}_g^{ij\times}\boldsymbol{C}_{ij}^{bi}\boldsymbol{u}_{ij} = -\sum_{k\in\mathcal{M}_i} {}^i\boldsymbol{L}_k \end{cases} \tag{15}$$

After the control force $\boldsymbol{\tau}_i$ and the external control torque ${}^i\boldsymbol{L}_k$ from the upper level are transferred to the lower level, control policy of each microsatellite and the generated control torque of the subsystem can be computed by solving the above problem.

3.1.2. Derivation of QP problem

In order to approximate the optimal control allocation scheme through local interaction of microsatellites, the Pareto optimal solution of the cooperative game problem is obtained in the way of distributed communication and centralized computation.

Through local interaction of microsatellite, the weight value α_{ij} is updated. Defining the variable to be optimized as $\boldsymbol{U}_i = [\boldsymbol{u}_{i1}{}^{\mathrm{T}}, \boldsymbol{u}_{i2}{}^{\mathrm{T}}, \cdots, \boldsymbol{u}_{iN_i}{}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{3N_i\times 1}$ and treating the control torque constraints as soft to enhance solution feasibility, the optimization problem in the lower level is rewritten as

$$\min_{\boldsymbol{U}_i} C_i = \frac{1}{2}\begin{bmatrix} 1 \\ \boldsymbol{U}_i \end{bmatrix}^{\mathrm{T}} \boldsymbol{H}_i \begin{bmatrix} 1 \\ \boldsymbol{U}_i \end{bmatrix} \tag{16}$$

$$s.t.\begin{cases}\mathbf{1}_3\otimes(-u_{max}\mathbf{1}_3)\le \boldsymbol{U}_i\le \mathbf{1}_3\otimes(u_{max}\bar{\mathbf{1}}_3^{\bar{3}}),\forall t\in[t_{wn},t_{wn+1})\\ \boldsymbol{U}_i=\mathbf{0}_{3N_i},\forall t\in[t_{wn+1},t_{(w+1)n})\\ \mathbf{1}_{3\times 3N_i}\boldsymbol{U}_i=\boldsymbol{\tau}_i\end{cases}$$

where $\boldsymbol{H}_i$ is a positive-definite matrix with the following form

$$\boldsymbol{H}_i=\begin{bmatrix}1&0\\0&\boldsymbol{H}_{i1}\end{bmatrix}+\boldsymbol{H}_{i2},$$

$$\boldsymbol{H}_{j1}=\begin{bmatrix}\alpha_{i1}\boldsymbol{I}_3&0&0\\&\ddots&\\0&&\alpha_{iN_i}\boldsymbol{I}_3\end{bmatrix}\in\mathbb{R}^{3N_i\times 3N_i}$$

$$\boldsymbol{H}_{i2}=\begin{bmatrix}\sum\limits_{k\in\mathcal{M}_{\hat{i}}}{}^{i}\boldsymbol{L}_k&0&0&0\\0&{}^{i}\boldsymbol{r}_g^{i1\times}\boldsymbol{C}_{i1}^{bi}&0&0\\0&0&\ddots&0\\0&0&0&{}^{i}\boldsymbol{r}_g^{iN_i\times}\boldsymbol{C}_{iN_i}^{bi}\end{bmatrix}^T\times\begin{bmatrix}\sum\limits_{k\in\mathcal{M}_{\hat{i}}}{}^{i}\boldsymbol{L}_k&0&0&0\\0&{}^{i}\boldsymbol{r}_g^{i1\times}\boldsymbol{C}_{i1}^{bi}&0&0\\0&0&\ddots&0\\0&0&0&{}^{i}\boldsymbol{r}_g^{iN_i\times}\boldsymbol{C}_{iN_i}^{bi}\end{bmatrix}$$

It is an QP problem with linear equality constraints and inequality constraints. It can be solved easily by any one microsatellite in the subsystem.

Remark 1. Eq. (15) is a constrained optimization problem, which can be solved by centralized and distributed optimization [30,31]. In spite of the distributed methods can avoid the computational burden in the central node of centralized method and improve the fault-tolerance, these methods are mainly based on multiple times of iteration, which would increase the communication burden. In this paper, after converting the problem in an QP problem, it can be easily solved by any one microsatellite in the subsystem. Inspired by the consensus algorithm in task allocation, the type of distributed communication for consistent situational awareness inside subsystem and individual strategy computation of QP problem is used in the lower level. It can avoid the low fault-tolerance of centralized method and avoid multiple iterations of the other distributed methods.

3.2. Graphical game in the upper level

3.2.1. Formulation of graphical game among subsystems

Given the desired orbit state of the combination, the aim of the upper level is to get the optimal control strategies of each subsystem which can achieve control object while simultaneously minimizing its individual performance index function. Therefore, the coordination among subsystems is formulated as a graphical game based on the definition of graphical game [32]

(1) Players: All subsystems in $\mathcal{M}$ are viewed as players in the graphical game. Each subsystem is driven by the interconnected dynamics Eq. (2). $\mathcal{E}_{ik}$ in the communicate graph $\mathcal{G}_i$ guarantees that there exists at least one path between i and any $k\in\mathcal{M}_{\hat{i}}$.

(2) Strategies: Each subsystem selects its own control force $\boldsymbol{\tau}_i$ to minimize its individual performance function. The strategy set of each subsystem is within the control force constraints from the lower level, which is formulated as

$$\begin{aligned}&\boldsymbol{\tau}_{min}\le\boldsymbol{\tau}_i(t)\le\boldsymbol{\tau}_{max},\forall t\in[t_{wn},t_{wn+1})\\&\boldsymbol{\tau}_i(t)=\mathbf{0}_{3\times1},\forall t\in[t_{wn+1},t_{(w+1)n})\end{aligned}\tag{17}$$

where $\boldsymbol{\tau}_{min}$ and $\boldsymbol{\tau}_{max}$ can be computed by Eq. (3) and Eq. (7).

(3) Payoff: The individual performance index function of each subsystem that depends on the state error, the control of itself and the control of other subsystems has the form

$$J_i=\int_{t_0}^{t_f}((\boldsymbol{x}_i-\boldsymbol{x}_{id})^{\mathrm{T}}\boldsymbol{Q}_i(\boldsymbol{x}_i-\boldsymbol{x}_{id})+\boldsymbol{\tau}_i^{\mathrm{T}}\boldsymbol{R}_i\boldsymbol{\tau}_i+\sum_{k\in\mathcal{M}_{\hat{i}}}\boldsymbol{\tau}_k^{\mathrm{T}}\boldsymbol{R}_{ik}\boldsymbol{\tau}_k)dt\tag{18}$$

where $\boldsymbol{R}_i$ satisfies $\boldsymbol{\tau}_i^{\mathrm{T}}\boldsymbol{R}_i\boldsymbol{\tau}_i=\sum_{j\in\mathcal{N}_i}\boldsymbol{u}_{ij}^{\mathrm{T}}\boldsymbol{R}_{ij}\boldsymbol{u}_{ij}$ in Eq. (4). Each subsystem seeks to minimize its own individual objective value independently, whose unique solution is the Nash equilibrium.

Definition 2 *(Nash equilibrium solution [29]).* A sequence of control laws $\{\boldsymbol{\tau}_i^*,\boldsymbol{\tau}_{\hat{i}}^*\}$ is known as a global Nash equilibrium solution for all $i\in\mathcal{M}$ if for $\forall i\in\mathcal{M}$ the following condition can be satisfied

$$J_i^*=J_i(\boldsymbol{\tau}_i^*,\boldsymbol{\tau}_{\hat{i}}^*)\le J_i(\boldsymbol{\tau}_i,\boldsymbol{\tau}_{\hat{i}}^*)\tag{19}$$

where $\{J_i^*,J_{\hat{i}}^*\}$ is taken as the Nash equilibrium of the differential game.

