Available online at www.sciencedirect.com
ScienceDirect
IFAC Papers Online
CONFERENCE PAPER ARCHIVE
IFAC-PapersOnLine 49-17 (2016) 391–396

Fault-Tolerant Low-Thrust Trajectory Design with Backups for Multiple Targets

Akifumi Wachi * Naoya Ozaki * Shinichi Nakasuka *

* *The University of Tokyo*
(e-mail: {wachi,ozaki,nakasuka}@space.t.u-tokyo.ac.jp).

Abstract: This paper presents a method to design robust trajectories against the possible faults of low-thrust engines. Recently, increasing number of space probes with low-thrust engines have been developed. Due to the insufficient reliability of low-thrust propulsion system, almost all probes with ion engines have experienced the failure of engines. Conventional methods to design the low-thrust trajectory have pursued fuel minimum solution to only one target celestial body, which does not necessarily mean the maximization of mission achievement. In other words, it can enhance the mission accomplishment to intelligently change the trajectory when engine failures occur. In this research, the objective function is defined as *expected scientific gain*, and the optimal solution is searched by a proposed method. Resulting method, Bellman Rapidly-exploring Random Trees (Bellman RRT) are efficiently extended by applying the Bellman equation. Finally, the numerical simulation demonstrated that the Bellman RRT improved the mission achievement using real deep space exploration scenario.

Keywords: Trajectory Planning, Fault Tolerance, Backup, Dynamic Programming

1. INTRODUCTION

Low-thrust propulsion systems have received considerable attention for space missions since we can efficiently achieve large delta-V by adopting them. Because of low reliability of low-thrust engines, most of the space probes with the engines have experienced engine failures. Based on this background, this research aims to realize fault-tolerance by intelligently designing a trajectory. In order to accomplish this purpose, we consider the probability of the engine permanent failure, and we introduce the flexibility that backup trajectories can be chosen as additional decision variables. This paper presents the robust trajectory design method by choosing *backup trajectories* when *permanent failure* of low-thrust engines occurs.

As to space engineering which is highly conservative to improve reliability, previous works have maintained the robustness by building redundant systems. In addition to building such systems, some of the previous works ensure higher robustness by quantitatively evaluating the probability of failure of components. Past work on Model-based programming by Williams et al. (2003) has presented the method to build embedded systems robust to the risks or extreme uncertainty. Our research is based on the similar concept in the sense that we also focus on the failure possibility. Even though their method has maintained the robustness in terms of system design, our method maintains it in terms of trajectory design. Recently, there has been an increasing number of work on the typical problems of the low-thrust trajectory design. The issues especially addressed in recent years are possible under performance of the engines or missed thrust (Olympio (2010), Rayman et al. (2007)). A great deal of work has addressed the issue of temporal failure such as missed thrust, but permanent failure of low-thrust propulsion systems also should be considered. Most of the trajectory design method including low-thrust trajectory design has solved optimization problem defining the objective function as fuel consumption or orbit transfer time (Betts (1998), Sims and Flanagan (1999)). Also, previous work by Coverstone-Carroll et al. (2000) or Lee et al. (2005) solved the multi-objective optimization problem, focusing on both of the final spacecraft mass and time-of-flight. However, it is not essential objective for space missions to minimize fuel consumption or orbit transfer time. On the other hand, especially in the field of robotics, the objective function directly expressing the mission accomplishment is defined and optimized. Previous research by Beard et al. (2002) defined the cost function coming from a weighed sum of threat and path length. Alterovitz et al. (2007) has proposed Stochastic Motion Roadmap (SMRM) algorithm to generate the path maximizing the probability of avoiding collisions and successfully reaching the goal.

