Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Predictive pursuit-evasion game control method for approaching space non-cooperative target ⋆

Yuan Chai * **Jianjun Luo** * **Mingming Wang** * **Min Yu** *

* *Research & Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen, 518057, China(e-mail:chaiyuan@mail.nwpu.edu.cn; jjluo@nwpu.edu.cn;mwang@nwpu.edu.cn;yumin@mail.nwpu.edu.cn).*

Abstract: This paper designs the predictive pursuit-evasion game-based orbit control method for the chasing spacecraft to approach a space uncooperative maneuvering target. Firstly, the trajectory planning algorithm combines RRT* and cubic splines to generate a feasible reference trajectory in the relative motion space for the chaser, which enables to deal with the boundary constraints and avoid collisions with the attachments of target. Then, based on the dynamics model, input constrains and objective functions, the pursuit-evasion game model between the target and the chaser is formulated, in which each one acts independently to satisfy its own objective function. With the game formulation, a predictive pursuit-evasion game controller for the chaser is designed to track reference trajectory. Under the frame of model predictive control, the multiple objective constraint optimization can be transferred into to quadratic programming problem to handle input constraints. By predicting opponent's best move and changing its optimal strategy for its benefit iteratively, the saddle points of the game can be obtained without opponent's maneuvering. Numerical simulations verify the effectiveness of the predictive pursuit-evasion game control method for approaching an uncooperative target.

Keywords: pursuit-evasion game, uncooperative target, trajectory planning, orbit dynamics, model predictive control.

1. INTRODUCTION

As one of the key technologies of on-orbit servicing, the autonomous rendezvous to approach the target spacecraft has become an increasing necessity [Croomes (2006)]. However, the rendezvous with non-cooperative maneuvering targets such as a satellite with communication failure has been a difficult problem to be solved so far.

In those existing references [Gao et al. (2019); Kosari et al. (2017)], the chasing spacecraft approaches the uncooperative target by improving the robustness of orbit controller. Nevertheless, the orbit maneuvering of the target is not always in a small bounds. The pursuit-evasion game (PEG) theory is a suitable frame to study the two-opponent-agent optimal decision and control problems, where one player attempts to track down another [Isaacs (1999)]. In this paper, the uncooperative target is seen as a rational player with input constraints to game with the chaser.

The key problem of PEG is to get the saddle point whose definition is if a player unilaterally deviates from this solution, then the player's situation would deteriorate. Usually, the techniques of optimal control are employed. Horie and Conway (2006) formulated a PEG problem involving two different aerial vehicles into a double optimization problem and used a semi-direct method to get the optimal strategies. Based on dynamics programming, Jagat and Sinclair (2017) employed the State-dependent Riccati Equation method (SDRE) to find the saddle point of PEG, whereas the control inputs constraints are not considered. Based on the variational method, PEG leads to a two-point boundary value problem (TPBVP). In [Hafer et al. (2015)], the sensitivity methods are applied to get the global solution fast. Li et al. (2019) presented a dimension-reduction method to solve the high-level TPBVP. However, the solution is influenced by the initial guess. In order to take inputs constraints and avoidance of attachments of target into account, new method is needed to get the saddle point of the game. As a feedback optimal control, MPC has the outstanding ability to deal with constraints [Weiss et al. (2015); Li et al. (2017)]. And the essence of the pursuit-evasion game is also a set of optimal control problem in which players independently optimize their own objective functions. By solving a PEG problem under the frame of MPC, the inputs constraints can be handled convenient. Considering that the control inputs of the target are not available, the target's optimal control strategies is predicted and used to optimize the strategy of the chaser. By repeating this computation several times, the optimized solution for the chaser or the target leads the game to Nash equilibrium. Because of the predictive

⋆ This work was supported by the Science Foundation of Science, Technology and Innovation Commission of Shenzhen Municipality under Grant No. JCYJ20180508151938535, the National Natural Science Foundation of China under Grant No. 61690210, 61690211 and 61603304.

model in MPC and the prediction of the target's optimal strategies, we name it predictive game controller.

However, it appears to be infeasible in scenarios where there are solar panels or other large structure accessories around the target. In this case, trajectory planning method is presented to provide a smooth reference trajectory in the relative motion space for predictive game controller to track. Recently, sampling-based path planning algorithms, such as probabilistic roadmap (PRM), rapid exploring random tree (RRT) and so on, have been widely adopted to plan a path in an unstructured environment [Karaman and Frazzoli (2011)]. Among these sampling-based methods, the efficient RRT* algorithm plans a path with provably asymptotically optimal by adding the 'rewiring' operation. However, the trajectories generated by sampled-based method are often jerky, which are not suitable for spacecraft to track. Current existing smoothing techniques include shortcut methods [Hu et al. (2019)] and optimization-based methods [Zhou et al. (2019)]. Because of the overhead computation of the optimization-based methods, the shortcut methods which replace the jerky portion into curve segments is more feasible. Herein, cubic splines are investigated to smooth the jerky portions of the initial path generated by RRT*.

In this paper, a predictive PEG-based control method including a RRT*- spline trajectory planning and a predictive PEG-based tracking controller is proposed for a chasing spacecraft with higher maneuvering ability to approach an uncooperative maneuvering target. By combining cubic splines into RRT*, the trajectory planning algorithm can generate a feasible trajectory in the relative motion space with boundary constraints. By transforming the multiple-objective constraints optimization problem into quadratic programming (QP) problem under the frame of MPC, the controller can get the saddle points of PEG.