Therefore, the graphical game in the upper level can be written in the following form

$$\min_{\boldsymbol{\tau}_i}J_i=\int_{t_0}^{t_f}((\boldsymbol{x}_i-\boldsymbol{x}_{id})^{\mathrm{T}}(\boldsymbol{x}_i-\boldsymbol{x}_{id})+\boldsymbol{\tau}_i^{\mathrm{T}}\boldsymbol{R}_i\boldsymbol{\tau}_i+\sum_{k\in\mathcal{M}_{\hat{i}}}\boldsymbol{\tau}_k^{\mathrm{T}}\boldsymbol{R}_{ik}\boldsymbol{\tau}_k)dt$$
$$s.t.\begin{cases}\dot{\boldsymbol{x}}_i=\boldsymbol{A}_i\boldsymbol{x}_i+\boldsymbol{B}_i\boldsymbol{\tau}_i+\sum\limits_{k\in\mathcal{M}_{\hat{i}}}\boldsymbol{B}_k\boldsymbol{\tau}_k\\ \boldsymbol{\tau}_{min}\le\boldsymbol{\tau}_i(t)\le\boldsymbol{\tau}_{max},\forall t\in[t_{wn},t_{wn+1})\\ \boldsymbol{\tau}_i(t)=\mathbf{0}_{3\times1},\forall t\in[t_{wn+1},t_{(w+1)n})\\ \boldsymbol{x}_i(t_0)=\boldsymbol{x}_{i0}\end{cases}\tag{20}$$

At each control time t_c, each subsystem gets Nash equilibrium strategy by solving the above problems to realize the optimization of their individual objective functions.

Remark 2. Because the number of subsystems is changing with the assembly of substructures, it is desired to get the Nash equilibrium in a distributed manner. The graphical game gives a suitable framework to capture structure that is present in the strategic interaction, which allows the solution of the game in a distributed fashion.

3.2.2. Derivation of decoupled QP problem

The essence of the above graphical game is a set of optimal control problem in which players independently optimize their individual performance index functions. MPC is a well-known optimal control law, which has great performance in dealing with constraints. Accordingly, based on the frame of MPC, the above graphical game is reformulated in the predictive horizon.

Denote the starting and ending time of the control process as t_0 and t_f. The total time interval is divided by sampling period ΔT into N_s sub-intervals evenly, such that $[t_0,t_1,\cdots,t_l,\cdots,t_f]$. Considering the periodic impulsive form, the control input sequences are periodically recomputed at each instant of the impulse $t_c\in\{t_0,t_n,\cdots,t_{wn},\cdots\}$, not at each sampling instant t_l. Thus, both the predictive horizon and the control horizon are set as p times of impulse period, that is the step length of prediction horizon is $N_p=p\times n$.

Firstly, the dynamic model of each subsystem is set up in the predictive horizon by discretizing and stacking the dynamic model Eq. (2). At each sampling time $t_l\in\{t_c,t_{c+1},\cdots,t_{c+N_p-1}\}$, Eq. (2) is discretized with Δt as

$$\boldsymbol{x}_{i,l+1} = \hat{\boldsymbol{A}}_{i,l}\boldsymbol{x}_{i,l} + \hat{\boldsymbol{B}}_{i,l}\boldsymbol{\tau}_{i,l} + \sum_{k\in\mathcal{M}_{\hat{i}}} \hat{\boldsymbol{B}}_{k,l}\boldsymbol{\tau}_{k,l} \tag{21}$$

where $\boldsymbol{x}_{i,l} = \boldsymbol{x}_i(t_l)$, $\boldsymbol{\tau}_{i,l} = \boldsymbol{u}_i(t_l)$ and $\boldsymbol{\tau}_{k,l} = \boldsymbol{\tau}_k(t_l)$. $\hat{\boldsymbol{A}}_{i,l} = e^{\boldsymbol{A}_i\Delta t} \in \mathbb{R}^{6\times6}$, $\hat{\boldsymbol{B}}_{i,l} = \int_0^{\Delta t} e^{\boldsymbol{A}_i\Delta t}\boldsymbol{B}_i dt \in \mathbb{R}^{6\times3}$. By defining the error state at instant t_l as $\boldsymbol{x}_{ie,l+1} = \boldsymbol{x}_{i,l+1} - \boldsymbol{x}_{id}$, one can get the discrete error dynamic model

$$\boldsymbol{x}_{ie,l+1} = \hat{\boldsymbol{A}}_{i,l}\boldsymbol{x}_{i,l} + \hat{\boldsymbol{B}}_{i,l}\boldsymbol{\tau}_{i,l} + \sum_{k\in\mathcal{M}_{\hat{i}}} \hat{\boldsymbol{B}}_{k}\boldsymbol{\tau}_{k,l} - \boldsymbol{x}_{id} \tag{22}$$

To get the stack of dynamic model over the predictive horizon $[t_c, t_{c+N_p-1}]$, the stack of error state vector $\boldsymbol{x}_{ie,c}$, control force $\boldsymbol{\tau}_{i,c}$ of subsystem i and the coupling inputs of other subsystem $\boldsymbol{\tau}_{k,c}$ are defined, respectively

$$\boldsymbol{X}_{ie,c} = [\boldsymbol{x}_{ie,c+1}^{\mathrm{T}}, \boldsymbol{x}_{ie,c+2}^{\mathrm{T}}, \cdots, \boldsymbol{x}_{ie,c+N_p}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{6N_p\times1}$$
$$\boldsymbol{T}_{i,c} = [\boldsymbol{\tau}_{i,c}^{\mathrm{T}}, \boldsymbol{\tau}_{i,c+1}^{\mathrm{T}}, \cdots, \boldsymbol{\tau}_{i,c+N_p-1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{3N_p\times1}$$
$$\boldsymbol{T}_{k,c} = [\boldsymbol{\tau}_{k,c}^{\mathrm{T}}, \boldsymbol{\tau}_{k,c+1}^{\mathrm{T}}, \cdots, \boldsymbol{\tau}_{k,c+N_p-1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{3N_p\times1}$$

Then, one can get the evolution of the error relative motion dynamics over the predictive horizon

$$\boldsymbol{X}_{ie,c} = \boldsymbol{\Lambda}_i\boldsymbol{x}_{i,c} + \boldsymbol{\Xi}_i\boldsymbol{T}_{i,c} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{\Xi}_k\boldsymbol{T}_{k,c} - \boldsymbol{X}_{id} \tag{23}$$

where $\boldsymbol{x}_{i,c}$ is the state vector sampled at the impulsive instant t_c. $\boldsymbol{X}_{id} = \boldsymbol{I}_{N_p} \otimes \boldsymbol{x}_{id}$ is the stack of the invariable desired state. And

$$\boldsymbol{\Lambda}_i = \begin{bmatrix} \hat{\boldsymbol{A}}_{i,c}^{\mathrm{T}} & (\hat{\boldsymbol{A}}_{i,c}^{2})^{\mathrm{T}} & \cdots & (\hat{\boldsymbol{A}}_{i,c}^{N_p})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in {}^{6N_p\times6}$$

$$\boldsymbol{\Xi}_i = \begin{bmatrix} \hat{\boldsymbol{B}}_{i,c} & 0 & 0 & 0 \\ \hat{\boldsymbol{A}}_{i,c}\hat{\boldsymbol{B}}_{i,c} & \hat{\boldsymbol{B}}_{i,c} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ \hat{\boldsymbol{A}}_{i,c}^{N_p-1}\hat{\boldsymbol{B}}_{i,c} & \hat{\boldsymbol{A}}_{i,c}^{N_p-2}\hat{\boldsymbol{B}}_{i,c} & \cdots & \hat{\boldsymbol{B}}_{i,c} \end{bmatrix} \in \mathbb{R}^{6N_p\times3N_p}$$

Secondly, the local performance index function of each subsystem in the predictive horizon is derived from Eq. (18)

$$J_{i,c} = \sum_{g=c}^{c+N_p-1} \boldsymbol{x}_{ie,g+1}^{\mathrm{T}}\boldsymbol{Q}_i\boldsymbol{x}_{ie,g+1} + \boldsymbol{\tau}_{i,g}^{\mathrm{T}}\boldsymbol{R}_i\boldsymbol{\tau}_{i,g} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{\tau}_{k,g}^{\mathrm{T}}\boldsymbol{R}_{ik}\boldsymbol{\tau}_{k,g} \tag{24}$$

According to $\boldsymbol{X}_{ie,c}$ and $\boldsymbol{T}_{i,c}$, the above equation is rewritten in the matrix form

$$J_{i,c} = \boldsymbol{X}_{ie,c}^{\mathrm{T}}\boldsymbol{Q}_{i,c}\boldsymbol{X}_{ie,c} + \boldsymbol{T}_{i,k}^{\mathrm{T}}\boldsymbol{R}_{i,c}\boldsymbol{T}_{i,c} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{T}_{k,c}^{\mathrm{T}}\boldsymbol{R}_{ik,c}\boldsymbol{T}_{k,c} \tag{25}$$

where $\boldsymbol{Q}_{i,c} = \boldsymbol{I}_{N_p} \otimes \boldsymbol{Q}_i$, $\boldsymbol{R}_{i,c} = \boldsymbol{I}_{N_p} \otimes \boldsymbol{R}_i$ and $\boldsymbol{R}_{ik,c} = \boldsymbol{I}_{N_p} \otimes \boldsymbol{R}_{ik}$.

