Generating Feasible Trajectories for Autonomous On-Orbit Grasping of Spinning Debris in a Useful Time

Roberto Lampariello and Gerd Hirzinger

***Abstract*— The grasping and stabilization of a spinning, non-cooperative target satellite by means of a free-flying robot is addressed. A method for computing feasible robot trajectories for grasping a target with known geometry in a useful time is presented, based on nonlinear optimization and a look-up table. An off-line computation provides a data base for a mapping between a four-dimensional input space, to characterize the target motion, and an N-dimensional output space, representing the family of time-parameterized optimal robot trajectories. Simulation results show the effectiveness of the data base for computing grasping maneuvers in a useful time, for a sample range of spinning motions. The debris object consists of a satellite with solar appendages in Low Earth Orbit, which presents collision avoidance and timing challenges for executing the task.**

I. INTRODUCTION

The advent of the use of robots for the removal of space debris in Earth orbit may be very close. A missing step in the methodology available for autonomously grasping and stabilizing a non-cooperative tumbling debris object (or Target), like an uncontrolled satellite or the upper stage of a launcher, by means of a free-flying robot, is that of being able to generate a feasible robot reference trajectory, for any typical geometry and motion of the debris object and in a useful time. This paper presents an analysis of the implementation of a grasping control method, first described in [1] and extended here, with focus on its success rate for providing feasible trajectories in a useful time, where classical local feedback control methods could fail.

The grasping problem of interest here was treated in [1] as a trajectory planning problem (open loop control), and was solved with gradient-based nonlinear optimization. The related Target motion prediction and robot tracking control tasks, which support the open loop control approach, were partly addressed in [2] and [3] respectively. The planning task at hand results in a highly nonlinear, constrained optimization problem, to account for the typical robot position and velocity box constraints, but also for collision avoidance, which is particularly important due to the possible presence of appendages (solar panels, antennas, etc.) on the debris object. Furthermore, the tumbling motion of the debris object gives rise to a timing issue, due to the fact that a given grasping point on it will appear in front of the robot at irregular time intervals and in different positions and orientations relative to it. Last but not least, the argued necessity for a communication link to ground during the execution of the grasping maneuver may provide a limited operational window for its execution. It is argued here that local control methods [4] [5] may fail in these conditions, if they do not make use of a feasible reference trajectory.

A well known consequence of the presence of nonlinear constraints is that the motion planner developed in [1] might fall into a local minimum, or not converge to a solution at all, for a given task, without a judiciously defined initial guess for its parameters. The free-flying dynamics of the robot, as well as the orbital scenario, present further difficulties for executing the motion planning task, with respect to a fixed-base robot task on ground (see for example [6]), which are described here.

In this paper a method is developed to provide a good initial guess for the motion planner. A suitable mapping is formulated between a four-dimensional input space, which characterizes the Target spinning motion (spinning is considered here for simplicity, although it is argued that the method can be extended to the general tumbling case), and an N-dimensional output space, which represents the family of time-parameterized optimal robot trajectories (N is practically the number of optimization parameters for each of two motion planning subproblems). A look-up table is constructed, with aid of a global search, for a sample range of spinning motions. For this, an extended version of the trajectory planning method presented in [1] is used.

The generated look-up table is then used to perform a statistical analysis of the success rate of the motion planner, assumed to be implemented in a realistic scenario, by providing an initial guess for a set of arbitrary spinning states of the given Target. Simulation results are obtained for a scenario in which the Target has two solar panels (see Fig. 1).

The paper is then structured as follows: a bibliography is first provided in subsection I-A and a problem statement is presented in subsection I-B. Section II describes the constrained optimization problem. Section III addresses the generation of trajectories in a useful time. Section IV analyses the simulation results and section V draws the conclusions.

R. Lampariello and Gerd Hirzinger are with the Robotics and Mechatronics Center (DLR), 82234 Weßling, Germany, email: *first-name.lastname@dlr.de*.

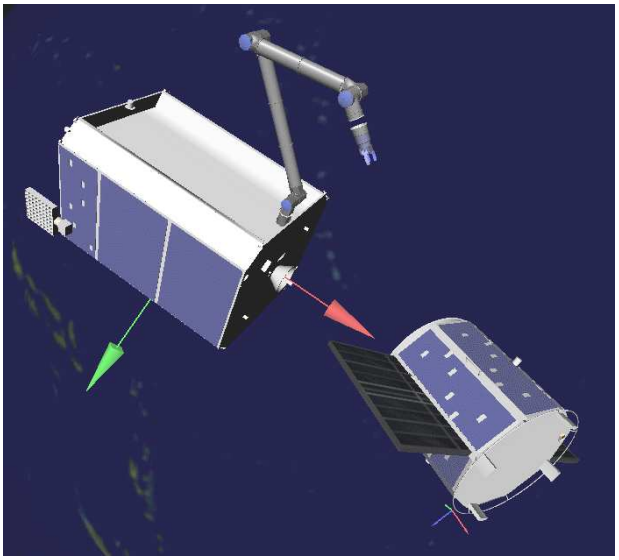

Fig. 1. Orbital scenario: Servicer satellite with 7 DOF manipulator and Target satellite with solar panels. Orbital coordinate system shown: x = V-bar (red); y = out of orbital plane (green); z = -ve R-bar (blue - not visible in figure). One grasping point coordinate system on Target ring structure shown.

