A Nonlinear Optimization Method to Provide Real-Time Feasible Reference Trajectories to Approach a Tumbling Target Satellite

Samantha Stoneman and Roberto Lampariello

Robotics and Mechatronics Center, DLR, Germany, e-mail: firstname.lastname@dlr.de

Abstract

We present a motion planning method based on nonlinear optimization to provide reference trajectories for a robotic spacecraft approaching a non-cooperative target satellite. We perform a first investigation of a planning method containing an offline and an online element which can be used to provide reference trajectories in a useful time. The method capabilities include identifying feasible solutions in the presence of a non-static obstacle space and the treatment of a Target satellite in any general tumbling state with angular velocities less than 5 deg/s. We use a safety metric defined as the time-to-collision (ttc) with the Target satellite and show that the planning method can find the global minima for the defined problem within 50 min. The approach maneuver is parametrized using a 6 degree of freedom (dof) task space which describes the final position of the grasping point on the Target satellite and the satellite's final angular velocity. First a global search composed of a coarse, randomized optimization plus a finer smoothing is performed, then the online algorithm interpolates between the pre-computed solutions and provides the reference trajectory to an on-board controller. An estimate for the offline computation times, up to 6 days for a sparse discretization, shows that in the presence of inertia uncertainties of the Target satellite, using parallelized optimization in an online global search can provide solutions on the order of minutes and is more attractive.

1 INTRODUCTION

Motivations for developing robotic, on-orbit servicing technologies are abundant. A particular use case which has become increasingly interesting the last few years is the de-orbiting of defective satellites. Approaching an uncontrolled body in orbit is a difficult problem due to the risk of collision between the Target and the robot, as well as operational constraints such as communication link coverage. Several independent groups have defined such a servicing mission to require a series of operational segments: initial approach, fly-around and visual inspection, final approach, coarse manipulation, visual-servoing and capture [1, 2]. We address the problem of planning reference trajectories for the final approach segment of an on-orbit servicing mission, where a Servicer maneuvers to a synchronous mating position with a generally tumbling Target, shown in Fig. 1. The problem is treated as a Nonlinear Boundary Value Problem (NLBVP) for a free flying, rigid body in the presence of nonlinear dynamics and constraints, including a time-varying obstacle space created by the

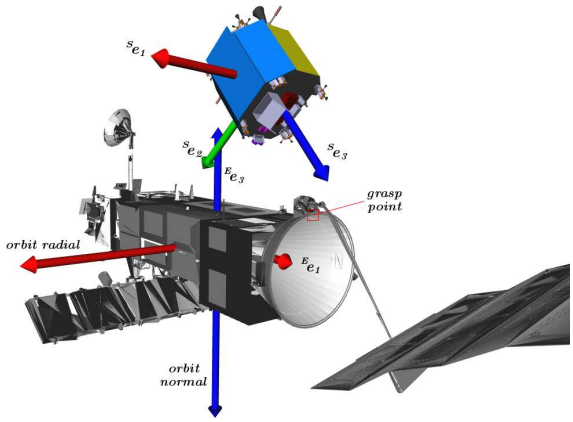

Figure 1: *A robotic Servicer in a synchronous mating postion with a Target.* $^{E}\mathbf{e}$ *and* $^{S}\mathbf{e}$ *denote the Target and Servicer body-fixed frames. The inertial frame is aligned with Envisat's body-fixed frame. The orbit radial direction points away from Earth and the along-track direction is aligned with the inertial* $-x$*-axis.*

tumbling Target. Feasible reference trajectories are important for the approach maneuver since the use of a traditional feedback controller without a reference trajectory gives no guarantee on its performance, for example maneuver execution time or behaviour under active motion contraints.

A Target example which is currently being investigated by the European Space Agency is the Envisat satellite (Fig. 1). Communication with Envisat was lost in 2012, and it is currently estimated to be in a low velocity, uncontrolled tumble [3]. Envisat poses a high risk for creating orbital debris, presenting an ideal application for our real-time motion planner. [1] proposed that the most crucial metric for the approach segment of robotic servicing is safety. This is a good assumption since the approach maneuver requires much less delta-V in comparison to deorbiting with the Target and as such a classical energy consumption metric is less interesting. The motion planning method solves the resulting NLBVP using a Gradient Based Optimization (GBO) technique which assures that the nonlinear motion constraints are satisfied, and that a safety cost metric is minimized. The method uses clamped, nonuniform B-Splines to represent the trajectories. Similarly, a generally tumbling target was considered in [4] where only energy and time were used as the cost metrics. Recently [5] solved as well a similar problem making use of an offline computation element and numerical solutions for a spinning Target. [6] uses a two part, offline and on-line method to compute optimal solutions for the on-orbit capture problem of a small spinning Target, inspiring the LUT method in this work.

The BVP is first solved offline using an exhaustive search where the tumble state of the Target satellite is the variable parameter for the general planning task. Given the dof of the tumble, the search is performed for all possible combinations of task parameters, generating a Look-Up Table (LUT) of feasible, safe trajectories discretized by the possible states of the Target. In the online segment, following in situ measurement of the actual motion state of the Target, we choose the closest matching solution available in the LUT, resulting in a reference trajectory in real-time. Real-time is defined given estimated communication and operational time constraints of the problem and resulting, allowable online computation time on the order of minutes. This can then be fed to an on-line control method, which can track the reference trajectory, while reaching the desired goal.

However, the inertia of Envisat is subject to uncertainties and therefore a method which can be computed on ground during a mission is attractive. For that purpose we as well evaluate the efficacy of a purely online method which may be more appropriate for application to Envisat.