Lastly, by stacking of Eq. (17) in the predictive horizon, the following control constraint is obtained

$$\boldsymbol{1}_p \otimes \begin{bmatrix} \boldsymbol{\tau}_{min} \\ \boldsymbol{0}_{3(n-1)\times1} \end{bmatrix} \le \boldsymbol{T}_{i,c} \le \boldsymbol{1}_p \otimes \begin{bmatrix} \boldsymbol{\tau}_{max} \\ \boldsymbol{0}_{3(n-1)\times1} \end{bmatrix} \tag{26}$$

Combining Eq. (23), (25) and (26), the following game formulation in the predictive horizon is obtained

$$\min_{\boldsymbol{T}_{i,c}} J_{i,c} = \boldsymbol{X}_{ie,c}^{\mathrm{T}}\boldsymbol{Q}_{i,c}\boldsymbol{X}_{i,c} + \boldsymbol{T}_{i,c}^{\mathrm{T}}\boldsymbol{R}_{i,c}\boldsymbol{T}_{i,c} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{T}_{k,c}^{\mathrm{T}}\boldsymbol{R}_{k,c}\boldsymbol{T}_{k,c}$$
$$s.t. \begin{cases} \boldsymbol{X}_{ie,c} = \boldsymbol{\Lambda}_i\boldsymbol{x}_{i,c} + \boldsymbol{\Xi}_{i,c}\boldsymbol{T}_{i,c} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{\Xi}_{k,c}\boldsymbol{T}_{k,c} - \boldsymbol{X}_{id} \\ \boldsymbol{1}_p \otimes \begin{bmatrix} \boldsymbol{\tau}_{min} \\ \boldsymbol{0}_{3(n-1)\times1} \end{bmatrix} \le \boldsymbol{T}_{i,c} \le \boldsymbol{1}_p \otimes \begin{bmatrix} \boldsymbol{\tau}_{max} \\ \boldsymbol{0}_{3(n-1)\times1} \end{bmatrix} \end{cases} \tag{27}$$

Based on the measured state of current impulsive instant t_c, the optimal control sequence $\boldsymbol{T}_{i,c}$ of each subsystem can be obtained by optimizing the above local performance index function with constraints over a finite predictive horizon and the first $3n$ items will be passed to the lower level.

However, it is noted that the local performance index function and the constraints of subsystem i are coupled with all other subsystems, which cannot be solved independently by each subsystem. It is desired to decouple the above problem to allow the solution of the game in a distributed fashion.

In order to deal with the coupled dynamics constraint in the predictive horizon, the local performance index function becomes the following form by substituting Eq. (23) into Eq. (25)

$$\begin{aligned} J_{i,c} = {} & (\boldsymbol{x}_{i,c}^{\mathrm{T}}\boldsymbol{\Lambda}_i^{\mathrm{T}} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{T}_{k,c}^{\mathrm{T}}\boldsymbol{\Xi}_{k,c}^{\mathrm{T}} - \boldsymbol{X}_{id}^{\mathrm{T}})\boldsymbol{Q}_{i,c}(\boldsymbol{\Lambda}_i\boldsymbol{x}_{i,c} \\ & + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{\Xi}_k\boldsymbol{T}_{k,c} - \boldsymbol{X}_{id}) + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{T}_{k,c}^{\mathrm{T}}\boldsymbol{R}_{ik,c}\boldsymbol{T}_{k,c} \\ & + 2(\boldsymbol{x}_{i,c}^{\mathrm{T}}\boldsymbol{\Lambda}_i^{\mathrm{T}} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{T}_{k,c}^{\mathrm{T}}\boldsymbol{\Xi}_k^{\mathrm{T}} - \boldsymbol{X}_{id}^{\mathrm{T}})\boldsymbol{Q}_{i,c}\boldsymbol{\Xi}_i\boldsymbol{T}_{i,c} \\ & + \boldsymbol{T}_{i,c}^{\mathrm{T}}(\boldsymbol{R}_{i,c} + \boldsymbol{\Xi}_{i,c}^{\mathrm{T}}\boldsymbol{Q}_{i,c}\boldsymbol{\Xi}_i)\boldsymbol{T}_{i,c} \end{aligned} \tag{28}$$

From the perspective of subsystem i, it can just optimize its own control sequence $\boldsymbol{T}_{i,c}$. For this reason, the terms which do not contain $\boldsymbol{T}_{i,c}$ in the local performance index function Eq. (28) can be omitted. Then, one can get the following local performance index function

$$\begin{aligned} J_{i,c} = {} & 2(\boldsymbol{x}_{i,c}^{\mathrm{T}}\boldsymbol{\Lambda}_i^{\mathrm{T}} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{T}_{k,c}^{\mathrm{T}}\boldsymbol{\Xi}_j^{\mathrm{T}} - \boldsymbol{X}_{id}^{\mathrm{T}})\boldsymbol{Q}_{i,c}\boldsymbol{\Xi}_i\boldsymbol{T}_{i,c} \\ & + \boldsymbol{T}_{i,c}^{\mathrm{T}}(\boldsymbol{R}_{i,c} + \boldsymbol{\Xi}_{i,c}^{\mathrm{T}}\boldsymbol{Q}_{i,c}\boldsymbol{\Xi}_i)\boldsymbol{T}_{i,c} \end{aligned} \tag{29}$$

If the optimal solution of other subsystem $\boldsymbol{T}_{k,c}^{*}, k \in \mathcal{M}_{\hat{i}}$ is known, the optimization problem of subsystem i in the predictive horizon can be transferred into the following QP problem

$$\begin{aligned} & \min_{\boldsymbol{T}_{i,c}} J_{i,c} = \tfrac{1}{2}\boldsymbol{T}_{i,c}^{\mathrm{T}}\boldsymbol{M}_{i2}\boldsymbol{T}_{i,c} + \boldsymbol{M}_{i1}^{\mathrm{T}}\boldsymbol{T}_{i,c} \\ & s.t. \boldsymbol{1}_p \otimes \begin{bmatrix} \boldsymbol{\tau}_{min} \\ \boldsymbol{0}_{3(n-1)\times1} \end{bmatrix} \le \boldsymbol{T}_{i,c} \le \boldsymbol{1}_p \otimes \begin{bmatrix} \boldsymbol{\tau}_{max} \\ \boldsymbol{0}_{3(n-1)\times1} \end{bmatrix} \end{aligned} \tag{30}$$

where $\boldsymbol{M}_{i2,c} = 2(\boldsymbol{R}_{i,c} + \boldsymbol{\Xi}_i^{\mathrm{T}}\boldsymbol{Q}_{i,c}\boldsymbol{\Xi}_i) \in \mathbb{R}^{3N_p\times3N_p}$ and $\boldsymbol{M}_{i1,c} = 2\boldsymbol{\Xi}_i^{\mathrm{T}}\boldsymbol{Q}_{i,c}(\boldsymbol{\Lambda}\boldsymbol{x}_{i,c} + \sum_{k\in\mathcal{M}_{\hat{i}}} \boldsymbol{\Xi}_k\boldsymbol{T}_{k,c}^{*} - \boldsymbol{X}_{id}) \in \mathbb{R}^{3N_p\times1}$. It can be solved in real time using any number of generic solvers.

Remark 3. Due to the decoupled optimization problem, the Nash equilibrium of the upper level can be got in a distributed manner, which can improve the autonomy of each subsystem. Due to the simple QP formulation of each subsystem, the low computational complexity enables microsatellites to compute online.

3.3. Bi-level predictive game control algorithm

As can be seen from Eq. (16), in order to obtain the optimal control input of each microsatellite at current impulsive period, the optimal control sequence of subsystem to which it belongs and the control torques of all other subsystems are required from the upper level. Therefore, the problem by Eq. (16) can be denoted as

$$\boldsymbol{u}_{ij,c} = f_{low}(\boldsymbol{T}_{i,c}, {}^{i}\boldsymbol{L}_{\hat{i},c}) \tag{31}$$

Moreover, it is noticed that in Eq. (30), in order to get the optimal control sequence of subsystem i at current impulsive instant, the optimal control sequences of other subsystem $\boldsymbol{T}_{\hat{i},k}^{*}$ are required.