2. PROBLEM STATEMENT

In order to repair an uncooperative maneuvering target in space such as a satellite with communication failures, a chasing spacecraft is driven to approach it. For clarity of description, Fig.1 is used to depict the scene. Moreover, *Earth Centered Inertial frame* $O_I x_I y_I z_I$ and *Orbit frame* $O_t x_t y_t z_t$ are marked in Fig.1.It is assumed that the uncooperative target is regarded as a non-rotating one, which can avoid to deal with attitude control.

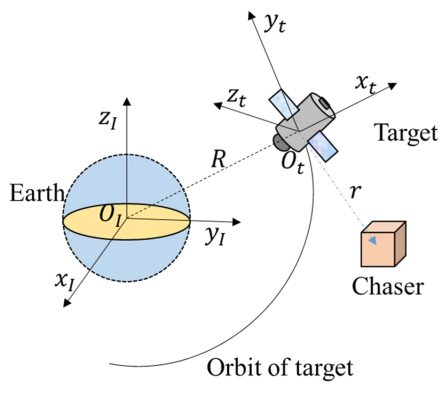


Fig. 1. Approaching an uncooperative target

Define $\boldsymbol{r}_t \in \mathbb{R}^3$ and $\boldsymbol{r}_c \in \mathbb{R}^3$ as the position vector from the center of the earth O_I to the mass center of the target O_t and of the chasing spacecraft O_c, respectively. The orbit motion of the chaser and the target are described as follows, respectively

$$\ddot{\boldsymbol{r}}_c = -\mu \frac{\boldsymbol{r}_c}{||\boldsymbol{r}_c||^3} + \boldsymbol{u}_c, \ddot{\boldsymbol{r}}_t = -\mu \frac{\boldsymbol{r}_t}{||\boldsymbol{r}_t||^3} + \boldsymbol{u}_t \tag{1}$$

where $\mu \in \mathbb{R}$ is gravitational parameter. $\boldsymbol{u}_c \in \mathbb{R}^3$ and $\boldsymbol{u}_t \in \mathbb{R}^3$ are the control acceleration of the chaser and the target, respectively.

Let $\boldsymbol{r} = \boldsymbol{r}_c - \boldsymbol{r}_t \in \mathbb{R}^3$. Mapping it into $O_t x_t y_t z_t$, we can get

$$\begin{aligned}\ddot{\boldsymbol{r}} =& -2\boldsymbol{\omega} \times \dot{\boldsymbol{r}} - \dot{\boldsymbol{\omega}} \times \boldsymbol{r} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) - \mu \frac{\boldsymbol{r}_c}{||\boldsymbol{r}_c||^3} \\ &+ \mu \frac{\boldsymbol{r}_t}{||\boldsymbol{r}_t||^3} + \boldsymbol{u}_c - \boldsymbol{u}_t\end{aligned} \tag{2}$$

where $\boldsymbol{\omega} \in \mathbb{R}^3$ is the orbit angular velocity of the chaser.

In a typical manner, the target is assumed to be in a near-circular orbit. And the distance between the target and the chasing spacecraft is small compared to the orbit radius. Then, the linear model of the relative translational dynamics is derived as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_c \boldsymbol{u}_c + \boldsymbol{B}_t \boldsymbol{u}_t \tag{3}$$

where $\boldsymbol{x} = [\boldsymbol{r}^{\mathrm{T}}, \dot{\boldsymbol{r}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$ is the state vector, and $\boldsymbol{A} = [\boldsymbol{0}_3, \boldsymbol{I}_3; \boldsymbol{A}_{21}, \boldsymbol{A}_{21}]$, $\boldsymbol{A}_{21} = [3n^2, 0, 0; 0, 0, 0; 0, 0, -n^2]$, $\boldsymbol{A}_{22} = [0, 2n, 0; -2n, 0, 0; 0, 0, 0]$, $\boldsymbol{B}_c = [\boldsymbol{0}_3, \boldsymbol{I}_3]^{\mathrm{T}}$, $\boldsymbol{B}_t = -\boldsymbol{B}_c$. $n = \sqrt{\mu/||\boldsymbol{r}_c||^3}$ denotes the orbit rate of the chaser.

The objective of this paper is to design an orbit control frame for the chaser. Considering the limited control ability of spacecraft, the reference trajectory is designed for the chaser to track. The designed trajectory must satisfy the following requirements:
(1) The chaser must has no collision with the attachments of target.
(2) Trajectory for the chaser must consider initial state and final state constraints.
(3) Trajectory has to be smooth enough for spacecraft to track.
The design of the controller also has some constraints:
(1) The chaser have to follow the desired trajectory.
(2) Noncooperative target has unknown maneuvering.
In order to guarantee the tracking performance, we assume that the chasing spacecraft has better maneuvering ability than the uncooperative target.

3. DESIRED TRAJECTORY DESIGN

In this section, the RRT*-spline method is designed to generate a feasible relative path in the relative motion space under kinodynamic constraints.

In step 1, RRT* algorithm is employed to get an initial path. Considering the maneuvering of target, path planning in the inertial space has to be repeated at each instant, which is time-consuming. Noticing that the initial relative position and the final desired relative position are constant vectors, path planning in the relative motion space can be executed only at the beginning of the mission. To seek a path from the initial relative position $\boldsymbol{r}(t_0)$ to the final desired relative position $\boldsymbol{r}(t_f)$ and avoid collisions with attachments of target, the RRT* algorithm is used to create a tree of paths in the relative motion space and get a collision-free path [Karaman and Frazzoli (2011)]. Consequently, the generated path is segmented into series local linear paths between $m+1$ intermediate waypoints including initial and terminal points. However, the chaser is expected to follow a smooth trajectory with velocity and acceleration constraints.