The paper is structured as follows: Section 2 describes the problem statement and the formulation of the optimization problem, Section 3 describes the first part of the planning method, the global minimum search required to build the LUT, Section 4 describes the on-line part of the method and in Section 5 is the conclusion.

2 MOTION PLANNING PROBLEM

The approach problem can be formulated as a general optimization problem to minimize a cost which is a function of free parameters $\mathbf{p}$ describing the path of the system $\mathbf{r}(\mathbf{p}, t)$ and its derivatives, given inequality constraints on the states along the trajectory. The robot pose is $\mathbf{r}(t) \in \mathbb{R}^n$ where n consists of three translations and three rotations expressed as the four components of a unit quaternion, $\mathbf{r}(\mathbf{p}, t) = [r_x\ r_y\ r_z\ q_x\ q_y\ q_z\ q_w]$. The time series of poses is the trajectory. The final position is at the mating point $\mathbf{r}_{mp}(t)$, where the motion is synchronized, the relative velocities and accelerations w.r.t. the Target are zero. The general problem statement is,

$$\min_{\mathbf{p}} \Gamma(\mathbf{p}) = \int_{t_0}^{t_f} f\left(\mathbf{r}(\mathbf{p}), \dot{\mathbf{r}}(\mathbf{p}), \ddot{\mathbf{r}}(\mathbf{p}), t\right) dt \tag{1}$$

subject to:

$$\begin{aligned}
\mathbf{c}_{ineq}(\mathbf{r}(\mathbf{p}), \dot{\mathbf{r}}(\mathbf{p}), \ddot{\mathbf{r}}(\mathbf{p}), t) &\leq 0 \\
\mathbf{r}(\mathbf{p}, t_0) &= \mathbf{r}_0 \\
\dot{\mathbf{r}}(\mathbf{p}, t_0) = \ddot{\mathbf{r}}(\mathbf{p}, t_0) &= 0.0 \\
\mathbf{r}_{lin}(\mathbf{p}, t_f) &= \mathbf{r}_{mp} \\
\dot{\mathbf{r}}_{lin}(\mathbf{p}, t_f) &= \omega_e \times \mathbf{r}_{mp} \\
\ddot{\mathbf{r}}_{lin}(\mathbf{p}, t_f) &= \dot{\omega}_e \times \mathbf{r}_{mp} + \omega_e \times (\omega_e \times \mathbf{r}_{mp}) \\
\mathbf{r}_{rot}(t_f), \dot{\mathbf{r}}_{rot}(t_f), \ddot{\mathbf{r}}_{rot}(t_f) &= f(\mathbf{A}_E, \omega_e, \dot{\omega}_e),
\end{aligned} \tag{2}$$

where $\mathbf{A}_e$, ω_e and $\dot{\omega}_e$ are the rotation and angular velocity and acceleration of the Target and $t_0 \leq t \leq t_f$. The trajectory is discretized and must be feasible at a predefined number of via points n_{via}. The cost $\Gamma(\mathbf{p})$ and the inequality constraints $\mathbf{c}_{ineq}(\mathbf{r}(\mathbf{p}), t)$ encode the equations of motion of the robotic Servicer, which are coupled due to the orbital and rotational dynamics and a line of sight constraint,

$$\mathbf{c}_{ineq} = \begin{array}{c} \mathbf{c}_{collision}(\mathbf{r}(\mathbf{p}), t) \leq 0 \\ \mathbf{c}_{velocity}(\dot{\mathbf{r}}(\mathbf{p}), t) \leq 0 \\ \mathbf{c}_{actuation}(\mathbf{r}(\mathbf{p}), \dot{\mathbf{r}}(\mathbf{p}), \ddot{\mathbf{r}}(\mathbf{p}), t) \leq 0. \end{array} \tag{3}$$

We solve Eq. (1) using the sequentially least-squares quadratic programming (SLSQP) implementation from the open source NLopt library [8]. The algorithm is gradient based and the numerical gradients for Γ and $\mathbf{c}_{ineq}$ are provided by perturbing the parameters $\mathbf{p}_i + \delta$ and computing the forward finite differences.

2.1 Cost Function

The cost function of the approach problem is a weighted combinatorial cost of three different values, linear and rotational power and ttc. The ttc of the system is the more important metric and thus has the highest weight. However, it must be combined with the system energies in order to ensure that the resulting motion is smooth and feasible from an engineering standpoint.

2.1.1 Power

The power cost metric is expressed as follows, where the units are J^2/s,

$$P_L(\mathbf{p}) = \int_{t_0}^{t_f} \mathbf{F}(\mathbf{r}(\mathbf{p}, t))^2 \dot{\mathbf{r}}(\mathbf{p}, t)^2 dt, \tag{4}$$

$$P_R(\mathbf{p}) = \int_{t_0}^{t_f} \tau(\omega(\mathbf{p}, t), \dot{\omega}(\mathbf{p}, t))^2 \omega(\mathbf{p}, t)^2 dt. \tag{5}$$

2.1.2 Time-To-Collision

[1] describes the ttc as the time until the satellites collide given sudden, total loss of Servicer actuation. The system drift is computed by integrating the force and torque free equations of motion starting at each via point along the trajectory for a predefined length of time, ttc_{max}. ttc_{max} is the best representable cost and the global minimum of the model occurs when $ttc_{act}(\mathbf{p}, t) = ttc_{max}$ for all t. Given the geometry of the Servicer we argue that the rotational drift negligibly affects the ttc and treat the attitude as constant. The cost is formulated as the difference between the integral of the best possible ttc_{max} and the actual ttc_{act},

$$ttc(\mathbf{p}) = \int_{t_0}^{t_f} (ttc_{max} - ttc_{act}(\mathbf{p}, t))\, dt \;\; (s^2). \tag{6}$$