That is, the optimal control sequence of subsystem i is a function of the control sequences of other subsystems

$$\boldsymbol{T}_{i,c} = f_{up}(\boldsymbol{T}_{\hat{i},c}) \tag{32}$$

However, at the beginning of optimization, the control sequences and the generated control torque of other subsystems are unknown. Therefore, Algorithm 1 is proposed to approximate the equilibrium solution at each impulsive instant. After finite asynchronous iterations, the control sequence of each subsystem would approach equilibrium strategy in a certain precision.

Algorithm 1 Bi-level game-based control algorithm.

Input:

1: Update the state $\boldsymbol{x}_{i,c}$ at the impulsive instant;
2: Update the matrices $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{R}_i$, $\boldsymbol{Q}_i$;
3: Set the initial admissible strategies of each subsystem $\boldsymbol{T}_{i,c}^{[0]} = \boldsymbol{0}_{3N_p}, i \in \mathcal{M}$
4: Set the computing precision ε and maximum iteration times $iter_{max}$

Output:

5: For $iter > 0$, subsystem i solves $\boldsymbol{T}_{i,c}^{[iter]}$ at the upper level using

$$\boldsymbol{T}_{i,c}^{[iter]} = f_{up}(\boldsymbol{T}_{\hat{i},c}^{[iter-1]}) \tag{33}$$

and transmit the first $3n$ terms of $\boldsymbol{T}_{i,c}^{[iter]}$ and the control torque generated by other subsystems ${}^{i}\boldsymbol{L}_{i,c}^{[iter-1]}$ to the lower level;

6: Microsatellite A_j^i gets $\boldsymbol{u}_{ij,c}^{[iter]}$ and ${}^{i}\boldsymbol{L}_{i,c}^{[iter-1]}$ at the lower level using

$$\boldsymbol{u}_{ij,c}^{[iter]} = f_{low}(\boldsymbol{T}_{i,c}^{[iter]}, {}^{i}\boldsymbol{L}_{i,c}^{[iter-1]}) \tag{34}$$

and transmit the $\boldsymbol{L}_{i,c}^{[iter]}$ to the upper level;

7: Broad $\boldsymbol{T}_{i,c}^{[iter]}$ and $\boldsymbol{L}_{i,c}^{[iter]}$ to other subsystems and receive the information about other subsystems by communication topology at the upper level;
8: Update the differences of local performance index value between two serial control policy adjustments

$$E_{J^{i,c}} = |J_{i,c}^{iter} - J_{i,c}^{iter-1}| \tag{35}$$

9: If $maxE_{J^{i,k}} \leq \varepsilon$ or $iter > iter_{max}$, stop iterations and microsatellite A_j^i uses control strategy to implement orbit control; else, let $iter = iter + 1$ and return to step 5.

Theorem 1. *Assume that microsatellite A_j^i performs Algorithm 1, then, the algorithm converges to the unique equilibrium point after finite times of iteration.*

Proof. The iteration in Algorithm 1 is employed to realize the distributed optimization of the upper level and the lower game is nested to the iteration. Therefore, the proof will focus on the upper level.

For the existence of the Nash equilibrium, refer to [33], the graphical game is a special case of the non-cooperative game. It is noted that the existence of Nash equilibrium of non-cooperative game should follow the following statement: (1) The strategy space of each player is a nonempty convex and compact subset of Euclidean space. (2) The function $J_{i,c}(\boldsymbol{T}_{i,c})$ is continuous about $\{\boldsymbol{T}_{i,c}, \boldsymbol{T}_{\hat{i},c}\}$ and convex in $\boldsymbol{T}_{i,c}$. For the first statement, it is apparent from Eq. (30) that the strategy space of upper game is nonempty, convex, and compact subset of Euclidean space. For the second statement, taking second-order derivative of $J_{i,c}$ with respect to $\boldsymbol{T}_{i,c}$ yields

$$\frac{\partial^2 J_{i,c}}{(\partial \boldsymbol{T}_{i,c})^2} = \boldsymbol{M}_{i2,c} \tag{36}$$

It is noticed that $\boldsymbol{M}_{i2,c}$ is always positive definite, which illustrates that $J_{i,c}$ is convex in $\boldsymbol{T}_{i,c}$. Therefore, Nash equilibrium exists in the game of the upper game.

Then, the convergence of iteration for the upper game is proved. It can be seen from the proposed performance index value of each subsystem that the objective of the subsystem i is the same with any other subsystem $\hat{i}$. Therefore, if each J_i of subsystem i is optimized independently to get $\boldsymbol{T}_{i,c}$, then $J_{\hat{i}}$ which is influenced by $\boldsymbol{T}_{i,c}$ would be more optimal. Then, after any round of iteration, if subsystem i can obtain all the required latest information of all other subsystems, for each subsystem

$$J_i^{iter} \leq J_i^{iter-1} \tag{37}$$

Therefore, it holds

$$\lim_{iter \to \infty} J_i^{iter} = J_i^* \tag{38}$$

where J_i^* is corresponding to Nash equilibrium of the upper game. This completes the proof of convergence for the upper game. □

Remark 4. After the strategy of subsystem i is updated at this iteration, it releases this information to its neighborhood. Here, one microsatellite in each subsystem as the role of supervisor carries out the computation and information exchange. In order to enable that each subsystem can obtain all the required information of all other subsystems, the undirected connected graph between supervisors is required. It enables the subsystem to acquire information of all other subsystems for individual control strategy calculation. And the communication topology can be further optimized based on minimum cost arborescence or other techniques [34].

Remark 5. It is noticed that a maximum iteration time $iter_{max}$ is introduced for online implementation, which is to guarantee the real-time property of the method.

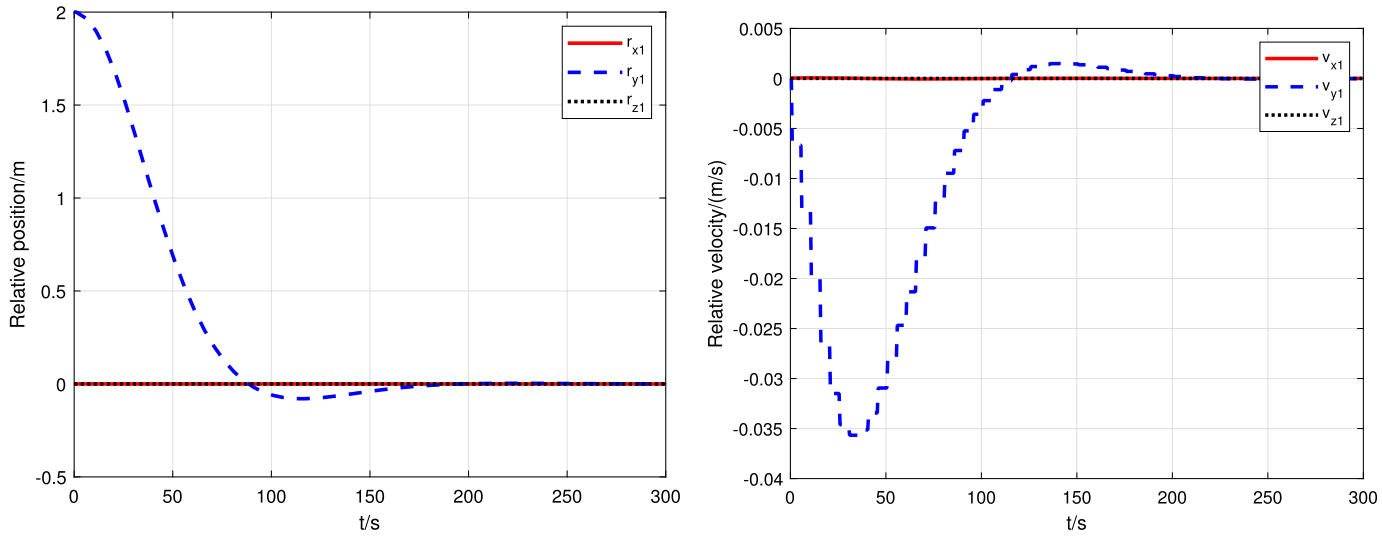
Remark 6. If there is only one microsatellites in one substructure, the lower game does not exist and the control torque constraints of this subsystem should be added into the upper game to be satisfied. If there is no microsatellites attached to one substructure, it can be taken as a player with zero-input strategy in the upper level.