In step 2, the cubic spline method is used to guarantee the continuity of the curvature between these waypoints. In order to avoid collisions and smooth the trajectory, the following procedures are proposed to choose as few interpolation points as possible from $m+1$ waypoints of RRT* and generate a smooth obstacle-free trajectory.
(1)After numbering all waypoints in order as $[0, 1, \cdots, m]$, choose the waypoints of number $[0, m/3, 2m/3, m]$ as the interpolation points for cubic spline based on the triple-section method.
(2)Considering kinodynamics constraints, generate the path by three-dimension cubic-spline interpolation methods between interpolation points [Yang et al. (2014)].
(3)Check if these local path $\boldsymbol{r}_{des} = \{\boldsymbol{r}_{des,0-m/3}, \boldsymbol{r}_{des,m/3-2m/3}, \boldsymbol{r}_{des,2m/3-m}\}$ are obstacle free. Once there is points of the local path between two chosen waypoints being inside the defined area of obstacles, the new interpolation points are chosen from waypoints to generate new local path until a collision-free local path between these two chosen waypoints is got.
Repeating these steps, the smooth and obstacle-free trajectory $\boldsymbol{r}_{des}(t) : \mathbb{R} \to \mathbb{R}^n$ is obtained as a piecewise-cubic function that is the concatenation of the different cubic trajectories. By differentiating the cubic function $\boldsymbol{r}_{des}(t)$, we can also get the desired velocity $\dot{\boldsymbol{r}}_{des}(t) : \mathbb{R} \to \mathbb{R}^n$. Therefore, the desired state is written as $\boldsymbol{x}_{des} = [\boldsymbol{r}_{des}^{\mathrm{T}}, \dot{\boldsymbol{r}}_{des}^{\mathrm{T}}]^{\mathrm{T}}$.

Remark 1. The trajectory planning method proposed in this paper can be extended in other applications.

4. PREDICTIVE GAME FORMULATION

4.1 Game formulation in continuous horizon

In the two-player PEG, the chasing spacecraft performs as the pursuer and the noncooperative target is the evader. The objective of the chaser is to minimize the tracking error between relative state and the predefined reference trajectories using minimal fuel. The target has the conflict objective to maximize the relative state and minimize the fuel. Therefore, define the objective function of chaser and the target as

$$J_c(\boldsymbol{u}_c, \boldsymbol{u}_t) = \frac{1}{2}\int_0^{\infty} (\boldsymbol{e}^{\mathrm{T}}\boldsymbol{Q}_c\boldsymbol{e} + \boldsymbol{u}_c^{\mathrm{T}}\boldsymbol{R}_c\boldsymbol{u}_c)dt \tag{4}$$

$$J_t(\boldsymbol{u}_c, \boldsymbol{u}_t) = \frac{1}{2}\int_0^{\infty} (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_t\boldsymbol{x} - \boldsymbol{u}_t^{\mathrm{T}}\boldsymbol{R}_t\boldsymbol{u}_t)dt \tag{5}$$

where $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_{des}$ means the trajectory tracking error of the chaser. The weighting matrices $\boldsymbol{Q}_c \in \mathbb{R}^{6\times 6}$,$\boldsymbol{Q}_t \in \mathbb{R}^{6\times 6}$,$\boldsymbol{R}_c \in \mathbb{R}^{3\times 3}$,$\boldsymbol{R}_t \in \mathbb{R}^{3\times 3}$ impose the specifications in relative states and inputs, which are all positive-definite and symmetry matrices.

According to the requirements of the controller, PEG between the chaser and the target is formulated as following multiple-objective optimization with the dynamics constraints and inputs constraints:
Problem 1:

$$\begin{aligned} &\min_{\boldsymbol{u}_c}\max_{\boldsymbol{u}_t} J(\boldsymbol{u}_c, \boldsymbol{u}_t) = \min_{\boldsymbol{u}_c} J_c \vee \max_{\boldsymbol{u}_t} J_t \\ s.t.\,&\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_c\boldsymbol{u}_c + \boldsymbol{B}_t\boldsymbol{u}_t \\ &||\boldsymbol{u}_c||_\infty \le u_{c,max}, ||\boldsymbol{u}_t||_\infty \le u_{t,max} \end{aligned} \tag{6}$$

where $u_{t,max}$ and $u_{c,max}$ are maximum control acceleration amplitude of the target and the chaser, respectively.

The solution of the above optimization problem $\boldsymbol{u}_c^*, \boldsymbol{u}_t^*$ is called the saddle point and satisfies the following equation:

$$J(\boldsymbol{u}_c^*, \boldsymbol{u}_t) \le J(\boldsymbol{u}_c^*, \boldsymbol{u}_t^*) \le J(\boldsymbol{u}_c, \boldsymbol{u}_t^*) \tag{7}$$

However, it is challenging to directly solve Problem 1 which is multi-objective constrained optimization. Moreover, the spacecraft PEG problem is not always a zero-sum game, but a constant-sum game [Tzannetos et al. (2016)].

4.2 Predictive PEG problem

MPC is a feedback control frame by solving an optimal control problem repeat, which have great performance in dealing with constraints and disturbance for its three principles, including predictive model, receding horizon optimization and feedback correction. Based on the frame of MPC, PEG is reformulated in the receding horizon to handle constraints easier. The game model in receding horizon is composed of a predictive model, input constraints and objective functions, which will be defined below to formulate the predictive PEG model.