The combinatorial cost function can then be expressed as the weighted sum of each separate metric,

$$\Gamma(\mathbf{p}) = w_0 P_L(\mathbf{p}) + w_1 P_R(\mathbf{p}) + w_2 ttc(\mathbf{p}). \tag{7}$$

2.2 Equations of Motion and Visual Line of Sight

The system is a fully actuated rigid body governed by orbital dynamics and the Euler equations of motion. The robotic Servicer and Envisat are assumed to be in the same orbit and so the linear Hill dynamics are used as the linear motion model,

$$\begin{array}{c} \ddot{x} + 2n\dot{y} - 3n^2 x = f_x \\ \ddot{y} + 2n\dot{x} = f_y \\ \ddot{z} + n^2 z = f_z, \end{array} \tag{8}$$

such that,

$$\frac{\mathbf{F}}{m} = \begin{bmatrix} -3n^2 & 0 & 0 & 0 & 2n & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2n & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & n^2 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_{lin} \\ \dot{\mathbf{r}}_{lin} \\ \ddot{\mathbf{r}}_{lin} \end{bmatrix}, \tag{9}$$

and Euler's equations of motion,

$$\mathbf{I}\dot{\omega}(t) + \omega(t) \times \mathbf{I}\omega(t) = \tau(t), \tag{10}$$

where $n = 0.0012$ is the orbital mean motion. Only the three linear dofs are included in the optimization problem; however, the robot must maintain a line of sight with the defined grasping point, $\mathbf{r}_{gp}$, such that two of the three rotational dof are fixed given the rotational motion of Envisat. For simplicity only exact tracking of the defined point on Envisat is used; however, the tracking could as well be computed as a region specified by the field of view of the on-board sensors. The tracking is implemented by defining the forward pointing unit

vector of the Servicer $\hat{\mathbf{n}}_{nom}$ and solving for the rotation which projects it onto the vector connecting the Servicer CoM and $\mathbf{r}_{gp}$, $\hat{\mathbf{n}}_{des}$. All rotations are represented as unit quaternions such that,

$$\mathbf{q}_{des}(t) = f(\mathbf{r}(t), \mathbf{r}_{gp}(t), \hat{\mathbf{n}}_{nom}), \tag{11}$$

where f is a function projecting $\hat{\mathbf{n}}_{nom}$ onto $\hat{\mathbf{n}}_{des}$. This creates a coupling effect between the robot position and orientation, which affects the system cost functions as well as the constraints. Due to the geometry of the Servicer, the rotation about the third axis was not included as a parameter in the NLP. However, all 6 dof are constrained by $\mathbf{c}_{ineq}$.

The Servicer's initial state at t_0 is in the orbital plane phased behind Envisat such that ${}^{I}\mathbf{r}(t_0) = (39\ 0.0\ 0.0)$, where the inertial frame is located at Envisat's CoM and aligned with the initial attitude of Envisat, depicted along with the orbital frame in Fig. 1. We assume at t_0 and t_f the Servicer has zero linear velocity and acceleration w.r.t. Envisat. From this goal state the robot could then perform a capture maneuver, where the position is $\mathbf{r}_{mp}$,

$${}^{I}\mathbf{r}(t_f) = {}^{I}\mathbf{r}_{mp} = \mathbf{A}_{EI}(t_f)\,{}^{E}\mathbf{r}(t_f), \tag{12}$$

where ${}^{E}\mathbf{r}(t_f)$ is a fixed position w.r.t. the Target CoM and the Servicer is pointing at $\mathbf{r}_{gp}$.

2.3 Simulation of the Envisat satellite

We identify the parameters of the approach task to be the final pose of the grasping point, $\mathbf{r}_{gp}(t_f) = f(\theta_{gp}, \delta_{gp}, \phi_{gp})$ and Envisat's final rotational velocity vector $\omega(t_f) = (\omega_x, \omega_y, \omega_z)$, which fully describe the general tumbling state of the the Target. The tumbling motion of the Target satellite is modelled using a Runge-Kutta 4th order integrator and the force-free case of Eq. (10), using the estimated inertia of Envisat given in [3] and the final orientation $\mathbf{q}_{t_f} = f(\theta_{gp}, \delta_{gp}, \phi_{gp})$. [3] gives the nominal angular velocity to be approximately 3.5 deg/s. Example results of the Envisat motion for the nominal angular velocity about the major and middle axes are shown in Fig. 2 and 3. For the flat spin nutation shown in Fig. 2, the angular velocity is relatively constant and the collision space where the Servicer must maneuver around is also constant from one rotation period to the next. In contrast, in the tumble case the nutation angle oscillates from zero to 25 deg. As shown

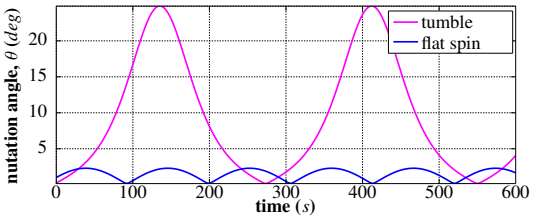

Figure 2: *Nutation angle over time for the nominal tumble motion $\omega_y(t_0) = 3.5$ deg/s compared to a relatively flat spin where $\omega_z(t_0) = 3.5$ deg/s.*