Most of the previous work takes into account only short-term failure, missed thrust of low-thrust propulsion systems. On the other hand, this research considers permanent failures of electrically powered spacecraft propulsion system. In order to maintain the robustness against the failure of low thrust engines, we define the objective function as the *expected scientific gain*, and directly optimize the essential objective for space missions. This objective function can include the idea of *backup trajectories*. Fig. 1 describes the concept of this study. This figure is the real trajectory on B plane of the deep space probe *PROCYON* (Funase et al. (2014)). Conventional methods (green dashed line) have pursued fuel minimum solution to the primary candidate. However, proposed trajectory (magenta) are more likely to enable the space probe to

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.09.067

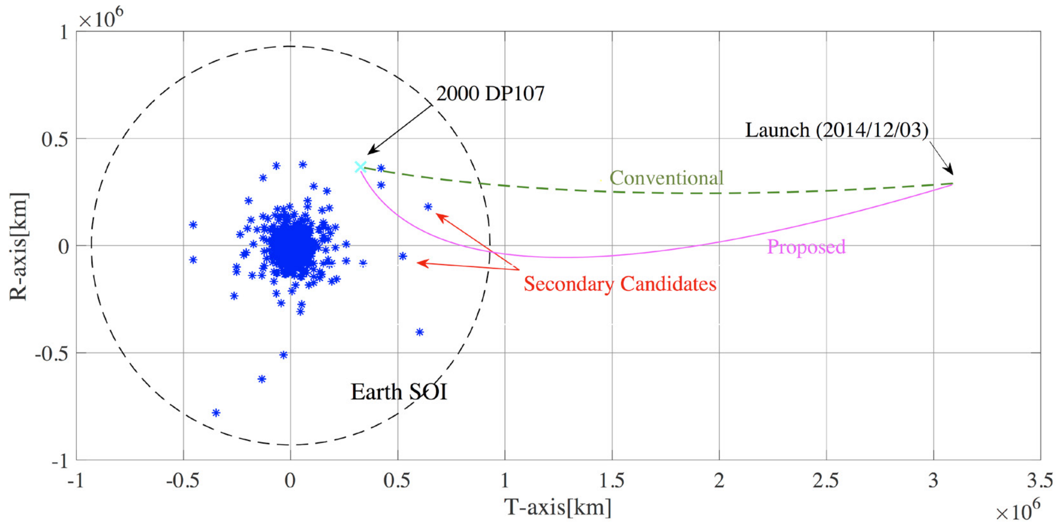

Fig. 1. Concept of this study. This figure is the real trajectory on B plane. B plane indicates the positions of space probe or celestial bodies when space probe is proximate to the origin. The concept of B plane is significant because the trajectory after gravity assist is changed according to the position with respect to the origin. Blue dots denote the position on B-plane to flyby to the celestial objects. Black dashed circle means the sphere of influence of the Earth. The space probe will pass through the position "Launch" if it does not use thrusters at all. When the space probe arrive at the point "2000DP107", it can explore the asteroid.

accomplish the worthwhile missions by change the target celestial object even if engine failure occurs.

In this paper, the object function is defined as expected worth of mission accomplishment. Optimal trajectory is searched by our proposed method, a novel sampling method which gains efficiency by applying the Bellman equation. Finally, the numerical simulation demonstrated that Bellman RRT improved the mission achievement.

2. PROBLEM STATEMENT

This paper considers the problem of finite-horizon optimal control of dynamic systems, with state and control constraints. We assume a discrete-time, continuous-state dynamics model.

Optimal control policy is always changeable according to the state. The planning problem is discretized in time. The state vector consists of the position and velocity written as

$$\boldsymbol{x}_k = [\ \boldsymbol{r}_k^\top,\ \boldsymbol{v}_k^\top\ ]^\top, \tag{1}$$

where $\boldsymbol{r} \in \mathbb{R}^3$ is the position vector and $\boldsymbol{v} \in \mathbb{R}^3$ is the velocity vector. The subscript k $(= 1, 2, \cdots, N-1)$ indicates the decision stage (i.e. time). The state of the system $\boldsymbol{x}_k$ is fully observable at all times. Let $\boldsymbol{u}_k \in \mathbb{R}^3$ denote the control input, which means the velocity increment. Consider the time-variant system and next state vector $\boldsymbol{x}_{k+1}$ is given by the following equation.