Denote the starting and ending time of the rendezvous process as t_0 and t_f. The total time interval is divided by sampling period Δt into N_s subintervals evenly, such that $[t_0, t_1, \cdots, t_k, \cdots, t_f]$. At time instant t_k, Equation(2) is discretized with Δt as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\boldsymbol{x}_k + \boldsymbol{B}_{c,d}\boldsymbol{u}_{c,k} + \boldsymbol{B}_{t,d}\boldsymbol{u}_{t,k} \tag{8}$$

where $\boldsymbol{x}_k = \boldsymbol{x}(t_k)$, $\boldsymbol{u}_{c,k} = \boldsymbol{u}_c(t_k)$, $\boldsymbol{u}_{t,k} = \boldsymbol{u}_t(t_k)$, and $\boldsymbol{A}_d = e^{\boldsymbol{A}\Delta t} \in \mathbb{R}^{6\times 6}$, $\boldsymbol{B}_{c,d} = \int_0^{\Delta t} e^{\boldsymbol{A}\tau}\boldsymbol{B}_c d\tau \in \mathbb{R}^{6\times 3}$, $\boldsymbol{B}_{t,d} = \int_0^{\Delta t} e^{\boldsymbol{A}\tau}\boldsymbol{B}_t d\tau \in \mathbb{R}^{6\times 3}$.

Considering a fixed receding horizon N, $\boldsymbol{X}_k$,$\boldsymbol{U}_{c,k}$,$\boldsymbol{U}_{t,k}$ are defined as the stack of relative state vector, control inputs of the chaser the control inputs of the target, respectively

$$\begin{aligned} \boldsymbol{X}_k &= [\boldsymbol{x}_{k+1}^{\mathrm{T}}, \boldsymbol{x}_{k+2}^{\mathrm{T}}, \cdots, \boldsymbol{x}_{k+N}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{6N} \\ \boldsymbol{U}_{c,k} &= [\boldsymbol{u}_{c,k}^{\mathrm{T}}, \boldsymbol{u}_{c,k+1}^{\mathrm{T}}, \cdots, \boldsymbol{u}_{c,k+N-1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{3N} \\ \boldsymbol{U}_{t,k} &= [\boldsymbol{u}_{t,k}^{\mathrm{T}}, \boldsymbol{u}_{t,k+1}^{\mathrm{T}}, \cdots, \boldsymbol{u}_{t,k+N-1}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{3N} \end{aligned} \tag{9}$$

Based on (9), the evolution of the relative motion dynamics over the receding horizon $[t_k, t_{k+N-1}]$ can be formulated as

$$\boldsymbol{X}_k = \boldsymbol{\Lambda}\boldsymbol{x}_k + \boldsymbol{\Xi}_c\boldsymbol{U}_{c,k} + \boldsymbol{\Xi}_t\boldsymbol{U}_{t,k} \tag{10}$$

where $\boldsymbol{x}_k$ is the state vector sampled at the instant t_k. The matrices in (10) are

$$\boldsymbol{\Lambda} = \left[\, \boldsymbol{A}_d^{\mathrm{T}} \;\; \boldsymbol{A}_d^{2\,\mathrm{T}} \;\; \cdots \;\; \boldsymbol{A}_d^{N\,\mathrm{T}} \,\right]^{\mathrm{T}} \in \mathbb{R}^{6N\times 6}$$

$$\Xi_c = \begin{bmatrix} \boldsymbol{B}_{c,d} & 0 & 0 & 0 \\ \boldsymbol{A}_d\boldsymbol{B}_{c,d} & \boldsymbol{B}_{c,d} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ \boldsymbol{A}_d^{N-1}\boldsymbol{B}_{c,d} & \boldsymbol{A}_d^{N-2}\boldsymbol{B}_{c,d} & \cdots & \boldsymbol{B}_{c,d} \end{bmatrix} \in \mathbb{R}^{6N\times 3N}$$

$$\Xi_t = \begin{bmatrix} \boldsymbol{B}_{t,d} & 0 & 0 & 0 \\ \boldsymbol{A}_d\boldsymbol{B}_{t,d} & \boldsymbol{B}_{t,d} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ \boldsymbol{A}_d^{N-1}\boldsymbol{B}_{t,d} & \boldsymbol{A}_d^{N-2}\boldsymbol{B}_{t,d} & \cdots & \boldsymbol{B}_{t,d} \end{bmatrix} \in \mathbb{R}^{6N\times 3N}$$

Based on the reference relative trajectory of the chaser, the tracking error dynamics for chaser can be obtained as

$$\boldsymbol{E}_k = \boldsymbol{\Lambda}\boldsymbol{x}_k + \Xi_c\boldsymbol{U}_{c,k} + \Xi_t\boldsymbol{U}_{t,k} - \boldsymbol{X}_{des,k} \tag{11}$$

where $\boldsymbol{E}_k$ and $\boldsymbol{X}_{des,k}$ is the stack of the trajectory tracking error and the relative state,respectively. $\boldsymbol{E}_k$ and $\boldsymbol{X}_{d,k}$ have following form

$$\begin{aligned} \boldsymbol{E}_k &= [\boldsymbol{e}_{k+1}^{\mathrm{T}}, \boldsymbol{e}_{k+2}^{\mathrm{T}}, \cdots, \boldsymbol{e}_{k+N}^{\mathrm{T}}] \in \mathbb{R}^{6N} \\ \boldsymbol{X}_{des,k} &= [\boldsymbol{x}_{des,k+1}^{\mathrm{T}}, \boldsymbol{x}_{des,k+2}^{\mathrm{T}}, \cdots, \boldsymbol{x}_{des,k+N}^{\mathrm{T}}] \in \mathbb{R}^{6N} \end{aligned} \tag{12}$$

where $\boldsymbol{x}_{des,k} \in \mathbb{R}^3$ is the desired relative state at the sampling time t_k and $\boldsymbol{e}_k = \boldsymbol{x}_k - \boldsymbol{x}_{des,k}$.