4. Simulations

In this section, numerical simulations are conducted to demonstrate the performance of the proposed reconfigurable control. In order to show the rationality of the bi-level design, we will compare the proposed method with the single lower game-based method which takes the combination as only one subsystem, the single upper game-based method which takes each microsatellite as one subsystem and the traditional allocation-based control which uses linear quadratic regulation as control command and sequential quadratic programming as control allocation.

Assuming that the circular orbit radius of the combination is $r_0 = 42164$ km, then the orbit angular velocity is $n_0 = 7.2922 \times 10^{-5}$ rad/s. The orbit inclination is 30°, the longitude of the ascending node is 20°. Without loss of generality, there is one substructure in the combination. There are four microsatellites attached to this substructure, which is taken as one of subsystems s_1 with the mass of 100 kg. The positions of the CoM of microsatellites in $O_{bs_1}x_{bs_1}y_{bs_1}z_{bs_1}$ are ${}^{s_1}\boldsymbol{r}_{s_1}^{s_1 1} = [0, -1, 1]^{\mathrm{T}}$, ${}^{s_1}\boldsymbol{r}_{s_1}^{s_1 2} = [0, 1, 1]^{\mathrm{T}}$, ${}^{s_1}\boldsymbol{r}_{s_1}^{s_1 3} = [0, -1, -1]^{\mathrm{T}}$ and ${}^{s_1}\boldsymbol{r}_{s_1}^{s_1 4} = [0, 1, -1]^{\mathrm{T}}$. The transition matrices of microsatellites are

$$\boldsymbol{C}_{s_1 1}^{bs_1} = \boldsymbol{C}_{s_1 2}^{bs_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{C}_{s_1 3}^{bs_1} = \boldsymbol{C}_{s_1 4}^{bs_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$



(a) relative orbit postion (b) relative orbit velocity

Fig. 4. Relative orbit state of subsystem s_1.

It is assumed that two substructures are assembled to the combination at the same time with the assistance of four microsatellites. These two subsystems with the mass of 100 kg are denoted as s_2 and s_3. The position of the CoM of microsatellite A_1^{s2} and A_2^{s2} in $O_{bs_2}x_{bs_2}y_{bs_2}z_{bs_2}$ is ${}^{s_2}\boldsymbol{r}_{s_2}^{s_21} = [-1, 0, 0]^{\mathrm{T}}$ and ${}^{s_2}\boldsymbol{r}_{s_2}^{s_22} = [1, 0, 0]^{\mathrm{T}}$. The position of the CoM of microsatellite $A_1^{s_3}$ and $A_2^{s_3}$ in $O_{bs_3}x_{bs_3}y_{bs_3}z_{bs_3}$ is ${}^{s_3}\boldsymbol{r}_{s_3}^{s_31} = [-1, 0, 0]^{\mathrm{T}}$ and ${}^{s_3}\boldsymbol{r}_{s_3}^{s_32} = [1, 0, 0]^{\mathrm{T}}$. The transition matrices of microsatellites are

$$\boldsymbol{C}_{s_21}^{bs_2} = \boldsymbol{C}_{s_31}^{bs_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{C}_{s_22}^{bs_2} = \boldsymbol{C}_{s_32}^{bs_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

At the beginning of the post-assembly stage, there is an unexpected position deviation of the combination due to the mass properties changes. For this reason, the initial orbit state of subsystem s_1 is set as $\boldsymbol{x}_{s_1}(t_0) = [0, 2, 0, 0, 0, 0]^{\mathrm{T}}$. The desired orbit state of s_1 is the original point. Because of the fixed connection between substructures, the initial/desired orbit state of other two subsystems can be got by relative relationship. The relative position relationship among s_1 and s_2 is $\boldsymbol{x}_{s_2} - \boldsymbol{x}_{s_1} = [0, -3, 0]^{\mathrm{T}}$ and among s_1 and s_3 is $\boldsymbol{x}_{s_3} - \boldsymbol{x}_{s_1} = [0, 3, 0]^{\mathrm{T}}$. The transition matrices from other subsystems to s_1 are set as

$$\boldsymbol{C}_{bs_2}^{bs_1} = \boldsymbol{C}_{bs_3}^{bs_1} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

The amplitude of microsatellites' control force is set as $u_{max} = 0.7$ N. Each impulsive thrust action lasts for $\Delta t = 0.5$ s, which means that the time interval is $\Delta t = 0.5$ s. Set the sampling times of the impulsive period is $n = 10$, then the impulsive period is $T = 5$ s.

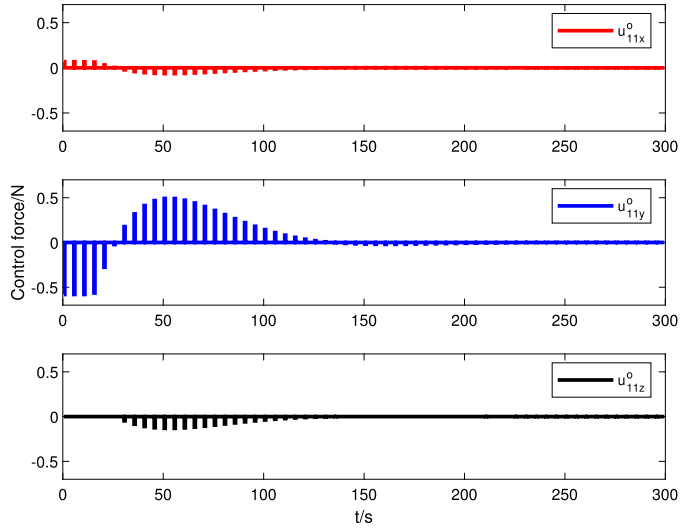
To implement the bi-level game-based control, the weighting matrices are chosen as $\boldsymbol{Q}_{s1} = \boldsymbol{Q}_{s2} = \boldsymbol{Q}_{s3} = 0.0009\boldsymbol{I}_6$ and $\boldsymbol{R}_{s1} = \boldsymbol{R}_{s2} = \boldsymbol{R}_{s3} = 0.001\boldsymbol{I}_3$. The predictive horizons are $N_p = 200$. Error tolerance $\varepsilon = 10^{-6}$ and maximum iteration time $iter_{max} = 3$.

Based on the above settings, the relative orbit position and relative orbit velocity of each subsystem are driven by the bi-level game-based controller. Because of the fixed relative positional relationship between subsystems, only the state of subsystem s_1 is presented in Fig. 4. As we can see, the states of the proposed method converge to the desired state as time increases.

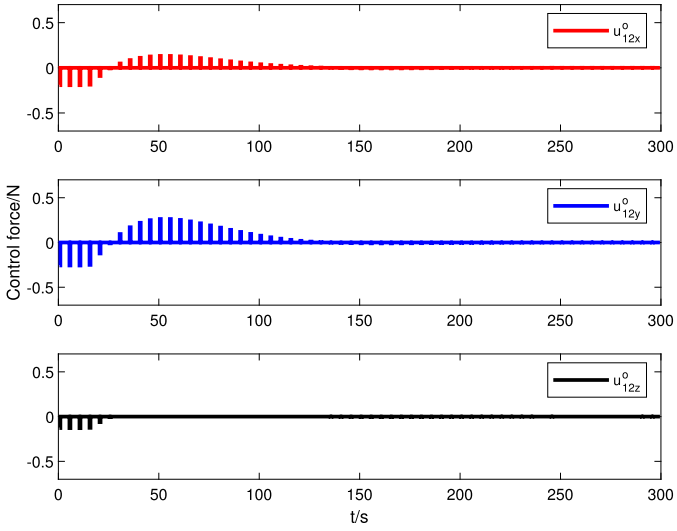
Figs. 5-7 present the control force produced by microsatellites, respectively. It is clearly seen that the control forces generated by each microsatellite are periodic and always within the predefined boundary and the control forces produced along the z-axis are not positive. Therefore, the proposed method can satisfy the control force constraint described by Eq. (7). Fig. 8 presents the resultant control torque generated by microsatellites, respectively. It is apparent that the control torques constraints by Eq. (6) is satisfied. The above results illustrate that the designed method has superior capability to handle the control constraints.