A. Related work

A great deal of the work to be found in the literature on the space robot grasping task, which covers a time span of over 15 years, is based on nonlinear feedback control [7] [4], optimal control [5][8] or model predictive control [9]. All these approaches however, do not guarantee a feasible trajectory and require operator intervention. It is in fact evident that there is no simple measure to determine if and when the grasping point will be reachable from the current configuration (see Fig. 1) and whether the trajectory which derives from a local control law will be feasible at all times (accounting for collision avoidance with appendages, kinematic and dynamic singularities, sustainability of necessary robot forces during the stabilization phase which follows grasping, other motion constraints). These methods also do not provide any information on the necessary time synchronization between the motion of the grasping point on the Target and that of the robot. Furthermore, since the methods are local, the nonlinear nature of the robot kinematics is not exploited to favor a successful grasp.

More generally, with regards to robot motion planning in the context of nonlinear optimization, different applications may be found, as for industrial robots [10] [11], for humanoid robots [12] [13] [14] and for flying vehicles [15]. Collision avoidance is treated in different ways, including the method of growth distance in [16] [10] and the method of strict-convexity bounding volumes [17]. The grasping task of interest here was treated in this context in [1].

In [6] the optimal motion planning problem for a catching task with a fixed-base redundant manipulator on ground is addressed, while treating the real-time implementation and the local minima issues. Different learning methods are implemented to map a three-dimensional input space, which represents the Target trajectory (e.g. a ball), and the N-dimensional parameter space of global optimal robot trajectories.

Motion planning for free-flying robots with collision avoidance is treated in [18], where the trajectory generation of the approach phase is addressed, while treating the chaser satellite with robot as a point mass and the target as a rotating rigid body with a large span (to account for the solar panels). It is argued that the major danger is the potential of collision between the satellite and the robot. Optimization is used to find safe kinematical trajectories, while optimizing a safety metric, based on the ”time to collision” in case of robot control failure, as well as fuel expenditure and time. The problem of local minima is recognized but not treated.

B. Problem statement

The grasping and stabilization task may be described as follows:

- the Servicer satellite is at first in its initial position, called the Observation Point;
- the Servicer satellite then approaches its grasping position, called Mating Point, by means of its actuation;
- the robot manipulator on the Servicer then performs a maneuver to bring its end-effector in a vicinity of the grasping point on the tumbling Target;
- the robot end-effector tracks the grasping point for a few seconds with subsequent homing-in and closing of the grasp;
- the relative motion between the Servicer and the Target is stabilized with the robot;
- finally, the motion of the Servicer-Target compound is stabilized by means of the Servicer actuators.

Note that the Servicer is only actuated in the first approach and in the last compound stabilization steps.

We are interested here in considering the scenario shown in Fig. 1. This is because we want to address typical targets in Low Earth Orbit (orbital period 1,5 hours), for which solar panels may be present, but are generally relatively small (1 or 2 meters). The robot manipulator is assumed to have 7 DOF and the mass ratio between Servicer and robot is taken to be significantly small, such that a fixed-base robot control method would generally fail.

We will assume that the tumbling Target presents only one useful grasping point, for the purpose of our analysis. In fact, finding a suitable structure to grasp is a recognized problem (see [19]). A suitable position of the Servicer relative to the Target and a suitable robot trajectory are therefore of primary importance for a successful, collision-free operation.

It is also assumed that the grasping point is predefined by an operator, to relax the degree of autonomy of the proposed

method. The choice is dictated by the geometry of the point to be grasped, as well as its favorable location of the Target. If a model of the Target is not given, a communication link to ground will therefore be necessary during the grasping operation, to allow for the operator to define the grasping point. A communication link is however also necessary to upload the reference trajectory to the robot, which must be computed on ground due to the limited computational power on board, and because it is useful to supervise the grasping operation.

Two operating conditions for communication to ground are possible: in the first, the communication link takes place through a relay satellite in Geostationary Earth Orbit; in the second, communication takes place through a direct link to ground. While in the first case, a link can be assumed possible for at least a half-orbital period, for the second the time-of-contact has a duration of approximately ten minutes. We will assume the latter as the operating condition here.

The tumbling motion is assumed to be constrained to a flat spin, for which the angular velocity is directed about an inertially fixed rotation axis which also coincides with the major axis of inertia of the Target, and limited between +/- 4 deg./sec.. Its orientation however is assumed to be general. The flat spin motion is motivated by the well-known energy dissipation property of flexible appendages.

The Target trajectory, meaning the translational motion of its centre of mass and the rotational motion about it, is assumed to be determined by a motion prediction algorithm (e.g., [2]). This algorithm must also run on ground, due to its computational burden, and requires a motion estimate of the Target motion, which in turn is based on real-time visual data (from stereo camera or LIDAR) of the Targer. A motion planning solution should be delivered by the motion planner within a fraction of the total motion prediction time (in [2] the latter was assumed to be 100 seconds). In this way, the maneuver can be executed within the remaining motion prediction time, thus guaranteeing its feasibility. The notion of ”useful time” used here refers to this fact, to also allow for a synchronized execution.

II. FORMULATION OF THE CONSTRAINED OPTIMIZATION PROBLEM

In this Section the optimization problem which results from the problem formulation described in Section I-B is addressed.