Define the objective of the spacecraft PEG in the receding horizon as

$$J_k(\boldsymbol{X}_k, \boldsymbol{U}_{c,k}, \boldsymbol{U}_{t,k}) = [J_{c,k}, J_{t,k}] \tag{13}$$

where $J_{c,k}$,$J_{t,k}$ is the objective of the chaser and the target in the receding horizon, respectively

$$\begin{aligned} J_c(\boldsymbol{X}_k, \boldsymbol{U}_{c,k}, \boldsymbol{U}_{t,k}) &= \boldsymbol{E}_k^{\mathrm{T}}\boldsymbol{Q}_{c,k}\boldsymbol{E}_k + \boldsymbol{U}_{c,k}^{\mathrm{T}}\boldsymbol{R}_{c,k}\boldsymbol{U}_{c,k} \\ J_t(\boldsymbol{X}_k, \boldsymbol{U}_{c,k}, \boldsymbol{U}_{t,k}) &= \boldsymbol{X}_k^{\mathrm{T}}\boldsymbol{Q}_{t,k}\boldsymbol{X}_k - \boldsymbol{U}_{t,k}^{\mathrm{T}}\boldsymbol{R}_{t,k}\boldsymbol{U}_{t,k} \end{aligned} \tag{14}$$

where $\boldsymbol{Q}_{c,k} = \boldsymbol{I}_N \otimes \boldsymbol{Q}_c \in \mathbb{R}^{6N\times 6N}$, $\boldsymbol{R}_{c,k} = \boldsymbol{I}_N \otimes \boldsymbol{R}_c \in \mathbb{R}^{3N\times 3N}$, $\boldsymbol{Q}_{t,k} = \boldsymbol{I}_N \otimes \boldsymbol{Q}_t \in \mathbb{R}^{6N\times 6N}$, $\boldsymbol{R}_{t,k} = \boldsymbol{I}_N \otimes \boldsymbol{R}_t \in \mathbb{R}^{3N\times 3N}$.

Therefore, the multiple-objective constraints optimization Problem 1 is formulated under the frame of MPC, which is as follows:
Problem 2:

$$\begin{aligned} &\min_{\boldsymbol{U}_{c,k}} \max_{\boldsymbol{U}_{t,k}} J(\boldsymbol{X}_k, \boldsymbol{U}_{c,k}, \boldsymbol{U}_{t,k}) \\ s.t.&\boldsymbol{X}_k = \boldsymbol{\Lambda}\boldsymbol{x}_k + \Xi_c\boldsymbol{U}_{c,k} + \Xi_t\boldsymbol{U}_{t,k} \\ &- u_{c,max}\mathbf{1}_{3N} \leq \boldsymbol{U}_{c,k} \leq u_{c,max}\mathbf{1}_{3N} \\ &- u_{t,max}\mathbf{1}_{3N} \leq \boldsymbol{U}_{t,k} \leq u_{t,max}\mathbf{1}_{3N} \end{aligned} \tag{15}$$

At each sampling moment t_k, Problem 2 should be solved to get the saddle point for the chaser. However, because of the non-cooperative character of the game, Problem 2 cannot be solved by the chaser or the target.

5. PREDICTIVE PEG-BASED CONTROLLER

In this section, considering that the antagonistic character of the game, the transformed model and algorithm is proposed to predict the maneuver of the target and get the best control of the chasing spacecraft.

5.1 Individual optimization formulation

According to the definition of the saddle point, the optimization objective function in (15) is equal to the following function

$$\begin{aligned} &J_k(\boldsymbol{U}_{c,k}^*, \boldsymbol{U}_{t,k}^*) \\ &= \min_{\boldsymbol{U}_{c,k}} J_{c,k}(\boldsymbol{U}_{c,k}, \boldsymbol{U}_{t,k}^*) = \max_{\boldsymbol{U}_{t,k}} J_{t,k}(\boldsymbol{U}_{c,k}^*, \boldsymbol{U}_{t,k}) \\ &= \min_{\boldsymbol{U}_{c,k}} \max_{\boldsymbol{U}_{t,k}} J(\boldsymbol{U}_{c,k}, \boldsymbol{U}_{t,k}) = \max_{\boldsymbol{U}_{t,k}} \min_{\boldsymbol{U}_{c,k}} J(\boldsymbol{u}_{c,k}, \boldsymbol{U}_{t,k}) \end{aligned} \tag{16}$$

For the chaser which can only optimize its own control policy, the objective function in the optimization problem (15) can be transferred into

$$\begin{aligned} J_k &= J_{c,k}(\boldsymbol{X}_k, \boldsymbol{U}_{c,k}, \boldsymbol{U}_{t,k}^*) \\ &= const + \frac{1}{2}\boldsymbol{U}_{c,k}^{\mathrm{T}}\boldsymbol{M}_{c,2}\boldsymbol{U}_{c,k} + \boldsymbol{M}_{c,2}^{\mathrm{T}}\boldsymbol{U}_{c,k} \end{aligned} \tag{17}$$

where $\boldsymbol{M}_{c,1} = \Xi_c^{\mathrm{T}}\boldsymbol{Q}_{c,k}(\boldsymbol{\Lambda}\boldsymbol{x}_k + \Xi_t\boldsymbol{U}_{t,k}^* - \boldsymbol{X}_{d,k}) \in \mathbb{R}^{3N}$, $\boldsymbol{M}_{c,2} = \Xi_c^{\mathrm{T}}\boldsymbol{Q}_k\Xi_c + \boldsymbol{R}_{c,k} \in \mathbb{R}^{3N\times 3N}$.