The simulation results of the three comparison methods are also given. The time responses of the relative orbit state of these two methods are shown in Fig. 9. It can be seen that the states of these three methods can also converge to the desired state at last. To further compare these methods, several performance evaluation metrics are set up, and the comparison results are presented in Table 1. Define the evaluation function of fuel consumption of microsatellite A_j^i as $P_j^i = \int_{t_0}^{t_f} ||\boldsymbol{u}_{ij}||dt$ and use the variance σ to imply the energy balance.

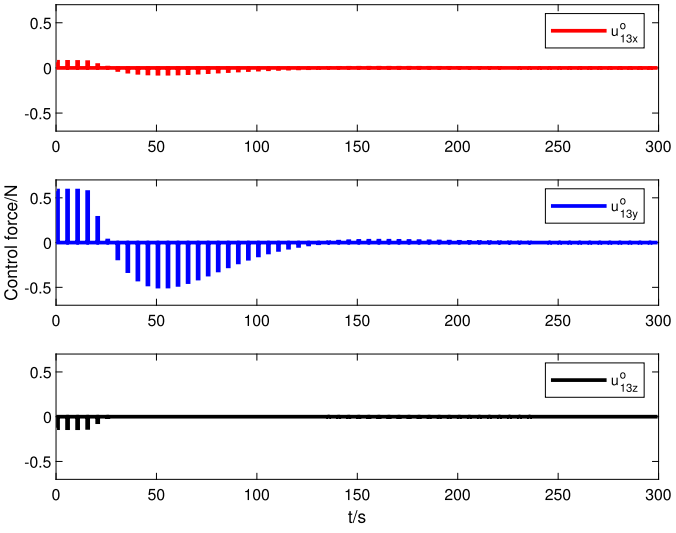
Firstly, we compare the game-based methods with the traditional method. From the perspective of constraints satisfaction, there exists constraints violation of the traditional method. From the perspective of fuel consumption, the performance index value of fuel consumption by the game-based method is lower than that by the traditional method with continuous thrust. From the perspective of computational and communication, the information exchange and control updating of the traditional method take place at each sampling time, while that of the other methods take place only at the impulsive time. Generally speaking, the traditional method can realize a better optimization results. However, the introduction of periodic impulsive thrust form can save fuel consumption and reduce the computational and communication burden to a certain extent. The more serious defect of the traditional method is the dependence of central node. As the number of microsatellites increases, the information about each microsatellite that the central node has to collect, handle and broad will increase as well. Once the central node cannot deal with the process burden, the whole system would be failure. Conversely, the game-based methods are based on the individual optimization, which can avoid the centralized control command calculation and control allocation. Then, not only the fault-tolerance is guaranteed, but



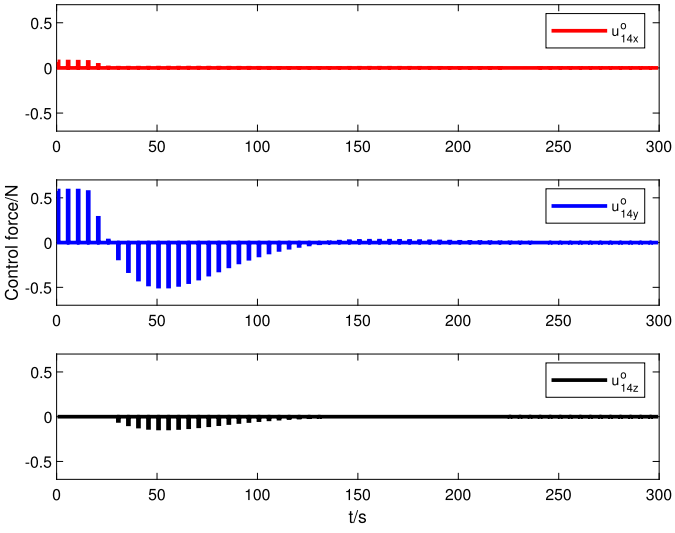
(a) microsatellite $A_1^{s_1}$



(b) microsatellite $A_2^{s_1}$

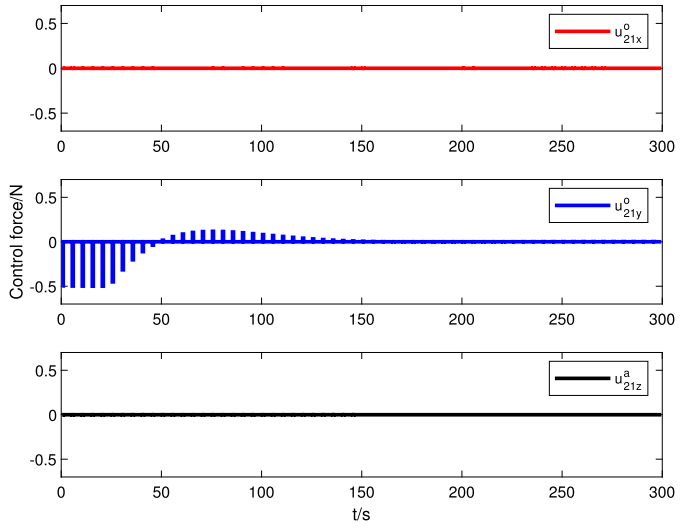


(c) microsatellite $A_3^{s_1}$

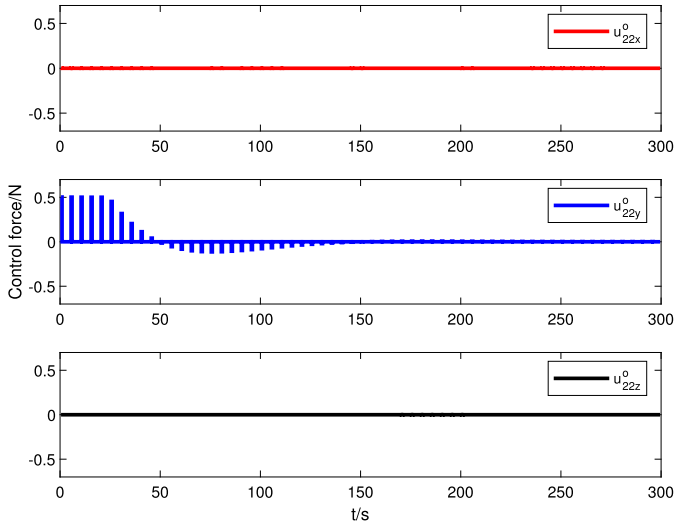


(d) microsatellite $A_4^{s_1}$

Fig. 5. Control force produced by s_1.

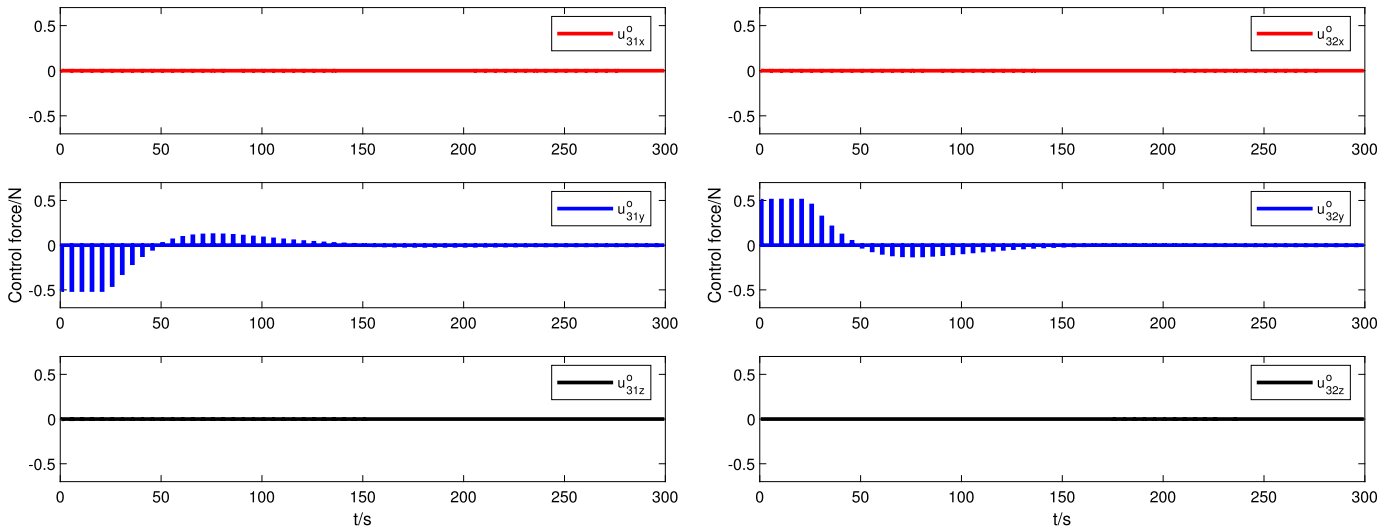


(a) microsatellite $A_1^{s_2}$



(b) microsatellite $A_2^{s_2}$

Fig. 6. Control force produced by s_2.