A. Dealing with the short communication time

The operational condition of a limited communication time, added to a general orientation of the spinning Target and a single grasping point on it, led to the idea of developing a strategy after which a feasible trajectory to any possible Cartesian position of the grasping point, within a predefined suitable subregion of space which may be spun by it, can be obtained in a useful time. In fact, since the orientation

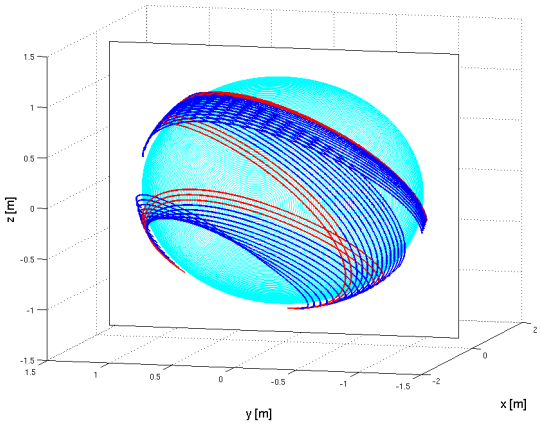

Fig. 2. Trajectory of grasping point shown in Fig. 1 for initial angular velocity [-2 -4 -2] deg/sec and communication link coverage for half-orbit period (blue line) and 8 minutes (red line). Predefined hemisphere subregion shown (in cyan). Reference frame directed as in Fig. 1, centre of mass of Target in coordinate frame origin.

of the Target is initially unknown and is general, nothing can be said *a priori* about the region which the grasping point will span in a single communication window (this issue is addressed in some detail in [20]). This way, as long as the grasping point crosses the chosen subregion within the communication window, a feasible solution, if physically possible, is guaranteed.

We confine the subregion of interest in which favorable locations of the grasping point may occur to a hemisphere, the plane of which is orthogonal to the line connecting the two satellites, as shown in Fig. 2. This idea derives from the simple fact that the Servicer will approach the Target from some direction (generally V-bar or R-bar, defined in Fig. 1. Note that V-bar and R-bar are in the direction of flight and in the orbital radius directions respectively) and will therefore have a limited reaching capability. The figure also shows an example of the curves which a grasping point will trace onto this hemisphere, as the Target tumbles, for a given initial angular velocity and for the two different communication operational conditions. It is important to note that the traces shown are those for one given orientation of the assumed constant angular momentum vector of the Target. Therefore for an arbitrary orientation of the angular momentum, the same traces will result to be rotated about the origin accordingly.

B. Free-flying robot motion planning problem

Given the strategy chosen in Section II-A, we want to now define the optimization problem at hand in more detail. The problem is similar to the one tackled in [1], the salient features of which are reported here, however with some modifications.

The problem is divided into two subproblems: first the approach and tracking with grasping and second, the stabi-

lization (equivalent to bullets two to four and to bullet five in the task description provided in Section I-B respectively).

In the first subproblem, a point-to-point task is first solved, to bring the robot end-effector in the vicinity of the grasping point on the Target. This is followed by a short tracking of the grasping point, while homing in of the end-effector onto it, and finally closing the grasp.

Mathematically, the optimization problem for the robot approach phase (denoted with the upper superscript 1) can be written as follows:

$$\min_{t_0^1, t_f^1, \boldsymbol{\theta}_b{}^1(t_0^1), \boldsymbol{\theta}_m{}^1(t)} \Gamma^1(t_0^1, t_f^1, \boldsymbol{\theta}_b{}^1(t_0^1), \boldsymbol{\theta}_m{}^1(t)) \tag{1}$$

subject to

$$\mathbf{M}(\boldsymbol{\theta}^1)\, \ddot{\boldsymbol{\theta}}^{\mathbf{1}}(t) \;+\; \boldsymbol{c}(\boldsymbol{\theta}^1, \dot{\boldsymbol{\theta}}^1)\, \dot{\boldsymbol{\theta}}^{\mathbf{1}}(t) \;=\; \boldsymbol{\tau}^1 \tag{2}$$

$$\mathbf{h}^1(\boldsymbol{\theta}^1(t)) \leq 0 \tag{3}$$

$$\mathbf{h}^1{}_{coll}(\boldsymbol{\theta}^1(t)) \leq 0 \tag{4}$$

$$\mathbf{g}^1(\mathbf{r}^e({t_0}^1 + {t_f}^1)) = 0 \tag{5}$$

$$\boldsymbol{\theta}^1(t_0^1) = \boldsymbol{\theta}_{in}^1, \dot{\boldsymbol{\theta}}^1(t_0^1) = 0, \dot{\boldsymbol{\theta}}^1({t_0}^1 + {t_f}^1) = 0 \tag{6}$$

for ${t_0}^1 \leq t \leq {t_0}^1 + {t_f}^1$ and where $\boldsymbol{\theta} = [\boldsymbol{\theta}_b^T \;\; \boldsymbol{\theta}_m^T]^T$ expresses the degrees of freedom of the system, including those of the Servicer with subscript b and those of the robot manipulator with subscript m. Furthermore, ${t_0}^1$ is an initial time, ${t_f}^1$ is a predefined relative final time (added to ${t_0}^1$), Γ^1 is a predefined cost function, $\mathbf{h}^1$ are inequality box constraints of type $\mathbf{x}_{min} \leq \mathbf{x}(t) \leq \mathbf{x}_{max}$, for $\mathbf{x} = \{\boldsymbol{\theta}^1, \dot{\boldsymbol{\theta}}^1\}$ and $\mathbf{h}^1{}_{coll}$ are collision avoidance constraints. Note that the constraints on the robot joint positions are purposely reduced to 75% of their true values, in order to leave some margin for the stabilization task. Those on the velocities help to ensure that the solutions are free of singularities. To compute collision detections and to formulate the collision avoidance problem, bodies in the scene are represented as convex polytopes [1]. The relative inequality constraints then consist in ensuring that the penetration depth between the polytopes is always zero.