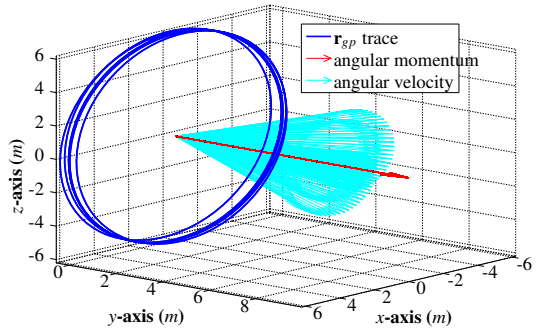

Figure 3: *A trace of $\mathbf{r}_{gp}(t)$ for the nominal $\omega_y(t_0) = 3.5$ deg/s, which nutates nearly 1 meter in and out of the initial spin plane.*

by the trace in Fig. 3, this causes the grasping point to oscillate one meter in and out of the plane perpendicular to the angular momentum vector. For the general case of a tumbling satellite the nutation angle can become sufficiently large that a particular method such as approaching the target along the angular momentum vector may result in collision.

2.4 Collision Model

The collision avoidance is integrated in the NLP as an inequality constraint at each viapoint in the trajectory,

$$\mathbf{c}_{collision}(\mathbf{r}(\mathbf{p}), t) = \mathbf{PD}(\mathbf{r}(\mathbf{p}), \mathbf{M}, t) \leq 0, \tag{13}$$

where $\mathbf{M}$ is the geometric model and **PD** is the penetration depth, defined as the minimum distance required to bring an object pair out of collision. The **PD** function is a nonlinear, iterative function. We have imple-

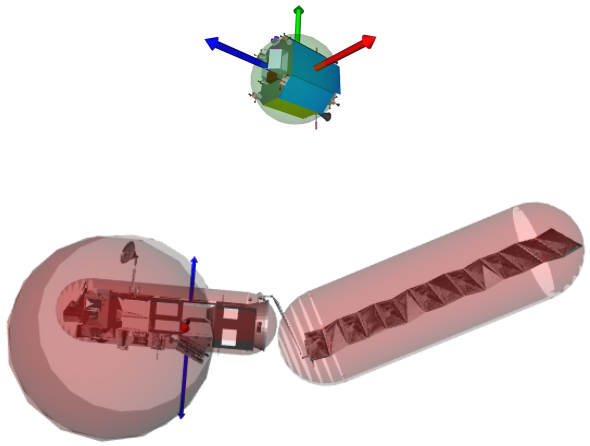

Figure 4: *The primitive model of the Servicer is a simple sphere. The composite model of the Target containing 3 separate capsules is the result of an effort to mitigate local minima in the NLP by removing as many nonconvex features in the model, such as the communication dish and radar antenna, at the cost of sub-optimality in the representation of the geometry.*

mented it using the open source Bullet Physics Collision Library [7], which makes use of the GJK algorithm to compute the contacts. The collision model used in all of the following simulations is shown in Fig. 4. Analyses showed that the collision constraints are the limiting constraint in the NLP and cause local minima in the cost function. This characteristic is due to the nonconvexity of the model; however, it is maximally reduced by the model shown.

2.5 Nonuniform Spline parametrization

The trajectories $\mathbf{r}(t)$ and $\mathbf{q}_z(t)$ are individually parametrized with clamped, *nonuniform* B-Splines of order $k = 4$ and degree $d = 3$. The general equation for B-Splines and algorithm for computation of the basis functions can be found in [9]. In [12] clamped, uniform spline curves were used. This method can be numerically problematic for sampling based planners in highly constrained spaces because the state BCs are encoded in the parametrization by fixing an equal number of spline vertices as the desired number of fixed conditions n_{fbcs} at $\tau_0 = 0$ and $\tau_f = 1$. Such that,

$$\mathbf{S}^r(\mathbf{p}, \tau_i) = \mathbf{N}^r(\bar{\tau}, \tau_i)\,[\mathbf{p}_0^T\ \mathbf{p}^T\ \mathbf{p}_f^T]^T, \text{ for } r \in [0\ d], \quad (14)$$

is solved for the fixed vertices $\mathbf{p}_0$ and $\mathbf{p}_f$ given $\mathbf{S}^r$ at t_0 and t_f for the desired number of r, where $\mathbf{S}^r(\mathbf{p}, \tau_i) = (n_{via} \times 1)$ is the spline curve, $\mathbf{N}^r(\bar{\tau}, \tau_i) = (n_{via} \times n_{vts})$ is the basis function matrix, $[\mathbf{p}_0^T\ \mathbf{p}^T\ \mathbf{p}_f^T]^T = (n_{vts} \times 1)$ are the fixed and free spline vertices, r is the time derivative, $\bar{\tau}$ is the knot vector and τ_i is the independent variable which we equate with time. We fix three time derivatives at each boundary, the positions, velocities and accelerations. The free vertices $\mathbf{p} = (n_p \times 1)$ for $n_{fbcs} = 3$ are the optimization parameters, where $n_p = n_{vts} - 2n_{fbcs}$. The free rotation $\mathbf{q}_z(t)$ is represented with a spline; however, the parameters are not free in the optimization. This is discussed further in Section 3.1. The initial and final spans of the spline curve are less or not affected at all by the free vertices. If constraint violations are active, this will result in nonconvergence of the NLP. This effect can be minimized by selecting $\bar{\tau}$ such that some knots have a short span, ensuring *high dynamics* are representable at the end spans. This improvement in the parametrization is presented as an additional contribution to the motion planning method.

