Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 8279–8284

Heuristic Guidance Techniques for the Exploration of Small Celestial Bodies

Francesco Capolupo * **Thierry Simeon** ** **Jean-Claude Berges** ***

* *Airbus Defence and Space, 31400 Toulouse, France (e-mail: francesco.capolupo@airbus.com)*
** *LAAS–CNRS, Université de Toulouse, CNRS, 31400 Toulouse, France (e-mail: nic@laas.fr)*
*** *Centre National d'Études Spatiales, 31400 Toulouse, France (e-mail: jean-claude.berges@cnes.fr)*

Abstract: Sampling Based Motion Planning (SBMP) techniques are widely used in robotics to plan feasible trajectories of a vehicle/robot evolving in a complex and constrained environment. Algorithms such as Rapidly Exploring Random Trees (RRT) and Sampling Based Model Predictive Optimization (SBMPO) allow for an efficient exploration of the state space, and the construction of a feasible sequence of maneuvers and trajectories that respect the kinodynamic and path constraints of the system. Proximity operations around small bodies are characterized by complex dynamics and constraints that can be easily and autonomously handled by motion planning techniques. This paper presents two motion planning algorithms designed to solve two different guidance problems: the landing on a small body and its observation. The mission scenarios considered to test the algorithms are the landing of Rosetta on the comet 67P/Churyumov-Gerasimenko and the observation of Didymain in the Didymos binary asteroid system. To conclude, the applicability of SBMP techniques to small body proximity operations are discussed. In particular, the advantages of implementing SBMP algorithms to solve complex high-level planning problems or to guide a spacecraft in a cluttered environment are highlighted.

Keywords: Guidance Navigation and Control of Vehicles, Space Exploration and Transportation, Autonomous Systems, Trajectory and Path Planning, Mission Planning and Decision Making

1. INTRODUCTION

Motion planning constitutes a very active research domain within the robotics community. Sampling Based Motion Planning (SBMP) algorithms were introduced in the '90 (Kavraki et al. (1996)) to overcome the computational complexity of the motion planning problem. The idea behind SBMP is to replace the explicit representation of the configuration space occupied by obstacles by a random sampling of the obstacle-free space, and the construction of admissible paths between samples.

The use of Sampling Based Motion Planning techniques for small bodies proximity operations was recently proposed by Pavone et al. (2014) as an effective way to deal with these challenging mission phases. Future space exploration missions will require an unprecedented level of autonomy. This need is driven by the communication delays between the ground and the spacecraft, as well as by the dynamical complexity of the explored environments. Future guidance and control solutions will also have to deal with stringent collision avoidance constraints, that considerably complicate the task of trajectory design and GNC engineers. This paper shows how complex guidance problems, such as the landing on a small body and its complete observation, can be efficiently solved with SBMP algorithms. In particular, two reference mission scenarios were considered to benchmark motion sampling techniques: the landing of Rosetta on the comet 67P/Churyumov-Gerasimenko and the observation of Didymain in the Didymos binary asteroid system. These two examples represent two very different guidance problems.

The case of Rosetta is an example of a typical constrained optimal control problem. The complex shape of the comet causes the resulting parameter optimization problem to be a hard to solve non-convex and non-smooth optimization problem. It will be shown how SBMP algorithms can solve this type of transfer problems without any difficulty.

The Didymos scenario is an example of "high-level" mission where the spacecraft is supposed to autonomously plan its trajectory in order to complete a scientific goal. No predefined waypoints are specified by mission analysts, so that the algorithm must be able to find the optimal path that fulfills a high-level task such as the observation of the entire surface of an asteroid. As it will be discussed in the following sections, SBMP algorithms are able to handle high-level objectives, and therefore can be successfully used for this type of autonomous trajectory planning problems.

⋆ This work has been co–funded by the Centre National d'Études Spatiales (the French Space Agency) under contract n°151180.

10.1016/j.ifacol.2017.08.1401

2. SMALL BODY LANDING

2.1 The Guidance Problem

The guidance problem of the small body landing scenario (i.e. Rosetta landing on the 67P/Churyumov-Gerasimenko comet, in our case) consists in bringing a lander from an initial state to a desired landing site, while minimizing the propellant consumption. The simplified translational dynamics of a spacecraft in the vicinity of the comet, written in the comet body fixed reference frame is given by

$$\ddot{\mathbf{r}} = -2\boldsymbol{\omega} \times \dot{\mathbf{r}} - \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r} + \mathbf{g}_{67P/V} + \mathbf{u} \tag{1}$$

where $\mathbf{r}$ is the spacecraft position vector with respect to the comet's center of mass, $\boldsymbol{\omega}$ is the angular rotation vector of the comet (supposed constant), $\mathbf{g}_{67P/V}$ is its gravitational attraction on the vehicle, and $\mathbf{u}$ is the control acceleration vector. The nonlinear dynamical system represented by Equation 1 can be written in a more general form as a non-linear first order autonomous system