Functions $\mathbf{g}^1(\mathbf{r}^e({t_0}^1 + {t_f}^1))$ are equality constraints on the final end-effector state $\mathbf{r}^e({t_0}^1 + {t_f}^1)$ to be in a desired relative position and orientation with respect to the grasping point. Differently as in [1], this equality constraint is relaxed here in one rotational direction to an inequality constraint, to allow for a rotation of the end-effector around the ring structure of the Target. Finally (6) expresses boundary conditions on position (where $\boldsymbol{\theta}_{in}$ is a predefined feasible and singularity-free initial configuration) and on velocity. More will be said about boundary conditions on acceleration and jerk in Section III-A.

In order to solve the problem posed by the strategy chosen in Section II-A, the initial time ${t_0}^1$ is fixed and the spinning motion of the Target is propagated in such a way that the Cartesian position of the grasping point is the desired one at that same time. In order to achieve this, the desired position of the grasping point is defined by the first two of the three Cardan angles, $\boldsymbol{\phi}_{des}$=$[\phi_{1\ des}\ \phi_{2\ des}\ \phi_{3\ des}]^T$, which define the orientation of the Target with respect to the inertial frame:

$$\mathbf{r}^{GP\,I} = \mathbf{A}(\boldsymbol{\phi}_{des})\mathbf{r}^{GP} \tag{7}$$

where $\mathbf{r}^{GP}$ are the coordinates of the grasping point in the Target body frame, $\mathbf{A}(\boldsymbol{\phi}_{des})$ is the transformation matrix between the Target body frame and the inertial frame, function of the Cardan angles $\boldsymbol{\phi}_{des}$, and $\mathbf{r}^{GP\,I}$ is the desired Cartesian position of the grasping point in the inertial frame. Note that this parameterization of the Target orientation is only used to span the subregion of interest shown in Fig. 2 (quaternions are used to propagate its motion in time). As such, parametric singularities do not pose any problem in this context.

In order to guarantee a feasible solution, the Servicer maneuver which precedes the robot approach maneuver, from the predefined initial position (Observation Point) to the found solution for the beginning of the robot approach maneuver (Mating Point), expressed by $\boldsymbol{\theta}_b{}^1({t_0}^1)$, is also subjected to velocity and collision avoidance constraints, expressed in (3) and (4). Note that this feature was absent in [1]. Note also that $\boldsymbol{\theta}_b{}^1({t_0}^1)$ only includes the two translational position components in the orbital plane, where it is imposed that all other degrees of freedom remain fixed during this maneuver [1].

The tracking phase consists of an inverse kinematics solution, which is dictated by the motion of the Target and the time at which it begins [1]. Its duration is defined by a relative time ${t_f}^2$. The same motion constraints apply for this phase as for the previous. In order to provide a reference final configuration for the following stabilization phase, the tracking phase is propagated further than ${t_f}^2$, to ${t_f}^{2+}$, to the robot configuration $\boldsymbol{\theta}_m{}^{2+}({t_0}^1 + {t_f}^1 + {t_f}^{2+})$.

The stabilization phase requires bringing the robot joints velocity to zero, while ensuring that the motion constraints are not violated:

$$\min_{t_f^3, \boldsymbol{\theta}_m{}^3(t)} \Gamma^3(t_f^3, \boldsymbol{\theta}_m{}^3(t)) \tag{8}$$

subject to

$$\boldsymbol{\theta}({t_0}^3) = \boldsymbol{\theta}({t_0}^1 + {t_f}^1 + {t_f}^2), \dot{\boldsymbol{\theta}}({t_0}^3) = \dot{\boldsymbol{\theta}}({t_0}^1 + {t_f}^1 + {t_f}^2) \tag{9}$$

$$\boldsymbol{\theta}_m({t_0}^3 + {t_f}^3) = \boldsymbol{\theta}_m({t_0}^1 + {t_f}^1 + {t_f}^{2+}), \dot{\boldsymbol{\theta}}_m({t_0}^3 + {t_f}^3) = 0, \tag{10}$$

$$\mathbf{h}^3(\boldsymbol{\theta}^3(t)) \leq 0 \tag{11}$$

$$\mathbf{h}^3{}_{coll}(\boldsymbol{\theta}^3(t)) \leq 0 \tag{12}$$

where Γ^3 is a predefined cost function. The inequality in (11) may here also express limits on the forces on the robot gripper. The initial conditions in (9) express the dependency on the final conditions of the previous tracking phase.