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \quad \forall k \in [1, N-1], \tag{2}$$

where $f : [1\ N-1] \times \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$ is a given function.

The objective function to maximize in this problem is dependent on the number of normal engines as well as the state vector. When a engine is not failed or broken, it is called *normal engine* hereafter. Let $n_k \in \mathbb{Z}$ denote the number of normal engines, which is also fully observable at all times. The number of engines is assumed to be a positive integer. In other words, indecisive or temporary failures are not considered in this paper. No engine is assumed to fail at stage 1, and then n_1 means the number of mounted engines $n_{\max} \in \mathbb{Z}$. The objective function to maximize in this research is defined as

$$J_k(\boldsymbol{x}_k, n_k, \boldsymbol{u}_{k:N-1}) := \sum_{j=1}^{m} w_j \Pr\Big[\ \boldsymbol{r}_N = \boldsymbol{r}^j\ \Big], \tag{3}$$

where $\boldsymbol{u}_{k:N-1}$ is a sequence of control and defined as $\boldsymbol{u}_{k:N-1} = \{\boldsymbol{u}_k, \cdots \boldsymbol{u}_{N-1}\}$. $w_j \in \mathbb{R}$ is the exploration worth of j-th candidate. $\boldsymbol{r}^j \in \mathbb{R}^3$ is the position of j-th candidate at final time. $\Pr[\boldsymbol{r}_N = \boldsymbol{r}^j]$ is the possibility that the space probe can arrive at j-th candidate. $m \in \mathbb{Z}$ is the number of candidates to consider in this problem. We call the first candidate (i.e. $w_1 > w_j$, $j = 2, \cdots, m$) *primary candidate*.

Therefore, the optimal control policy

$$\boldsymbol{\pi}_k(\boldsymbol{x}_k, n_k) := \{\boldsymbol{u}_k^*(\boldsymbol{x}_k, n_k), \cdots, \boldsymbol{u}_{N-1}^*(\boldsymbol{x}_{N-1}, n_{N-1})\} \tag{4}$$

to maximize the objective function $J_k(\boldsymbol{x}_k, n_k, \boldsymbol{u}_{k:N-1})$ should be searched such that

$$\boldsymbol{\pi}_k(\boldsymbol{x}_k, n_k) := \mathop{\arg\max}_{\boldsymbol{u}_{k:N-1} \in \boldsymbol{\mathcal{U}}_{k:N-1}} J_k(\boldsymbol{x}_k, n_k, \boldsymbol{u}_{k:N-1}), \tag{5}$$

where $\mathcal{U}_{k:N-1}$ is the admissible control set from stage k to stage $N-1$. The finite-horizon optimal control problem for the spacecraft at stage k is formally stated as follows.

$$\max : J_k(\boldsymbol{x}_k, n_k, \boldsymbol{u}_{k:N-1}) = \sum_{j=1}^{m} w_j \Pr\Big[\ \boldsymbol{r}_N = \boldsymbol{r}^j\ \Big] \tag{6}$$

$$\text{find} : \boldsymbol{\pi}_k = \{\boldsymbol{u}_k(\boldsymbol{x}_k, n_k), \cdots, \boldsymbol{u}_{N-1}(\boldsymbol{x}_{N-1}, n_{N-1})\} \tag{7}$$

$$\text{s.t.} : \boldsymbol{x}_{i+1} = \boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \quad \forall i \in [k, N-1] \tag{8}$$

$$\boldsymbol{x}_k = [\ \boldsymbol{r}_k^\top,\ \boldsymbol{v}_k^\top\ ]^\top \tag{9}$$

$$u^{LB}(n_i) \le \|\boldsymbol{u}_i\| \le u^{UB}(n_i) \quad \forall i \in [k, N-1] \tag{10}$$