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$$

Mathematically, the landing problem can be translated into a constrained trajectory optimization problem (or optimal control problem)

$$\begin{aligned} \underset{\mathbf{u}(t),\mathbf{x}(t),t_f}{\text{minimize}} \quad & J = \int_{t_0}^{t_f} \|\mathbf{u}\| \, dt \\ \text{subject to} \quad & \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \\ & \mathbf{x}(t_0) = \mathbf{x}_0 \\ & \mathbf{x}(t_f) = \mathbf{x}_f \\ & t_0 < t_f \leq t_{f,\max} \\ & \mathbf{x} \in X_{free} \\ & \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} \end{aligned} \tag{2}$$

where collision avoidance constraints are taken into account by defining a collision free state subspace X_{free}. The collision free subspace is mainly determined by the shape of the small body, and the presence of regions to be avoided, such as out-gassing cones. An irregularly shaped body like 67P leads to a non-convex domain X_{free}. Control authority limits are also taken into account, with the introduction of upper and lower control bounds, $\mathbf{u}_{\max}$ and $\mathbf{u}_{\min}$. There exist several numerical methods to solve the optimal control problem of Equation 2 (e.g. direct and indirect shooting and collocation methods). Nevertheless, the translation of the non-convex obstacle avoidance constraint into a nonlinear non-smooth function reduces the robustness of classic optimization methods and significantly increases the computational time. In addition, classic methods require an initial guess for both state and control profiles, and can only converge to a local optimal solution in the vicinity of the initial guess. To overcome these limitations, a new landing guidance algorithm is proposed. The new algorithm, described in detail in the following section, is based on motion planning techniques that are commonly used in robotics.

2.2 The Algorithm

The optimal Rapidly Exploring Random Tree algorithm (RRT*) was chosen to solve the landing guidance problem. RRT* was introduced by Karaman and Frazzoli (2011) to optimally solve motion planning problems in robotics. RRT* is a sampling-based motion planning algorithm designed to efficiently search non-convex, high-dimensional spaces by randomly building a space-filling tree. The tree consists of a set of vertices V (states) and edges E (trajectories connecting states), and is constructed incrementally from samples drawn randomly in the state space. The tree is rooted at the initial state and the exploration is performed until the goal is reached. Trajectories connecting state samples are computed by a local unconstrained optimization algorithm called the *steering method*.

As well explained by Karaman and Frazzoli (2011), the first step of the algorithm is to randomly sample a state vector $x_{\texttt{rand}}$ (i.e. position and velocity) from the open subspace X_{free}. The `sampleOpenSpace` function is designed to return the target state instead of a random one in a certain number of cases, as specified by the user (typically 1 to 10% of cases). The `nearest` function is then called to provide the closest node $x_{\texttt{nearest}}$ in V to $x_{\texttt{rand}}$. Next, the steering method is used to find a trajectory $\Gamma_{\texttt{new}}$ connecting $x_{\texttt{nearest}}$ to $x_{\texttt{rand}}$. As the steering method might not be able to exactly reach $x_{\texttt{rand}}$, a new node $x_{\texttt{new}}$ close to $x_{\texttt{rand}}$ is obtained (in our case, $x_{\texttt{new}} = x_{\texttt{rand}}$). If $\Gamma_{\texttt{new}}$ respects all the constraints of the problem, then a set of near (within a radius γ) neighbors $V_{\texttt{near}}$ are evaluated using the `near` function. Next RRT* calls `chooseParent` to find a candidate for a parent node to $x_{\texttt{new}}$. The function `chooseParent` returns the node in the set $V_{\texttt{near}}$ that reaches $x_{\texttt{new}}$ with minimum cost and respecting all the constraints, and it adds it to the search tree. The algorithm then tries to "rewire" the nodes in $V_{\texttt{near}}$ by calling the `rewire` function. If the feasible path that connects $x_{\texttt{new}}$ to the near node $x_{\texttt{near}}$ reaches $x_{\texttt{near}}$ with cost less than that of its current parent, then $x_{\texttt{near}}$ is rewired to $x_{\texttt{new}}$ by connecting $x_{\texttt{rand}}$ and $x_{\texttt{near}}$.

Algorithm 1 RRT*

$V \leftarrow \{x_{\text{init}}\}$
$E \leftarrow \emptyset$
$x_{\text{sol}} \leftarrow \emptyset$
for $i = 1, \ldots, n$ **do**
  $x_{\text{rand}} \leftarrow$ `sampleOpenSpace` (V_{target})
  $x_{\text{nearest}} \leftarrow$ `nearest` $((V, E), x_{\text{rand}})$
  $x_{\text{new}} \leftarrow$ `steer` $(x_{\text{nearest}}, x_{\text{rand}}, V_{\text{target}})$
  if `constraintsRespected` (Γ_{new}) **then**
    $V_{\text{near}} \leftarrow$ `near` $((V, E), x_{\text{new}}, \gamma)$
    $(V, E) \leftarrow$ `chooseParent` $(x_{\text{new}}, V_{\text{near}}, V_{\text{target}})$
    $(V, E) \leftarrow$ `rewire` $(x_{\text{new}}, (V, E), V_{\text{near}}, V_{\text{target}})$
    $x_{\text{sol}} \leftarrow$ `checkTargetReached` $(x_{\text{new}}, x_{\text{sol}}, V_{\text{target}})$
  end if
end for
return x_{sol}

The algorithm can be easily adapted to kinodynamic motion planning problems, i.e. problems having differential constraints such as $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$, provided that an appropriate steering method can be designed (Karaman and Frazzoli (2011)). As the steering method is repeatedly called during the execution of the algorithm, it must be fast enough to allow for reasonable execution times. In order to guarantee the optimality of the solution, the steering method must connect two arbitrary states by a local optimal trajectory. Unfortunately, no analytic optimal solution exists to connect two states of the system