C. Cost functions for the motion planning problem

Different cost functions can be defined for the approach phase, as for example the end time [5], or the robot manipulabitily [1], the distance from collisions [18] [1], or the Servicer actuation energy [1]. The first is however not relevant in the context of this work, due to the introduction of a synchronization between the Target and the robot. The others either express an attempt to maximize the success rate of the grasping maneuver or to perform a perhaps not indispensable energy optimization.

A different approach is developed here, after which the minimization of a cost function for the task in question is of secondary importance, since it is really only important to have a high success rate. Instead, the definition of a cost function, at least for the approach phase, is here dictated by a necessity which results from the chosen method of solution for the optimization problem at hand. In fact, given that the joint states are parameterized in time (see Section III-A) with a sufficiently high number of parameters to ensure an efficient convergence rate of the optimizer (see Section IV), it is necessary to introduce a cost function which provides physically sensible solutions (in practice, avoids unnecessarily high joint speeds and oscillations, which can easily result from a parameterization with a high number of parameters). One way to do this is to choose the mechanical energy of the robot manipulator as cost function:

$$\Gamma_{\text{energy}} = \int_{t_0^1}^{t_0^1+t_f^1} (\boldsymbol{\tau}^T(t)\, \dot{\boldsymbol{\theta}}(t))^2 \, dt \,, \tag{13}$$

which represents the integral of the mechanical power of the robot joints. This cost function is sufficient to regulate the robot motion solutions which result from the planner. Note that at the same time, since the mechanical energy is minimized, and with it the joint velocities, also the distance from a singularity, and as a result the manipulability, may indirectly be maximized. Note also that extra equality constraints need to be introduced onto the Servicer position, such that the latter remains fixed in the origin of the orbital (inertial) frame. Without these extra constraints, solutions would otherwise be found for which the robot hardly moves (i.e. $\Gamma_{\text{energy}} = 0$) and the Servicer brings the end-effector onto the grasping point. As we will see later (see Section III-E), a Servicer maneuver can still be utilized to reach any Cartesian position of the grasping point on the Target which is not reachable without one. In the process of planning such maneuver, the Servicer actuation energy can be optimized.

A second cost function is introduced here, related to the robot manipulability, which penalizes joint positions which are closest to their limit values:

$$\Gamma_{\text{margin}} = \sum_{i=1}^{DOF} -k\, \theta_{delta\,i} + (\theta_{\,i} - \theta_{mid\,i}) \tag{14}$$

where $\theta_{mid\,i}$ and $\theta_{delta\,i}$ are the middle and relative amplitude for the position of the i^{th} joint and k determines the threshold above which the positions are penalized. The terms in (14) are set to zero if negative. Furthermore, only the maximum value for each joint throughout the maneuver is taken for the cost computation. This cost function will be used to prioritize the solutions resulting from the global search (see Section III-B).

Regarding the cost function of the stabilization phase, this is of marginal importance. As we will see, this phase is relatively straightforward.

III. TRAJECTORY PLANNING IN A USEFUL TIME

In this Section, the method of solution for the optimization problem at hand is first addressed. The method for generating a look-up table and its utilization in a real-time scenario is then described.

A. Parameterization of the trajectories

The chosen parameterization for the joint states is an order-4 B-spline. This was chosen in order to allow for smoothness up to the third derivative. Details of the B-spline implementation can be found in [6].

Important to note here is the particular choice of the boundary conditions. These were already defined for position and velocity in Section II-B. It as also chosen here to set acceleration and jerk to zero at the boundaries, such as to minimize the number of optimization parameters and therefore minimize the computational time of the optimizer.

For the Servicer maneuver a trapezoidal function is used. Details of this may be found in [20]. The resulting number of parameters is one per state, for given maximum spacecraft velocity and acceleration. As described in Section II-B, only two translational states are used for the maneuver.

The resulting number of parameters for the first subproblem, i.e. the approach, is N, where: N-3 parameters belong to the parameterization of the B-splines ((N-3)/7 per joint), two parameters belong to the parameterization of the Servicer maneuver and one further parameter is introduced to represent the time along the Target trajectory at which the grasping takes place. This last parameter dictates the position of the Target grasping point at time ${t_0}^1+{t_f}^1+{t_f}^2$, in function of the given Target dynamics.

Note that for the second subproblem, the stabilization, the last three parameters are irrelevant and are therefore not included.

B. Global search

Given the highly nonlinear nature of the optimization problem at hand, local minima are present as well as multiple solutions, of any given problem. As such, we run the optimization for a given grasping point position and spin velocity 100 times, using different initial guesses for the starting parameters, chosen with the following procedure:

a robot configuration $\boldsymbol{\theta}_m$ is defined randomly, within the range of allowed values; a trajectory is determined as a straight line between the given initial and the randomly defined configuration, by algebraic computations of the B-spline parameters; these latter parameters are taken as initial guess for the robot states. For the two Servicer variable position states, random values are chosen between predefined maximum and minimum values. Subsequently, the starting parameters which yield the best optimization result of the 100 trials, in terms of the cost function Γ_{margin}, are taken as global optimium.