3 GLOBAL MINIMA SEARCH

The approach problem described in Section 2 is nonlinear and nonconvex. Statistical simulation data has shown it to be subject to local minima in the cost space due to the collision model and symmetries of the cost functions. The classical approach to finding the global minimum of such a system is to perform a Monte Carlo search on the optimization parameters, defining the best solution found as the *global minimum*. Instead of randomly sampling the parameters themselves the search can be done alternatively using random path planning algorithms such as RRT-type trees. This results in a multi-step process to solving Eq. (1), where the path to the goal is subdivided into many, connected edges.

We compared three RRT-type algorithms: RRT [10], RRT* [11], RRT*-GBO [12], as well as a random way point method. All four search algorithms sample the state space and in the case of RRT*-GBO and the random way point method,

optimize the intermediate paths as individual NLBVPs. The resulting intermediate paths are smoothed into one trajectory from the initial state of the Servicer to the required final state. The random way point method performed significantly faster than the tree-based algorithms and produced initial guesses which converged more often to a local minimum, thus it was selected as the best method to identify the global minima in the task space.

3.1 Random Way Point Method

The random way point method samples a way point $\mathbf{wp} = [\mathbf{r}(t_i)\ t_i]$, in the problem state space $\mathbb{R}^{n+1}$. $n+1$ is the random position, velocity and acceleration for each dof and additionally the instantaneous time at the way point t_i. The algorithm first attempts to solve a NLBVP from the Servicer initial state to the way point, $\mathbf{r}(t_0) \rightarrow \mathbf{r}(t_i)$ and upon success attempts to solve a NLBVP connecting the way point to the mating point, $\mathbf{r}(t_0) \rightarrow \mathbf{r}(mp, t_f)$. When a final edge is successful, the two separate edges which are continuous at the way point are returned as a solution. The free rotation about the Servicer's z-axis was also randomly sampled during the way point search and the rotation was superposed on the total pose of the Servicer.

The initial guess of the spline parameters for each edge in the random way point search was determined by solving a strictly convex QP for the minimum acceleration path given the BCs, the kinematics of the path and the linear constraints $\mathbf{c}_{velocity}$. We found that this is a good starting point, especially for the power cost metric which is often close to the minimum acceleration solution when the nonlinear actuation and collision constraints are not active.

The random way point method performs a coarse search of the state space and provides the resulting trajectories as the initial guess for additional iterations of smoothing, which are again formulated as the same NLBVP, with finer accuracies and function tolerances. The same settings were used for all of the results presented in this report. The function accuracies used in the NLBVP were for the coarse search: $\eta_{\Gamma,\,\text{coarse}} = \delta_{\Gamma,\,\text{coarse}} = \delta_{ineq,\,\text{coarse}} = 1e^{-6}$ and $\delta_{\Gamma,\,\text{coarse, ttc}} = \delta_{collision,\,\text{coarse}} = 1e^{-4}$ where η is the function accuracy and stopping condition of the optimization and δ refers to the numerical gradient perturbation step. The collision constraints and ttc constraints are functions of the penetration depth computation and require a larger step size to produce nonzero gradients. And for the fine smoothing: $\eta_{\Gamma,\,\text{fine}} = \delta_{\Gamma,\,\text{fine}} = \delta_{ineq,\,\text{fine}} = 1e^{-9}$ and $\delta_{\Gamma,\,\text{fine, ttc}} = \delta_{collision,\,\text{fine}} = 1e^{-4}$. Additionally the cost and its gradients were scaled by a factor $\alpha = 1e^{4}$ more than the constraints, such that $\Gamma_\alpha = \alpha\Gamma$ and $\left(\frac{\delta\Gamma}{\delta\mathbf{p}_n}\right)_\alpha = \alpha\frac{\delta\Gamma}{\delta\mathbf{p}_n}$.

The weighting scheme was set to $\mathbf{w} = (w_{P_L}\ w_{P_R}\ w_{ttc}) = (10\ 100\ 1)$. The maximum collision time parameter was set to $ttc_{max} = 300$ s. The planning method is capable of solving both finite- and infinite-time horizons; however, the approach problem is inherently finite and we set $t_f = 600$ s as the maneuver time. In the coarse search $n_p = 5$ and $n_{via} = 75$ was used for each edge, and in the fine smoothing, $n_p = 15$ and $n_{via} = 400$. In the fine smoothing this resulted in a total of 45 optimization parameters and 5200 inequality constraints. These settings strongly affect the convergence time, optimality and feasibility of the solution, where a decrease in the number of via points and reduction of accuracy will decrease convergence time at the expense of optimality and feasibility.

3.2 Statistical Analysis of the Cost Metrics

A statistical analysis was performed for both a pure power cost metric ($w_{ttc} = 0$) and the combinatorial metric to determine the computation time and search effectiveness for computing one Envisat motion state. The statistics were done as a Monte Carlo search for each cost metric, where 50 simulations were run. A simulation consisted of seeding a random number generator and sampling random way points until 20 solutions were found, resulting in 1000 solutions. The task parameters were $(\theta_{gp}\ \delta_{gp}\ \phi_{gp}) = (0\ 0\ 0)$ deg and $\omega_{t_f} = (0\ 3.5\ 0)$ deg/s. The initial position of the Servicer was phased behind the Envisat as well as out of the orbital plane such that $\mathbf{r}(t_i) = (39\ 39\ 39)$ m to test the method. The motion planning software was implemented entirely in C++ and the results were computed using a network of machines running Intel x86 and x86-64 processors.

