Sensitivity Analysis for an On-Orbit Robotic Approach Task

Caroline Specht[1,2], Roberto Lampariello[2], and Alin Albu-Schäffer[2]

***Abstract*— In application areas such as spacecraft guidance and robotics, optimal control typically requires feasibility with respect to motion constraints, in addition to optimality. Inevitably, the question arises of how to manage uncertainty in related task parameters. Via the Sensitivity Theorem, solutions of a given parametric optimal control problem can be approximated through a fast online sensitivity-based update of a given nominal solution for perturbed task parameter values. Using the presented estimation of the neighborhood of validity of this approximation, a region of task parameter space – called the task workspace – can be developed, within which it is possible to provide admissible solutions that are provably robust to task uncertainty. The method is demonstrated in simulation for an on-orbit robotic approach task, which precedes the capture of a target satellite. The non-holonomic kinematically redundant free-floating space robot executes the task under parametric uncertainty of the final six-dimensional pose of the end-effector. This simulation shows that the task workspace is a useful real-world tool for space mission design. The computational efficiency of the solution update within the approximated neighborhood is also demonstrated.**

I. Introduction

Trajectory planning and control of robotic maneuvers is a rich and varied field. The robots come in many forms, and various tactics are used in planning their motion - including convex [1] and non-convex [2] [3] optimization-based methods. In many situations, it is beneficial to track a pre-planned trajectory, see [3]–[5] and sources within. Invariably, the questions arise of how to robustify the solution and execution of a motion plan to uncertainties in the system and/or environment model upon which the plan is based [2] [3], and how large the deviation from the motion plan may provably be for the recovery to remain feasible [3] [6].

At the motion planning stage, these challenges are often tackled using online replanning, supported by machine learning based warm starting [7] [8] or other optimal control based methods [5] [9]. The chosen approach frequently involves the sensitivity analysis (SA) of the solution space of a parametric nonlinear program (NLP($\boldsymbol{p}$)), where $\boldsymbol{p}$ is the task parameter. These problems are typically highly constrained, nonlinear in nature, and non-convexifiable [1]. Existing online replanning methods do not provide guarantees of feasibility relative to the task uncertainty.

It was shown in [9] that, given a solution to the optimization problem at some nominal value of a task parameter $\hat{\boldsymbol{p}}$, it is possible to approximate a neighboring solution to the problem within a "small" neighborhood of $\hat{\boldsymbol{p}}$, based on the sensitivity of the nominal solution to changes in $\boldsymbol{p}$. This approximation can be used to re-plan a feasible reference trajectory – to be tracked online by a tracking controller – and drastically reduces the computation expense of resolving an updated constrained motion planning problem under perturbed conditions [6].

This robust control methodology, commonly referred to in literature as open-loop, real-time optimal control via sensitivity analysis [5], has been applied to several sample problems. Motion planning for large and complex systems are addressed in [10]–[12]. Consideration has been given to robotics tasks in [6], [13], and [14], and to simplified spacecraft maneuvers in [11].

Subsequent works [15] and [16] emphasized that the alternative to the sensitivity-based update is to re-evaluate the NLP at a given perturbed $\boldsymbol{p}$. In this literature, an attempt is made to circumvent excessive error in the updated trajectory using iterative correction methods, but do not account for the neighborhood of $\hat{\boldsymbol{p}}$ in which the update is valid.

In [6], a method was proposed to estimate the size of the neighborhood of the task parameter $\boldsymbol{p}$ within which it is valid to use the sensitivity-based approximation. The proposed estimate can be applied on a task parameter space such that soft guarantees can be placed on the existence of feasible and optimal solutions throughout it. Furthermore, unlike in nonlinear regression-based methods [7] [8], the method of [6] provides solutions with theoretically provable robustness.

The work presented here addresses the open questions of the scalability of the method presented in [6] to a task parameter space of greater dimension and to a greater number of robot DOF, emphasizing its real-world applicability. The motion planning for a realistic on-orbit robotic capture, based on a realistic model of a seven-joint free-floating robot as a demonstrator [2], is analyzed. This task involves the resolution of a highly constrained NLP that depends on the quality of *in-situ* target motion parameter estimation [3]. The sensitivity-based method from [9] is used to approximate optimal trajectories of the robot manipulator, operating in three-dimensional space, to perturbed final end-effector pose task parameters. For purpose of methodological analysis, the target satellite is assumed in the presented simulation to have angular velocity $\boldsymbol{\omega}_{\mathrm{T}} = 0$ [rad/s], but the presented theory applies generally for any $\boldsymbol{\omega}_{\mathrm{T}}$. The method derived in [6] is used to determine the neighborhood within which the nominal solution of the given parametric NLP can be used to approximate an updated trajectory with provable feasibility as a result of changes in the task parameter. A dynamic and efficient exploration of the task parameter space is proposed - improving on the equidistant grid and static neighborhood size approximation of [6]. A methodological comparison of the sensitivity-based update to cold and warm start optimization-based re-planning is then presented.