C. Offline method for local optimization problem

The optimization problem described above is solved as a nonlinear programming problem (NPL), by satisfying the equality and inequality constraints at a finite number of k via points. The proposed optimization method is based on direct single shooting, with parameterization of the system independent states in time, as done for e.g. in [10], i.e., $\boldsymbol{\theta} = \boldsymbol{\theta}(t, \mathbf{p})$, where $\mathbf{p}$ is a column matrix containing N-3 optimization parameters, as described in Section III-A. For the approach and for the stabilization problems described in Section II-B, two parameter sets $\mathbf{p}^1$ and $\mathbf{p}^3$ result. The system dependent states and input forces are then computed from the integration of the state transition equations [1]. Note that due to the possibility of relatively large joint velocities arising from the chosen parameterization, care needs to be taken to ensure that the integration accuracy is sufficiently high, such that the momentum of the free-floating system throughout the integration is constant (if not, significant errors in the end-effector position result). The NPL is solved with the Sequential Quadratic Programming algorithm from the MOPS library [21]. The numerical integration is performed with an explicit fifth order Runge-Kutta method, with step size control (DOPRI5).

To compute the penetration depth between two bodies, the ODE library is used [22]. The library allows representing objects as boxes or capsules. Each pair of intersecting objects is treated separately and penetration depth can be evaluated for each pair straightforwardly.

D. Efficient initialization of the local planner

To provide a good initial guess for the trajectory planning problem in a real-time scenario, in which the grasping point follows a trajectory provided by a motion prediction algorithm, the solutions of the global search are used in a form of a look-up table (first-order Nearest Neighbour (NN)). The spinning motion of the predefined grasping point on the Target is parameterized with four parameters, which include three parameters of the satellite orientation (the Cardan angles $\boldsymbol{\phi}_{des}$ defined in Section II-B) and one for the spin component of its angular velocity (note that a flat spin is always directed about the major axis of inertia). Note that we are only interested in the satellite orientation parameters which result in the region of interest shown in Fig. 2. By setting up a grid for these, a global search is performed for each grid point and the resulting parameter sets $\mathbf{p}^1$ and $\mathbf{p}^3$ stored in it. A mapping between the four input parameters and the N output parameters results at the grid points, for the complete grasping task. This grid needs to be computed only once, for a given target geometric model, and is computed off-line, therefore not affecting the autonomous character of the proposed grasping strategy.

E. Real-time implementation

In the real-time setting, it is assumed that a prediction of the Target motion is available. For a given set of the four input parameters which describe the Target trajectory, a simple Euclidean metric is used to identify the closest point in the look-up table. This point is used to define the initial guess for the optimizer. Subsequently, based on this initial guess, the motion planner is run on-line to compute a successful robot trajectory. For this on-line version of the motion planner the accuracy of the optimizer is reduced, in order to minimize its computational time. The on-line planner however still satisfies the equality and inequality constraints defined in Section II-B, therefore adjusting the discrepancy between the trajectory which results from the initial guess and the desired trajectory.

If the Target trajectory does not cross the subspace represented by the look-up table during the prediction time of 100 seconds, the observation and prediction procedure is repeated until it does, i.e. until the grasping point is in reach. Alternatively, an extra Servicer maneuver could be planned, such that the closest point on the trajectory is brought in the vicinity of the relative closest point in the look-up table. This maneuver could be optimized for actuation energy, independently of the robot maneuver. This procedure however would require an overall longer motion prediction time, due to the long execution time of the Servicer maneuver (in the order of 200 seconds for a Target size as the one considered here [20]).

IV. ANALYSIS OF RESULTS

In performing first global search computations, it was found that a parameterization of the joint states with one single parameter per joint gave a very low convergence rate. It was instead found that more than two parameters per joint did not make a difference. As a result, in this implementation N=17, to account for two parameters per robot joint, two parameters for the position of the Servicer at the beginning of the robot approach phase (in the orbital plane) and one parameter to determine the time along the Target trajectory for the grasping time. Note however, that to generate the look-up table the last three parameters are fixed by the motion constraints which define the optimization problem: the Servicer position is constrained to be fixed in the Observation Point, to avoid that the energy cost function goes

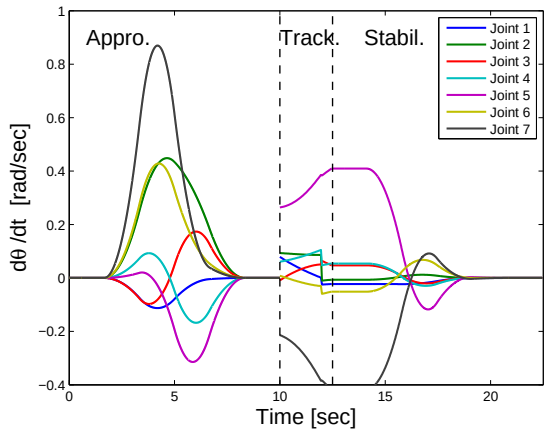

Fig. 3. Robot joint velocity profile for an on-line solution. Three phases approach, tracking and stabilization shown. Maximum allowed joint velocity: 1.0 rad/sec.

to zero, as described in Section II-C; the time along the Target trajectory is fixed, due to the necessity of the Target being in the desired Cartesian position, at the gasping time, which represents the point in the Look-up table to be computed.

The value for the integrator accuracy was set to 1e-9, that of the optimizer to 1e-2, while its delta for the computation of the Jacobian to 1e-5. The number of via points was set to k=20. The inequality constraint on the end-effector orientation relative to the Target was defined such as to allow a +/-75 deg. rotation.