Fig. 5 through 7 show the ttc at each via point along the trajectory and the trajectory components themselves for a global minimum (the best solutions found) for each cost metric. The respective cost histograms for all 1000

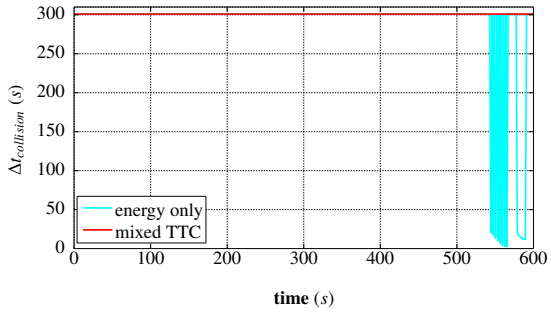

Figure 5: *ttc profiles for a power optimal solution (cyan) and a ttc optimal solution (magenta). The combinatorial cost metric is able to remove all dangerous points in the power optimal path, where the instantaneous ttc is less than the maximum and at some time steps approaches 1 s.*

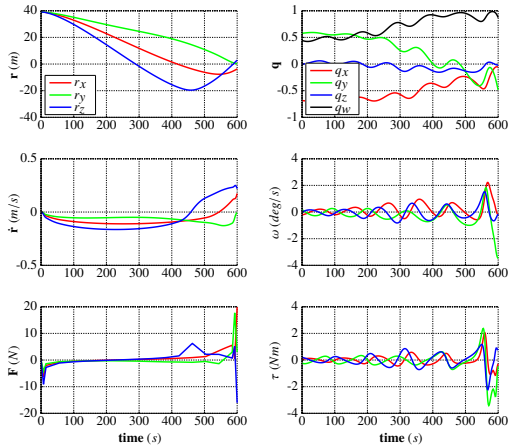

Figure 6: *The linear and rotational trajectories for a power optimal solution. The angular velocities appear to be unstable, but this effect is the nature of the line of sight motion constraint.*

solutions are shown in Fig. 8 and Fig. 9. The power metric contains five, well-defined minima. A global minimum is clearly defined, while the lesser minima have a much lower frequency. The small scattering of solutions into lesser minima is mostly due to the nonconvexity of the collision model. The cost histogram of the combinatorial cost which includes ttc is significantly different. While a clear minimum exists, the scatter

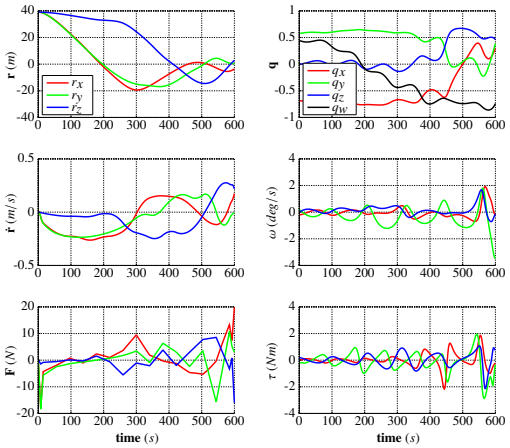

Figure 7: *The linear and rotational trajectories for a ttc optimal solution. The combinatorial cost improvement in safety is at the expense of power optimality and general smoothness, showing the two metrics most often have opposing descent directions in* **p**.

shows more than 500 solutions converged on a wide set of sub-optimal ttc. The NLP optimization parameters, $\mathbf{p} \in \mathbb{R}^{45}$, can be visualized by the translational positions, as shown in Fig. 10 and Fig. 11. In the case of the power metric the global minimum is clearly defined in the parameter space, while the combinatorial cost global minimum is densely scattered in the geometrically parameter space. However, there is a clear, symmetric global minimum in y during the last 100 s of the maneuver, showing that in order to maintain a optimal ttc the Servicer must choose an approach which moves laterally across the top of the Envisat bus.

The safety metric acts like a collision constraint and for this reason the argument could be made that the problem is rather a collision avoidance problem than an optimization one, although representing the ttc as the cost is necessary to optimize it at the end of the approach. The scatter of unconverged solutions for the combined metric shows the difficulty in solving for a safe trajectory. In this case the weighting of opposing metrics causes many solutions which are sub-optimal in both safety and power. This characteristic could be mitigated in future work by solving the problem rather as a two-step optimization, where the best ttc solution is

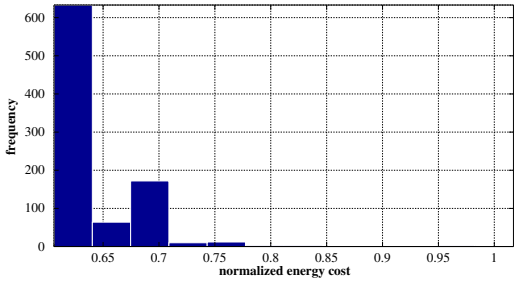

Figure 8: *The histogram of solutions for the Monte Carlo set which optimized only the power, where the y-axis is the number of solutions in each bin. All 1000 solutions fit in 5 solution bins and the global minimum is clearly defined. This minima are mostly due to the nonconvexity of the collision model.*

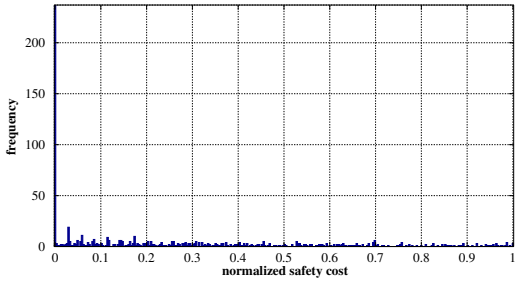

Figure 9: *The cost histogram of the ttc for the Monte Carlo set which minimized the combinatorial cost, where the y-axis is the number of solutions in each bin. A clear GM exists, but there is wide scatter of seemingly unconverged solutions.*

found and then held as a constraint while a smoothing metric such as power or distance is minimized.