(a) microsatellite $A_1^{s_3}$ (b) microsatellite $A_2^{s_3}$

Fig. 7. Control force produced by s_3.

Table 1
Comparison of performance properties for different methods.

Performance values	Bi-layer game method	Upper-game method	Lower-game method	Traditional method
Constraint violation	No	No	No	Yes
Fuel consumption P	41.027	43.013	35.785	51.096
Energy balance σ	0.757	30.141	0.174	0.013

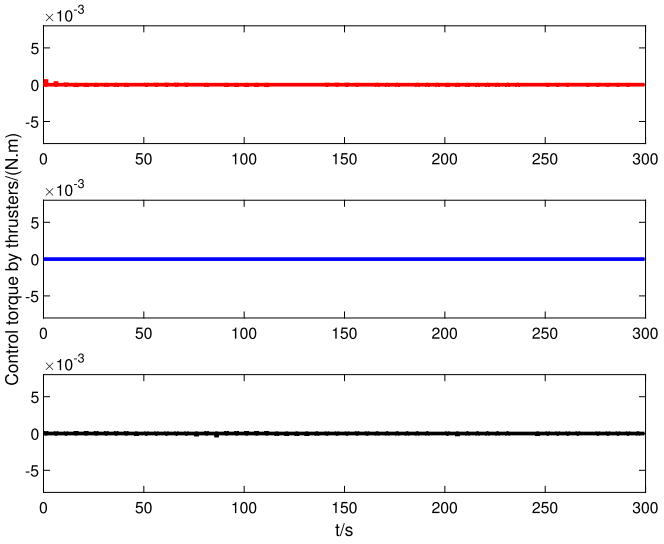


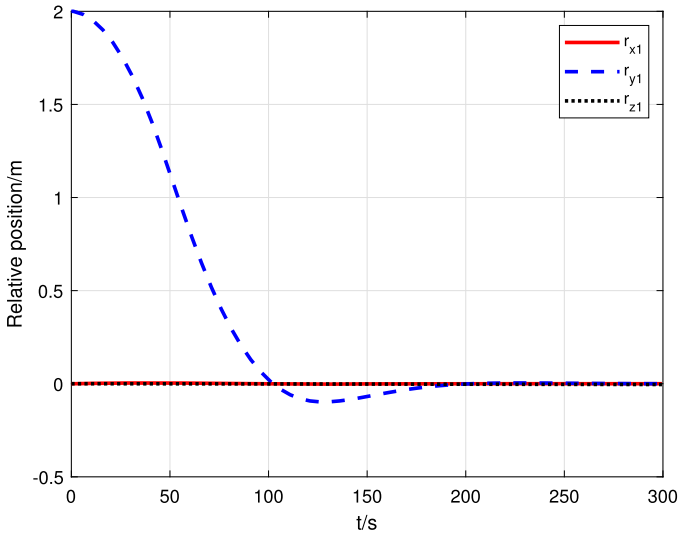
Fig. 8. Control torque generated by thrusters.

also the individual objective of each microsatellite can be considered.

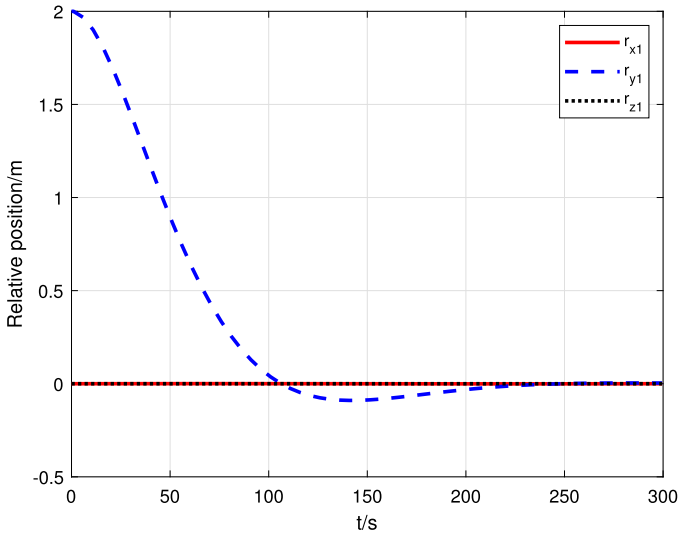
Secondly, we compare the proposed method with other two single-level game-based methods. From the perspective of constraints violation, the results in Table 1 show that the single upper game and the single lower game can satisfy the control constraints. These results indirectly show that each layer of the bi-level game-based control can converge to their respective equilibrium. From the perspective of consumption and energy balance, among these three methods, the consumption of the single lower game-based method is the lowest. That is because the single lower game-based method can get the control strategy by global optimization. Because of the introducing of the energy balance factor of the single lower game-based method and the bi-level method, the consumption of each microsatellite is basically balanced, which is shown in the variance value. The single upper game-based method is in a distributed manner. Through analysis, it is found that a larger $iter_{max}$ will lead to a smaller variance. Next, the analysis is from the perspective of communication and computation. For the single lower game-based method, it has to deal with the extra dynamic constraints. For the single upper game-based method, because each microsatellite has to get the control strategy of all other microsatellites at each iteration, the information package exchanged among microsatellites includes not only its own information but also the information got from its neighbors. By combining these two methods creatively, the bi-level game-based control can avoid the disadvantage of these two methods. In the upper level, the supervisor of one subsystem only needs to get the information from other subsystems not specific to individuals, which reduces the communication burden of microsatellites. In the lower level, the number of microsatellites in each subsystem would not increase and the dynamic constraint is turned into a linear constraint by the upper level, which avoids the individual burden.

5. Conclusions

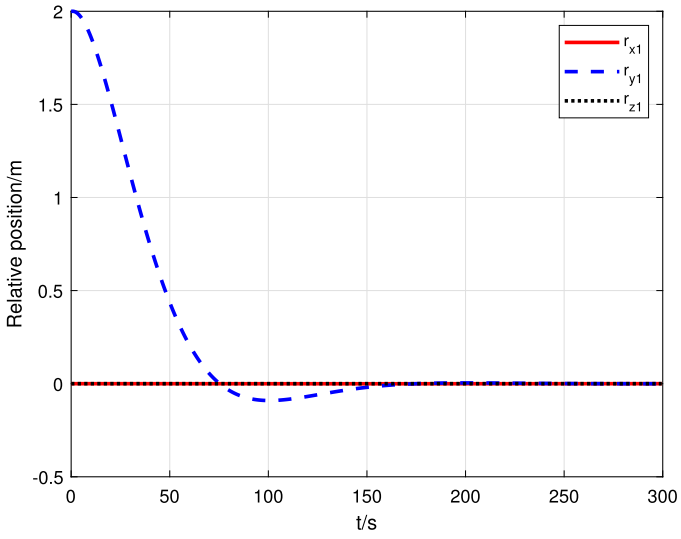
In this paper, a bi-level game-based control method for servicing microsatellites is developed to realize the reconfigurable control every time substructures are assembled to the combination consisting of the assembled substructures. Under the bi-level game formulation, the different objectives of subsystems and microsatellites can be fully considered in different levels to retain the autonomy of each substructure. The game equilibrium of the bi-level game can be computed by the proposed iteration control algorithm, which gives consideration to both flexibility and online burden. Through the numerical examples, compared with the single upper-game based method and the single lower game-based method, the effectiveness of the method is validated. Meanwhile, it



(a) Single upper-game method



(b) Single lower-game method



(c) Traditional method

Fig. 9. Relative orbit state of other methods.

should be pointed that the constraints such as the obstacle avoidance constraint and nonlinear dynamics are not given enough consideration in this paper. In future, it is expected to extend the work to these much complex engineering practice requirements.

Declaration of competing interest

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

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 12072269, 61690210 and 61690211, the Science Foundation of Science, Technology and Innovation Commission of Shenzhen Municipality under Grant No. JCYJ20180508151938535, and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University under Grant No. CX202020.