The energy cost function in (13) provides the expected bell-shaped joint velocity profiles for the approach phase as shown in Fig. 3. In the same figure a typical evolution of the joint velocities is shown for the tracking and stabilization phases. The jump in the velocity between the approach and the tracking phases is due to the fact that the boundary conditions for the end of the approach phase are set to zero robot joint velocities (see (6)). This is not thought to have any negative operational consequences, unless flexible appendages are present on the Servicer, in which case a smooth transition could be achieved via redefinition of the same boundary conditions or with a PD control term in the tracking phase, with a zero velocity at its beginning. Finally, the satellites and robot were modeled with 13 polytopes (boxes and capsules) and gave rise to a total of 15 collision checks for every via point. The span of the solar panels together was taken to be 3.2 meters.

A. Generation of look-up table

In order to generate an exhaustive set of grid points to cover the hemisphere described in Section II-A, equidistant points on a sphere should ideally be computed A simpler approach was adopted here and a set was defined by hand, which resulted in the 35 points shown in Fig. 4. Each point in this figure represents a Cartesian position of the grasping point at the grasping time (end of tracking phase). This position was parameterized here as described in Section II-B, by means of the first two of the Cardan angles ϕ_{des}, which define first a rotation about the z-axis and then one about the new y-axis. A preliminary rotation is however necessary to align the position vector of the grasping point with the negative Cartesian x-axis. The third Cardan angle was then used to rotate the Target about the same position vector (the new x-axis, which therefore does not influence the Cartesian position of the grasping point), to complete the orientation representation of the Target. The incremental steps of the third angle were taken to be 45 degrees, to cover a complete 360 rotation in 8 steps. Finally, the angular velocity, which is specified to have a maximum magnitude of 4 deg/sec, was varied for each of the resulting points (35x8), between the two values +4 deg/sec and -4 deg/sec..

This task parameterization so described results in a total of 560 (=35x8x2) grid points. For demonstration purposes, only four adjacent points were considered here, also shown in Fig. 4, which had the following values for the first two Cardan angles: (60, 45), (30, 45), (30,0) and (60,0) deg.. For the resulting 64 grid points (4x8x2) a global search was performed, as described in Setion III-B, to obtain their corresponding optimization parameters for the approach and stabilization maneuvers, $\mathbf{p}^1$ and $\mathbf{p}^3$. Note that the following parameters were set to the values: $t_0^1 = 0, t_f^1 = 10.0, t_f^2 = 2.5, t_f^3 = 10.0$ sec..

The convergence rate varied from point to point, between zero (e.g. due to collisions with the solar panels) and 20%. Of these, the best was selected according to their corresponding value for the cost function Γ_{margin} defined in (14).

The computation times of this global search was ap-

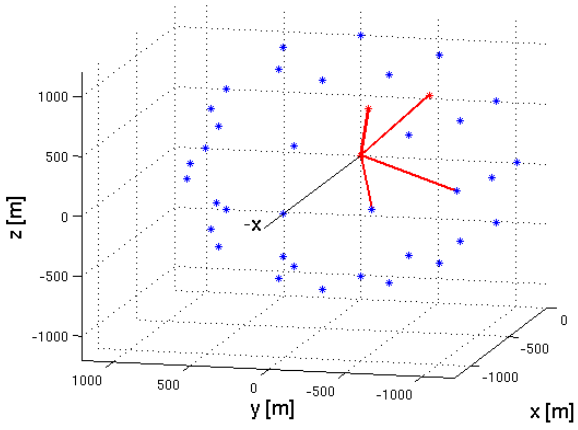

Fig. 4. Shown is the chosen set of grid points (in blue) to represent the hemisphere region defined in Fig. 2 as well as the four adjacent sample points used for the statistical analysis, defined by the first two Cardan angles: (60, 45) (in red), (30, 45) (in red), (30,0) and (60,0) deg.. The grasping point position vector is shown for each sample point in red. Origin in Target centre of mass.

proximately 30 hours per Cartesian point (1400 tasks) for the approach phase and 2 hours for the stabilization phase (only performed for the runs of the approach phase which converged), on an Intel Xeon CPU W3520 2.67GHz machine. Note that for a fully tumbling Target, the discretization of the angular velocity would have to include three components, resulting in 560x2x2=2240 points for a complete set. This computationally intensive task however needs to be computed for a given free-flying robot and a given Target geometry (given grasping point) only once, such that a wide range of tumbling states would be covered during the grasping task.

B. Real-time implementation

A statistical analysis was performed to assess the success rate of the trajectory planner when supported by the look-up table. Random values of the three Cardan angles were chosen within the range defined by the four sample points defined in the previous Section for the first two, between 0 and 45 for the third, and between +4 and -4 deg/sec. for the rotational velocity.

By choosing the first solution found in the global search, the success rate (for 100 trials) was found to be 75%, of which 83% for the approach and tracking phase and 92% for the stabilization phase. When however the solutions found in the global search were ordered after the cost function Γ_{margin}, such that the best one was used, then the success rate was found to be 90%, of which 92% for the approach and tracking phase and 98% for the stabilization phase.