The most important performance criterion is how long the coarse search must run to compute a grid point in the LUT. This was determined using the global minimum percent likelihood, shown in Fig. 12. For brevity only the combinatorial cost metric data is shown. If 90% success rate for finding the gloabl minimum is required, it can be determined from the plots that a grid point must run for a mean of 50 min, where the statistical $\pm\sigma$ bounds of the computation time were [20 80] min. The computation time for the power metric is less than half this result. This is intuitive considering the additional collision checks that must be computed at each via point in the trajectory. Additionally, the competing metrics in the weighted, combinatorial scheme further slow convergence. However, since the offline computation should be performed only once, one could argue that it is mostly irrelevant.

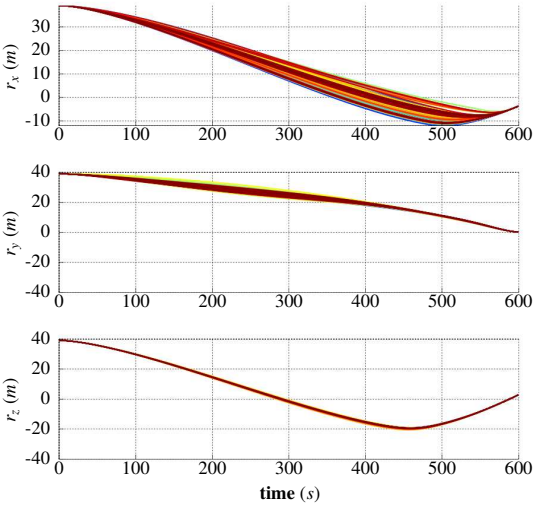

Figure 10: *The translational components of position for the global minimum solutions of the Monte Carlo power set. The global minimum is clearly defined geometrically.*

4 ONLINE COMPUTATION METHOD AND ANALYSIS

4.1 Computing a Look Up Table

The LUT is computed by performing the offline way point search for all combinations of final grasping point latitude, longitude and x-axis rotations as well as the final angular velocity of Envisat. Discretization of the grasping point coordinates was limited to a half sphere since the Servicer will begin the approach segment from a fixed initial position. We chose to discretize attitude and velocity with 7 and 5 grid points uniformly in each dof, such that, $\theta_{gp}, \delta_{gp}, \phi_{gp} \in [-90\ 90]$ and $\omega_x, \omega_y, \omega_z \in [-5\ 5]$. This amounts to pre-computing 117,649 initial condition combinations. Table 1 gives the offline computation time estimates for the statistical range of computation times shown in Fig. 12, assuming the solutions are computed on an available cluster with

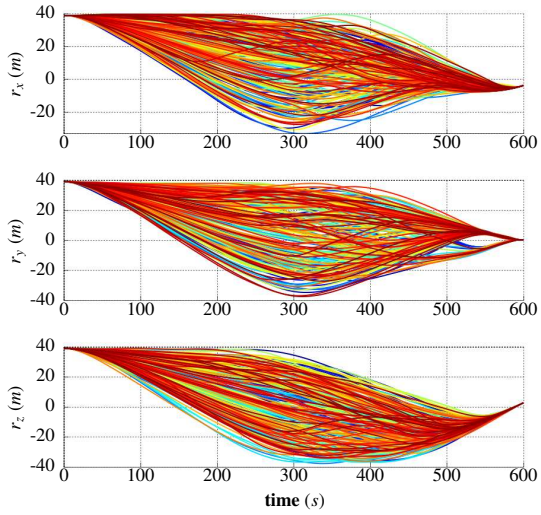

Figure 11: *The global minimum solutions for the combinatorial cost Monte Carlo set. The global minimum for ttc is densely scattered in the NLP parameter space, presenting many geometric symmetries.*

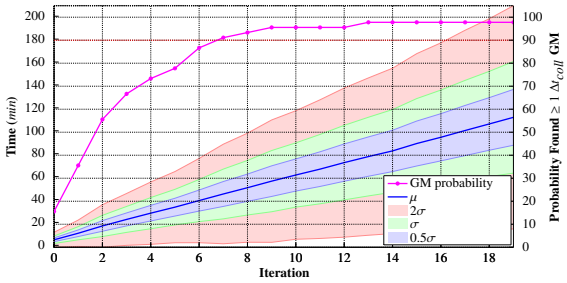

Figure 12: *The GM probability curve shows how many iterations in the way point search are statistically required until a global minimum is found. The mean computation time and standard deviations are also plotted as a reference.*

256 cores. The total computation is linear with the bin computation times and exponential with the dof of the task and discretization. The computation time is,

$$\Delta t_{LUT} = \frac{\Delta t_{bin} \times n_{gridpts}^{n_{dof}}}{60 * 24 * 256} \tag{15}$$

The online problem constitutes interpolating the pre-computed solutions. A preliminary interpolation was performed for one point between gridpoints, $(\theta_{gp}\ \delta_{gp}\ \phi_{gp})$ = (0 5 0) deg and ω_{t_0} = (0 3.5 0) deg/s. A 5-point interpolation was done using a simple inverse distance weighting of order $p = 6$ given the distance measured in the task parameter space for the **p** of the global minima from each bin. There is no guarantee that a global minimum lies between the parameter vectors for n-known minima. For the test problem, the resulting trajectory was infeasible and it was necessary to perform a post-processing rough optimization to remove active constraints and bring the ttc back to a global minimum, which required 2.7 min computation time.