Finally, this work introduces the concept of a task parameter workspace, or task workspace. The task workspace

 [1]Caroline Specht is a PhD candidate. [2]The authors are affiliated with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Münchener Str. 20, 82234 Wessling, Germany, email: firstname.lastname@dlr.de

specifies the admissible task parameters which are utilizable in the mission scenario and indicates where the robustness guarantees provided by the method of [6] apply. The analysis of the task workspace is demonstrated to be useful in space mission operations design as well as in mission verification and validation (V&V) procedures [17], expanding on the discussions of requirement satisfaction and robustness of optimal control-based planning methods in [18] [19] to address non-convex methods.

The remainder of this paper is structured as follows: In Sec. II, the theoretical basis for the proposed analysis is reviewed. In Sec. III, the demonstration scenario is described and the optimal control problem is formulated for a general target angular velocity $\boldsymbol{\omega}_{\mathrm{T}}$. In Sec. IV, the methodology is demonstrated in simulation with $\boldsymbol{\omega}_{\mathrm{T}} = 0$ and the results are discussed. Conclusions are drawn in Sec. V.

Notation: Variable scalars are italicized (x or X); fixed scalars are unformatted (x); vectors are in bold italics ($\boldsymbol{x}$ or $\boldsymbol{X}$); matrices are unformatted capital letters (X); and doublestruck letters ($\mathbb{X}$) are sets. Left superscripts indicate a reference frame and the absence of one indicates the inertial frame I. Right subscripts specify the body or quantity to which the variable refers. A hat indicates a nominal value. A rotation from reference frame a to b is indicated by ${}^{\mathrm{ba}}A(\cdot)$, $A(x_y)$ is the rotation of x deg about axis y, and the Cardan angles parameterizing a given rotation matrix are indicated by $\phi_{321}(\cdot)$.

II. Motion and Mission Planning via SA

Consider an NLP($\boldsymbol{p}$), which is: parametric in $\boldsymbol{p} \in \mathbb{R}^{\mathrm{n_P}}$; on a given time interval $\mathrm{t}_0 \leq t \leq \mathrm{t}_\mathrm{f}$, discretized into n uniform steps, or via points, such that $\delta\mathrm{t} = (\mathrm{t}_\mathrm{f} - \mathrm{t}_0)/(\mathrm{n}-1)$; and is of the general form

$$\begin{aligned} &\min_{\boldsymbol{z}\in\mathbb{R}^{\mathrm{n_z}}} \; J(\boldsymbol{z},\boldsymbol{p}) \\ &s.t.\; G_i(\boldsymbol{z},\boldsymbol{p}) \leq 0, i = 0,...,\mathrm{n_G} \\ &\quad H_j(\boldsymbol{z},\boldsymbol{p}) = 0, j = 0,...,\mathrm{n_H}, \end{aligned} \tag{1}$$

where $\boldsymbol{z}$ is the solution in terms of a judiciously selected parameterization of the robot state; J is the discretized objective function; $\boldsymbol{G}_i := [g_{i,k}(\boldsymbol{z},\boldsymbol{p}), k = 0:n]^\top$ are the discretized inequality constraints; $\boldsymbol{H}$ are the discretized equality constraints; and $\mathrm{n_G}$ and $\mathrm{n_H}$ are respectively the number of inequality and equality constraints. The NLP($\boldsymbol{p}$) possesses a set of admissible solutions $\mathbb{S}(\boldsymbol{p})$. Each admissible solution $\boldsymbol{z}$ has a set of active inequality constraints $\mathbb{A}(\boldsymbol{z},\boldsymbol{p})$.

In the course of planning the robotic motion of a space mission, it may be desired to know the optimal solution for any value across a sampled range of $\boldsymbol{p}$. It is very computationally intensive to determine an optimal solution for the problem in (1) for any given value of $\boldsymbol{p}$, and is a concern as the scale of the optimization problem grows [2].

The results permitting the approximation of the optimal solution of the NLP under a perturbed parameter are discussed in the sequel. First, a key property is defined for reference:

Definition 1: Strongly Regular Optimal Solution [5]
The solution $\boldsymbol{z}$ of the NLP($\boldsymbol{p}$) is a strongly regular local solution if:
(a) the solution $\boldsymbol{z}$ is admissible,
(b) the gradients $\nabla_{\boldsymbol{z}} G_i(\boldsymbol{z},\boldsymbol{p})$, $i \in \mathbb{A}(\boldsymbol{z},\boldsymbol{p})$, and $\nabla_{\boldsymbol{z}} H_j(\boldsymbol{z},\boldsymbol{p})$, $j = 1,...,\mathrm{n_H}$, are linearly independent,
(c) the Karush—Kuhn—Tucker (KKT) conditions are satisfied at $(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\lambda})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ are the Lagrange multipliers for $\boldsymbol{G}$ and $\boldsymbol{H}$, respectively,
(d) the strict complementary condition