The search strategy could be improved by looking for the point in time along the given trajectory which is closest to a grid point in the look-up table (in the analysis presented here the time along the Target trajectory used for the selection of the closest point in the look-up table was fixed). Note also that the results can always be improved by introducing more grid points in the look-up table, however at the expense of its computation time. The latter however can be substantially improved by performing the optimization gradient computations in parallel. Note in fact that the optimization gradient computations are currently computed numerically by finite differences.

The average computation times for the real-time runs was 102 seconds, however 10% with a time above 150 seconds. Further work will aim at reducing this computation time.

V. CONCLUSION

A method is described which provides feasible trajectories for a free-flying robot to grasp a target debris object in a state of pure spin about an inertially fixed axis and with general orientation, in a useful time. The method allows to deal with communication constraints typical of a Low Earth Orbit scenario. All relevant three-dimensional motion constraints are accounted for, such as collision avoidance and robot workspace boundaries. The method is based on the construction of a suitable lookup table off line, to provide a good initial guess for the trajectory planner. The method is also extendible to a tumbling motion of the target object. It is shown by statistical analysis, for a range of spinning states, that the method has a sufficiently high success rate.

REFERENCES

[1] Lampariello, R.: "Motion Planning for the On-orbit Grasping of a Non-cooperative Target Satellite with Collision Avoidance", i-SAIRAS 2010, Japan, Aug. 2010.

[2] Hillenbrand, U., Lampariello, R.: "Motion and Parameter Estimation of a Free-Floating Space Object from Range Data for Motion Prediction", i-SAIRAS 2005, Germany, Sept. 2005.

[3] Abiko, S. Lampariello, R. and Hirzinger, G.: "Impedance Control for a Free-Floating Robot in the Grasping of a Tumbling Target with Parameter Uncertainty", IROS 06, China, October 2006.

[4] Yoshida, K., *et al*: "On the Capture of Tumbling Satellite by a Space Robot", IROS 06, China, October 2006.

[5] Aghili, F.: "Optimal control for robotic capturing and passivation of a tumbling satellite with unknown dynamics", AIAA Guidance, Navigation and Control Conference and Exhibition, Honolulu, Hawaii, USA, 2008.

[6] Lampariello, R., Nguyen-Tuong, D., Castellini, C., Hirzinger, G., Peters, J., "Trajectory planning for optimal robot catching in real-time", ICRA 2011, China, May 2011.

[7] Papadopoulos, E., Moosavian, S.: "Dynamics and Control of Multi-arm Space Robots During Chase and Capture Operations", IROS 94, pp. 1554-1561, 1994.

[8] Aghili, F.: "Optimal control of a Space Manipulator for Detumbling of a Target Satellite", ICRA 09, Kobe, Japan, May, 2009.

[9] Henshaw, C.G., *A Unification of Artificial Potential Function Guidance and Optimal Trajectory Planning*, Mountain Guidance and Control Conference, Colorado, USA, 2005.

[10] Park, J., Bobrow, J.E.: *Minimum-Time Motions of Manipulators with Obstacles by Successive Searches for Minimum-Overload Trajectories*, ICRA 02, Washington, USA, May 2002.

[11] Stryk, O. von, M. Schlemmer: "Optimal control of the industrial robot Manutec r3". In: R. Bulirsch, D. Kraft (eds.): Computational Optimal Control. International Series of Numerical Mathematics 115 (Basel: Birkhuser, 1994) 367-382.

[12] S. Miossec, K. Yokoi and A. Kheddar, "Development of a software for motion optimization of robots - Application to the kick motion of the HRP-2 robot", *Proceedings of IEEE International Conference on Robotics and Biomimetics*, 2006.

[13] Harada, K., Hauser, K., Bretl, T., Latombe, J.P.: "Natural Motion Generation for Humanoid Robots", IROS 06, China, Oct. 2006.

[14] Werner, A., Lampariello, R., Ott, C., "Optimization-based generation and experimental validation of optimal walking trajectories for biped robots, IROS 12, Portugal, October 2012.

[15] Milam, M.B., Mushambi, K., Murray, R.M.: "A New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems", *39^{th} IEEE Conference on Decision and Control*, Sydney, Australia, December 2000.

[16] Ong, C.J., Gilbert, E.: "Robot Path Planning with Penetration Growth Distance", *Journal of Robotic Systems*, 15(2), 57-74, 1998.

[17] A. Escande, S. Miossec and A. Kheddar, "Continuous gradient proximity distance for humanoids free-collision optimized-postures", *Proceedings of International Conference on Humanoid Robots*, 2007.

[18] Jacobsen, *et al*, "Planning of Safe Kinematic Trajectories for Free-Flying Robots Approaching an Uncontrolled Spinning Satellite" ASME 2002 Conference, Montreal, Canada, 2002.

[19] De Peuter, et al: "Satellite Servicing in GEO by Robotic Service Vehicle", ESA Bull 78, 1993, pp 33-39.

[20] Lampariello, R.: "On Grasping a Tumbling Debris Object with a Free-Flying Robot (invited paper)", 19th IFAC Symposium on Automatic Control in Aerospace 2013.

[21] Joos, Hans-Dieter (2008) MOPS Multi-Objective Parameter Synthesis User's Guide V 5.3. DLR-Interner Bericht. DLR IB 515-08-37.

[22] http://opende.sourceforge.net/wiki/index.php/Manual_(Collision_Detection)