Thus, the optimization problem for chaser is transformed from Problem 3 into a QP problem:
Problem 3:

$$\begin{aligned} &\min_{\boldsymbol{U}_{c,k}} J_{c,k}(\boldsymbol{X}_k, \boldsymbol{U}_{c,k}, \boldsymbol{U}_{t,k}^*) \\ s.t.&\boldsymbol{X}_k = \boldsymbol{\Lambda}\boldsymbol{x}_k + \Xi_c\boldsymbol{U}_{c,k} + \Xi_t\boldsymbol{U}_{t,k} \\ &- u_{c,max}\mathbf{1}_{3N} \leq \boldsymbol{U}_{c,k} \leq u_{c,max}\mathbf{1}_{3N} \end{aligned} \tag{18}$$

where $J_{c,k} = \frac{1}{2}\boldsymbol{U}_{c,k}^{\mathrm{T}}\boldsymbol{M}_{c,2}\boldsymbol{U}_{c,k} + \boldsymbol{M}_{c,2}^{\mathrm{T}}\boldsymbol{U}_{c,k}$. If the best inputs of the target $\boldsymbol{U}_{t,k}^*$ is known, we can get the saddle point for the chaser.

Similarly, the optimization problem for target can be transformed as:
Problem 4:

$$\begin{aligned} &\max_{\boldsymbol{U}_{t,k}} J_{t,k}(\boldsymbol{X}_k, \boldsymbol{U}_{c,k}^*, \boldsymbol{U}_{t,k}) \\ s.t.&\boldsymbol{X}_k = \boldsymbol{\Lambda}\boldsymbol{x}_k + \Xi_c\boldsymbol{U}_{c,k} + \Xi_t\boldsymbol{U}_{t,k} \\ &- u_{t,max}\mathbf{1}_{3N} \leq \boldsymbol{U}_{t,k} \leq u_{t,max}\mathbf{1}_{3N} \end{aligned} \tag{19}$$

$J_{t,k} = \frac{1}{2}\boldsymbol{U}_{t,k}^{\mathrm{T}}\boldsymbol{M}_{t,2}\boldsymbol{U}_{t,k} + \boldsymbol{M}_{t,1}^{\mathrm{T}}\boldsymbol{U}_{t,k}$, $\boldsymbol{M}_{t,1} = \Xi_t^{\mathrm{T}}\boldsymbol{Q}_{t,k}(\boldsymbol{\Lambda}\boldsymbol{x}_k + \Xi_c\boldsymbol{U}_{c,k}^*) \in \mathbb{R}^{3N}$, $\boldsymbol{M}_{t,2} = \Xi_t^{\mathrm{T}}\boldsymbol{Q}_{t,k}\Xi_t - \boldsymbol{R}_{t,k} \in \mathbb{R}^{3N\times 3N}$. If the best inputs of the chaser $\boldsymbol{U}_{c,k}^*$ is known, the target can get control strategies corresponding to the saddle point.

5.2 Predictive PEG controller

For the chaser, the best inputs of the target $\boldsymbol{U}_{t,k}^*$ in the Problem 3 must be known. Considering the antagonistic character of the game, the chaser has to predict the optimal inputs $\boldsymbol{U}_{t,k}^*$ of the target through optimization problem of the target (19). In such a way, the chaser predicts the target's best response, which is its worst-case scenario, and after that the chaser determines its actual move. By repeating this procedure, the level of thinking is added before deciding its final input trajectory. For clear understanding, the proposed controller is illustrated in the block of Predictive game controller for chaser of Fig 2, while the target uses similar procedure to get its control strategies.

Remark 2. Because the objective function of the target 5 is not a positive quadratic function, choosing a suitable $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$ is important to guarantee that Problem 4 is a convex QP problem. Otherwise, the optimization would fall into local optimal results or could not give any feasible control solution.

Fig.2 presents the implementation scheme of the proposed approaching method. Firstly, the RRT*-spline algorithm provide the feasible path in the relative motion space by computing off-line at the first several seconds of the approaching mission. Then, the chaser tracks the designed trajectory by the proposed predictive PEG controller. Due to the simple QP formulation of (18) and (19), the proposed controller can be time efficient with proper thinking level.