References

[1] R. Bevilacqua, M. Romano, F. Curti, A.P. Caprari, V. Pellegrini, Guidance navigation and control for autonomous multiple spacecraft assembly: analysis and experimentation, Int. J. Aerosp. Eng. 2011 (2011) 1–18, https://doi.org/10.1155/2011/308245.

[2] C. Underwood, S. Pellegrino, V.J. Lappas, C.P. Bridges, J. Baker, Using cubesat/micro-satellite technology to demonstrate the autonomous assembly of a reconfigurable space telescope (AAReST), Acta Astronaut. 114 (2015) 112–122, https://doi.org/10.1016/j.actaastro.2015.04.008.

[3] N. Han, J. Luo, Z. Zheng, J. Sun, Distributed cooperative game method for attitude takeover of failed satellites using nanosatellites, Aerosp. Sci. Technol. 106 (2020) 106151, https://doi.org/10.1016/j.ast.2020.106151.

[4] Y. Chai, J. Luo, N. Han, J. Sun, Robust event-triggered game-based attitude control for on-orbit assembly, Aerosp. Sci. Technol. 103 (2020) 105894, https://doi.org/10.1016/j.ast.2020.105894.

[5] K. Xia, Y. Zou, Adaptive fixed-time fault-tolerant control for noncooperative spacecraft proximity using relative motion information, Nonlinear Dyn. 100 (11) (2020) 2521–2535, https://doi.org/10.1007/s11071-020-05634-2.

[6] X. Huang, J.D. Biggs, G. Duan, Post-capture attitude control with prescribed performance, Aerosp. Sci. Technol. 96 (2020) 105572, https://doi.org/10.1016/j.ast.2019.105572.

[7] S. Mohan, D.W. Miller, Spheres reconfigurable control allocation for autonomous assembly, in: AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA, 2008.

[8] C. Toglia, F. Kennedy, S. Dubowsky, Cooperative control of modular space robots, Auton. Robots 31 (2–3) (2011) 209–221, https://doi.org/10.1007/s10514-011-9238-z.
[9] C.M. Jewison, B. Mccarthy, D.C. Sternberg, S. D., C. Fang, Resource aggregated reconfigurable control and risk-allocative path planning for on-orbit servicing and assembly of satellites, in: AIAA Guidance, Navigation, and Control Conference, AIAA, 2014.
[10] D. Zhang, S. Zhang, Z. Wang, B. Lu, Dynamic control allocation algorithm for a class of distributed control systems, Int. J. Control. Autom. Syst. 18 (2020) 259–270, https://doi.org/10.1007/s12555-017-9768-z.
[11] Q. Hu, B. Li, A. Zhang, Robust finite-time control allocation in spacecraft attitude stabilization under actuator misalignment, Nonlinear Dyn. 73 (2013) 53–71, https://doi.org/10.1007/s11071-013-0766-2.
[12] A.R. Leite, F. Enembreck, J.-P.A. Barthès, Distributed constraint optimization problems: review and perspectives, Expert Syst. Appl. 41 (11) (2014) 5139–5157, https://doi.org/10.1016/j.eswa.2014.02.039.
[13] X. Lang, A. de Ruiter, A control allocation scheme for spacecraft attitude stabilization based on distributed average consensus, Aerosp. Sci. Technol. 106 (2020) 106173, https://doi.org/10.1016/j.ast.2020.106173.
[14] H. Chang, P. Huang, Y. Zhang, Z. Meng, Z. Liu, Distributed control allocation for spacecraft attitude takeover control via cellular space robot, J. Guid. Control Dyn. 41 (11) (2018) 2499–2506, https://doi.org/10.2514/1.G003626.
[15] W. Liu, Y. Geng, B. Wu, J.D. Biggs, Distributed constrained control allocation for cellularized spacecraft attitude control system, J. Guid. Control Dyn. 45 (2) (2022) 385–393, https://doi.org/10.2514/1.G006266.
[16] K.G. Vamvoudakis, H. Modares, B. Kiumarsi, F.L. Lewis, Game theory-based control system algorithms with real-time reinforcement learning: how to solve multiplayer games online, IEEE Control Syst. Mag. 37 (1) (2017) 33–52, https://doi.org/10.1109/MCS.2016.2621461.
[17] H. Ren, H. Zhang, Y. Wen, C. Liu, Integral reinforcement learning off-policy method for solving nonlinear multi-player nonzero-sum games with saturated actuator, Neurocomputing 335 (2019) 96–104, https://doi.org/10.1016/j.neucom.2019.01.033.
[18] Q. Zhang, D. Zhao, Y. Zhu, Data-driven adaptive dynamic programming for continuous-time fully cooperative games with partially constrained inputs, Neurocomputing 238 (2017) 377–386, https://doi.org/10.1016/j.neucom.2017.01.076.
[19] J. Sun, C. Liu, Distributed zero-sum differential game for multi-agent systems in strict-feedback form with input saturation and output constraint, Neural Netw. 106 (2018) 8–19, https://doi.org/10.1016/j.neunet.2018.06.007.
[20] G. Wen, X. Fang, J. Zhou, J. Zhou, Robust formation tracking of multiple autonomous surface vessels with individual objectives: a noncooperative game-based approach, Control Eng. Pract. 119 (2022) 104975, https://doi.org/10.1016/j.conengprac.2021.104975.
[21] G. Tzannetos, P. Marantos, K.J. Kyriakopoulos, A competitive differential game between an unmanned aerial and a ground vehicle using model predictive control, in: 2016 24th Mediterranean Conference on Control and Automation (MED), 2016, pp. 1053–1058.
[22] H. Jiang, H. Zhang, X. Xie, J. Han, Neural-network-based learning algorithms for cooperative games of discrete-time multi-player systems with control constraints via adaptive dynamic programming, Neurocomputing 344 (2019) 13–19, https://doi.org/10.1016/j.neucom.2018.02.107.
[23] Q. Zhang, D. Zhao, Y. Zhu, Data-driven adaptive dynamic programming for continuous-time fully cooperative games with partially constrained inputs, Neurocomputing 238 (2017) 377–386, https://doi.org/10.1016/j.neucom.2017.01.076.
[24] H. Zhang, C. Jiang, N.C. Beaulieu, X. Chu, X. Wang, T.Q.S. Quek, Resource allocation for cognitive small cell networks: a cooperative bargaining game theoretic approach, IEEE Trans. Wirel. Commun. 14 (6) (2015) 3481–3493, https://doi.org/10.1109/TWC.2015.2407355.
[25] J.M. Maestre, D. Munoz de la Pena, A. Jimenez Losada, E. Algaba, E. Camacho, A coalitional control scheme with applications to cooperative game theory, Optim. Control Appl. Methods 35 (5) (2015) 592–608, https://doi.org/10.1002/oca.2090.
[26] X. Yang, J. Yu, H. Gao, An impulse control approach to spacecraft autonomous rendezvous based on genetic algorithms, Neurocomputing 77 (1) (2012) 189–196, https://doi.org/10.1016/j.neucom.2011.09.009.
[27] D. Dimitrov, Models in Cooperative Game Theory, Springer, Berlin Heidelberg, 2005.
[28] M. Farina, P. Amato, On the optimal solution definition for many-criteria optimization problems, in: Fuzzy Information Processing Society, Nafips Meeting of the, North American, 2002, pp. 233–238.
[29] Engwerda, LQ Dynamic Optimization and Differential Games, WILEY, 2010.
[30] W. Lin, Differential games for multi-agent systems under distributed information.
[31] T. Hatanaka, N. Chopra, T. Ishizaki, N. Li, Passivity-based distributed optimization with communication delays using PI consensus algorithm, IEEE Trans. Autom. Control 63 (12) (2018) 4421–4428, https://doi.org/10.1109/TAC.2018.2823264.
[32] M.I. Abouheaf, F.L. Lewis, K.G. Vamvoudakis, S. Haesaert, R. Babuska, Multi-agent discrete-time graphical games and reinforcement learning solutions, Automatica 50 (12) (2014) 3038–3053, https://doi.org/10.1016/j.automatica.2014.10.047.
[33] A. Mia, B. Fll, C. Kgv, D. Sh, E. Rb, Multi-agent discrete-time graphical games and reinforcement learning solutions, Automatica 50 (12) (2014) 3038–3053, https://doi.org/10.1016/j.automatica.2014.10.047.
[34] B. Dutta, D. Mishra, Minimum cost arborescences, Games Econ. Behav. 74 (1) (2012) 120–143, https://doi.org/10.1016/j.geb.2011.05.007.