Δt_{bin} (min)	20	50	80
Δt_{LUT} $(days)$	2.33	5.82	9.30

Table 1: *Computation times for a complete LUT in days as $f(x)$ of Δt_{bin} in min for the $-\sigma$, mean and $+\sigma$ per bin computation times.*

Due to the disparities in the spline parameters between global minima and Envisat tumble states, a single point interpolation may be preferable, where the closest point in the LUT is immediately chosen as the guess for this desired point. This problem presents future work for defining the correct interpolation method which yields feasible, optimal reference trajectories. It is likely that to achieve good interpolation results a dense grid in the LUT is required, which will exponentially increase the computation time. Since the LUT is an offline computation, it can be argued that this is irrelevant, unless a case exists where one might want to recompute the LUT on ground, during the de-orbiting mission.

4.2 **Distributed, Online Global Minimum Search**

Given the long computation times for the LUT, we acknowledge a realistic strategy which entails a simple parallelization of the offline search, after determining the required task parameters during orbital observation. This allows for treating any inertial uncertainty of the Target, as shown by [13]. Given the mean computation time of 50 min for a single task point, distributing the gradient computations in the NLP during the global search would significantly reduce the total computation time without the need for threading within the SLSQP algorithm itself. To roughly determine the total compu-

	$n_{thr} = 1$	$n_{thr} = n_p$	$n_{thr} = n_p + n_{via}$
$\Delta t_{grads} = 98\%$	50 *min*	2.08 *min*	1.11 *min*

Table 2: *Estimate for the computation times of a global search using parallelized optimization as a function of the total fraction of computation time spent on computing the gradients Δt_{grads}, and the number of threads n_{thr}.*

tation required after parallelization, the fraction of total computation time for one optimization which is devoted to the gradient computations can be divided by the number of distributed threads. The time gains if 98% of the total computation time is spent on calculating the gradients, is shown in Table 2. A full analysis of the performance of parallelized optimization for the online planning presents future work.

5 CONCLUSIONS

We have presented a motion planning method which can optimally maximize the time-to-collision between a robotic Servicer and a Target satellite in orbit. Identifying the Envisat Satellite as a good use-case, the comparison between optimal power and safe solutions showed that a robotic servicing mission would benefit from planning a safe approach trajectory, that the safety cost is often competing with minimum power solutions. A two-part method is proposed, which performs the bulk of the computations offline. The resulting LUT can be computed in approximately 1-10 days time, depending on the discretization of the table and the accuracy of the global search. A first example showed that the method can be used to provide a feasible, optimal reference trajectory after online post-processing; however the raw interpolation result was not feasible. A rigorous analysis of the interpolation method presents future work, to determine if the method can reliably return *feasible* trajectories and how fine the discretization must be. Alternatively, we presented a method for parallelizing the offline computation and thereby performing global searches online, a useful result in the case that the Envisat inertia error is high enough that reference trajectories must be recalculated in situ.

References

[1] Jacobsen, S., Lee, C., Zhu C., Dubowsky, S., "Planning of Safe Kinematic Trajectories for Free Flying Robots Approaching an Uncontrolled Spinning Satellite". DETC 2002. September 2002. Montreal, Quebec, Canada.

[2] Yoshida, K., "Space Robot Dynamics and Control: To Orbit, From Orbit, and Future". Robotics Research. The Ninth International Symposium.

[3] B. Bastida Virgili, S. Lemmens, H. Krag. "Investigation on Envisat attitude motion". e.Deorbit Workshop, May 2014.

[4] G. Boyarko, O. Yakimenko, and M. Romano, "Optimal Rendezvous Trajectories of a Controlled Spacecraft and a Tumbling Object". Journal of Guidance Control and Dynamics, vol. 34, no. 4, pp. 1239–1252, 2011.

[5] Starek, Joseph A., Edward Schmerling, Gabriel D. Maher, Brent W. Barbee, and Marco Pavone. "Fast, Safe, and Propellant-Efficient Spacecraft Planning under Clohessy-Wiltshire-Hill Dynamics". 37th IEEE Aerospace Conference, 2016, Big Sky, USA.

[6] Lampariello, R., Hirzinger, G., "Generating Feasible Trajectories for Autonomous On-Orbit Grasping of Spinning Debris in a Useful Time", IROS, Tokyo, Japan, November 2013.

[7] Coumans, E. Bullet 3d real-time multiphysics library. *http://bulletphysics.org/wordpress*.

[8] Johnson, S. G. The nlopt non-linear optimization package. *http://ab-initio.mit.edu/nlopt*.

[9] De Boor, C. *A Practical Guide to Splines*. 1978. Springer-Verlag.

[10] La Valle, S., Kuffner, Jr., J.J., "Randomized Kinodynamic Planning". *The International Journal of Robotics Research*, Vol. 20, No. 5, 378-400 (2001).

[11] Karaman, S., Frazzoli, E., "Sampling-based Algorithms for Optimal Motion Planning". *The International Journal of Robotics Research*, vol. 30, iss. 7, pp. 846-894, 2011.

[12] Stoneman, S., Lampariello, R., "Embedding Nonlinear Optimization in RRT* for Optimal Kinodynamic Planning". CDC, Los Angeles, USA, December 2014.

[13] U. Hillenbrand, R. Lampariello. "Motion and Parameter Estimation of a Free-Floating Space Object from Range Data for Motion Prediction". 8th International symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2005). Munich, Germany. 5-9 September 2005.