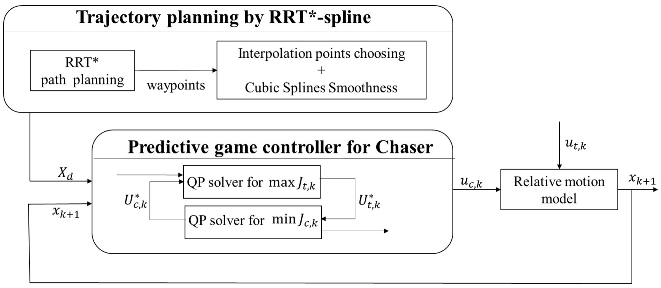


Fig. 2. Block diagram of predictive game control method

6. NUMERICAL SIMULATIONS

This section presents a numerical example of approaching an uncooperative target. In our implementation, the initial relative state between the chasing spacecraft and the target is $\boldsymbol{r}_0 = [300, 150, -100, 0, 0, 0]^{\mathrm{T}}$m. The distance between the target and the earth's center is $r_c = 6.72 \times 10^6$m. The model is discretized with $\Delta t = 1$s.

In the first part, we use the RRT*-spline method to generate the reference trajectory in the relative motion space, which will be an input of the predictive game-based controller. The initial point in the planning is $[300, 150, -100]^{\mathrm{T}}$. Due to the approaching mission, the terminal point is $[0, 0, 0]^{\mathrm{T}}$. Set the sampled point in the RRT* is 500, the sampling step is 0.5m. The initial relative velocity $\dot{\boldsymbol{r}}_{des,0} = [0, 0, 0]^{\mathrm{T}}$ and the terminal relative velocity $\dot{\boldsymbol{r}}_{des,m} = [0, 0, 0]^{\mathrm{T}}$ is the clamped boundary conditions. The planning time is 120s.

Under these parameters, the generated procedure is shown in Fig.3. Two spheres denote as the attachments of the target. At first, the random position point is sampling from task space, which is denoted as $\times$ with blue color. The green solid line denotes the tree whose vertex is chosen from random position point. Then, the red solid line which is extremely rough is the path by RRT*. After that, the trajectory smoothed by cubic spline is the blue line with line type of *. The reference relative position and velocity in three axis are cubic spline line as in fig 4(a) and fig 4(c). It is obvious that the boundary constraints is satisfied.

For PEG, the controller of the chaser is used to generate the control acceleration at each sampling instant. It is assumed that the uncooperative target is rational, which means that it will also use the strategy computed by the presented controller in this paper. The control acceleration inputs magnitude constraints as $u_{c,max} = 5$N/kg and $u_{t,max} = 2$N/kg. It is assumed that the weight matrices for target are $\boldsymbol{Q}_t = 10^{-5} \times diag(1, 1, 1, 1, 1, 1)$,

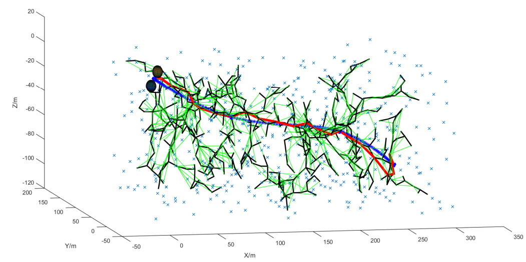
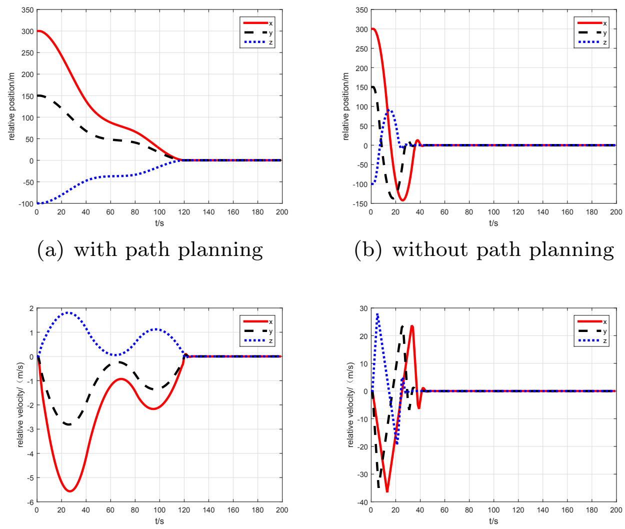


Fig. 3. Trajectory planning via $RRT^* - Spline$

$\boldsymbol{R}_t = 5 \times diag(1, 1, 1)$. Correspondingly, the wight matrices of the chasing spacecraft are chosen as $\boldsymbol{Q}_t = 1 \times diag(1, 1, 1, 1, 1, 1)$ and $\boldsymbol{R}_r = 1 \times diag(1, 1, 1)$. The receding horizon is N = 10. The thinking level are $l_c = 2; l_t = 2;$

With these parameters, the following simulation results illustrate the performance of the predictive PEG-based controller. In order to show the importance of the path planning, the simulations compare the result of controller with reference trajectory and the results without path planning. Fig.4 is the relative position and relative velocity between the chaser and the target using two controllers, respectively. At the end of the simulation, the relative position and velocity come to zero, which indicates that PEG-based controller is valid. Apparently, the curves of the controller without path planning are not smooth and consistently shocked, which is not the desired performance. Instead, the curves of the controller with smooth reference trajectory can approach target smoothly.



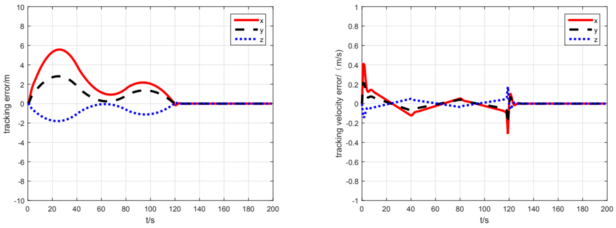
(a) with path planning (b) without path planning

(c) with path planning (d) without path planning

Fig. 4. relative position and velocity

The tracking error relative position and error relative velocity using the proposed controller is shown in the Fig.5. It can be seen that the reference relative trajectory can be tracked by the chaser using the proposed controller.

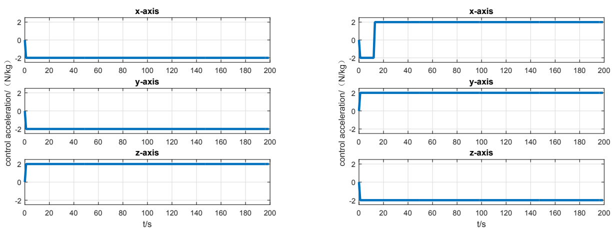
Fig.6 shows three-axis control acceleration of the target. We can deduce that the target tries its best to escape. Fig.7 are three-axis control acceleration of the chaser using two kinds of controllers. It can be seen that all the control acceleration are within the permissible range, which ver-



(a) tracking position error (b) tracking velocity error

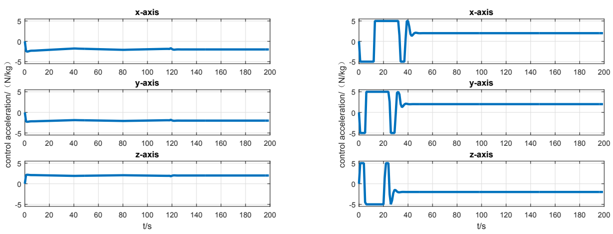
Fig. 5. tracking error

ifies the advantage of dealing with inputs constraints. In order to compare the fuel consumption, the cost function is designed as $L = \int_0^\infty ||\boldsymbol{u}_c||_1 dt$. The cost value using game controller with path planning is $L = 1194$. The cost value using game controller without path planning is $L = 1469$. Therefore, the controller with path planning is much suitable for approaching the uncooperative target.



(a) with path planning (b) without path planning

Fig. 6. control acceleration of the target



(a) with path planning (b) without path planning

Fig. 7. control acceleration of the chaser

7. CONCLUSIONS

This paper has proposed a predictive PEG control for the chasing spacecraft to approach an noncooperative target who has limited maneuver ability. By formulating the approaching problem into the game problem, the maneuvering of the target can be considered completely. In the frame of MPC, PEG-based controller have advantages in dealing with constraints. With the design of RRT*-spline planning, the smooth and feasible trajectory in the relative motion space is generated for chaser to track. Under unknown maneuvering of target, the proposed minimax solution can lead to a Nash equilibrium. The simulation results show the effectiveness of the proposed controller because of its prediction of the target maneuvering. The future work should focus on the trajectory planning and replanning with obstacles in the relative motion space.

REFERENCES

Croomes, S. (2006). Overview of the DART mishap investigation results. *NASA Report*, 1–10.

Gao, D., Luo, J., Ma, W., and Englot, B.J. (2019). Parameterized nonlinear suboptimal control for tracking and rendezvous with a non-cooperative target. *Aerospace Science and Technology*, 87, 15–24.

Hafer, W.T., Reed, H.L., Turner, J.D., and Pham, K. (2015). Sensitivity methods applied to orbital pursuit-evasion. *Journal of Guidance Control and Dynamics*, 38(6), 1118–1126.

Horie, K. and Conway, B.A. (2006). Optimal fighter pursuit-evasion maneuvers found via two-sided optimization. *Journal of Guidance Control and Dynamics*, 29(1), 105–112.

Hu, J., Ma, Z., Niu, Y., Tian, W., and Yao, W. (2019). Real-time trajectory replanning for quadrotor using octomap and uniform b-splines. 727–741.

Isaacs, R. (1999). *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization.* Courier Corporation.

Jagat, A. and Sinclair, A.J. (2017). Nonlinear control for spacecraft pursuit-evasion game using the state-dependent Riccati equation method. *IEEE Transactions on Aerospace and Electronic Systems*, 53(6), 3032–3042.

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7), 846–894.

Kosari, A., Jahanshahi, H., and Razavi, S.A. (2017). An optimal fuzzy PID control approach for docking maneuver of two spacecraft: Orientational motion. *Engineering Science and Technology, an International Journal*, 20(1), 293–309.

Li, Q., Yuan, J., Zhang, B., and Gao, C. (2017). Model predictive control for autonomous rendezvous and docking with a tumbling target. *Aerospace Science and Technology*, 69, 700–711.

Li, Z.y., Zhu, H., Yang, Z., and Luo, Y.z. (2019). A dimension-reduction solution of free-time differential games for spacecraft pursuit-evasion. *Acta Astronautica*.

Tzannetos, G., Marantos, P., and Kyriakopoulos, K.J. (2016). A competitive differential game between an unmanned aerial and a ground vehicle using model predictive control. 1053–1058.

Weiss, A., Baldwin, M., Erwin, R.S., and Kolmanovsky, I.V. (2015). Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies. *IEEE Transactions on Control Systems and Technology*, 23(4), 1638–1647.

Yang, K., Moon, S., Yoo, S., Kang, J., Doh, N.L., Kim, H.B., and Joo, S. (2014). Spline-based RRT path planner for non-holonomic robots. *Journal of Intelligent & Robotic Systems*, 73(1-4), 763–782.

Zhou, D., Yu, Z., Zhang, Y., and Li, S. (2019). Translational and rotational motion planning for spacecraft close proximity using sampling-based methods. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(10), 3680–3699.