$$\mu_i - G_i(\boldsymbol{z},\boldsymbol{p}) > 0, \forall i = 1,...,\mathrm{n_G} \tag{2}$$

holds, and
(e) $\boldsymbol{d}^\top \mathrm{L}''_{\boldsymbol{zz}}(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\lambda},\boldsymbol{p})\boldsymbol{d} > 0$ for all $\boldsymbol{d} \in \mathbb{T}_\mathrm{C}(\boldsymbol{z},\boldsymbol{p})$ with $\boldsymbol{d} \neq \boldsymbol{0}_{\mathrm{n_z}}$, where $L(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\lambda},\boldsymbol{p}) := J(\boldsymbol{z},\boldsymbol{p}) + \boldsymbol{\mu}^\top \boldsymbol{G}(\boldsymbol{z},\boldsymbol{p}) + \boldsymbol{\lambda}^\top \boldsymbol{H}(\boldsymbol{z},\boldsymbol{p})$ is the Lagrange function of the NLP($\boldsymbol{p}$) and

$$\begin{aligned} \mathbb{T}_\mathrm{C}(\boldsymbol{z},\boldsymbol{p}) := \{\boldsymbol{d} \in \mathbb{R}^{\mathrm{n_z}} | \\ &\nabla_{\boldsymbol{z}} G_i^\top(\boldsymbol{z},\boldsymbol{p})\; \boldsymbol{d} \leq 0,\; i \in \mathbb{A}(\boldsymbol{z},\boldsymbol{p}),\; \mu_i = 0; \\ &\nabla_{\boldsymbol{z}} G_i^\top(\boldsymbol{z},\boldsymbol{p})\; \boldsymbol{d} = 0,\; i \in \mathbb{A}(\boldsymbol{z},\boldsymbol{p}),\; \mu_i > 0; \\ &\nabla_{\boldsymbol{z}} H_j^\top(\boldsymbol{z},\boldsymbol{p})\; \boldsymbol{d} = 0,\; j = 1,...,\mathrm{n_H}\}. \end{aligned} \tag{3}$$

Together, conditions (c) and (e) of Definition 1 are sufficient for the optimality of solution $\boldsymbol{z}$ for fixed task parameter $\boldsymbol{p}$.

A. The Sensitivity Theorem

Provided that the solution $\hat{\boldsymbol{z}}$ to (1) at nominal task parameter $\hat{\boldsymbol{p}}$ is a strongly regular locally optimal solution, the Sensitivity Theorem allows the sensitivity of $\hat{\boldsymbol{z}}$ with respect to $\hat{\boldsymbol{p}}$ to be easily determined [5] [9] [6]:

Theorem 1: Sensitivity Theorem [5]
Let $J, G_1, ..., G_{\mathrm{n_G}}, H_1, ..., H_{\mathrm{n_H}}$ be twice continuously differentiable and let $\hat{\boldsymbol{p}}$ be a nominal task parameter. Let $\hat{\boldsymbol{z}}$ be a strongly regular local solution of NLP($\hat{\boldsymbol{p}}$) with Lagrange multipliers $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\lambda}}$. Then:
(a) $(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{\mu}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{\lambda}}(\hat{\boldsymbol{p}}))$ is continuously differentiable with respect to $\hat{\boldsymbol{p}}$ with

$$\begin{bmatrix} \frac{d\boldsymbol{z}}{d\boldsymbol{p}}(\hat{\boldsymbol{p}}) \\ \frac{d\boldsymbol{\mu}}{d\boldsymbol{p}}(\hat{\boldsymbol{p}}) \\ \frac{d\boldsymbol{\lambda}}{d\boldsymbol{p}}(\hat{\boldsymbol{p}}) \end{bmatrix} = - \begin{bmatrix} \mathrm{L}''_{\boldsymbol{zz}} & (\mathrm{G}'_{\boldsymbol{z}})^\top & (\mathrm{H}'_{\boldsymbol{z}})^\top \\ \hat{\Lambda}\mathrm{G}'_{\boldsymbol{z}} & \hat{\Gamma} & 0 \\ \mathrm{H}'_{\boldsymbol{z}} & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{L}''_{\boldsymbol{zp}} \\ \hat{\Lambda}\mathrm{G}'_{\boldsymbol{p}} \\ \mathrm{H}'_{\boldsymbol{p}} \end{bmatrix}, \tag{4}$$

where $\hat{\Lambda} := diag(\hat{\mu}_1, ..., \hat{\mu}_{\mathrm{n_G}})$, $\hat{\Gamma} = diag(G_1, ..., G_{\mathrm{n_G}})$, and all functions and their derivative are evaluated at $(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{p}})$, and,
(b) there exist neighborhoods $\mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$ and $\mathbb{B}_\delta(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\lambda}})$, such that the NLP($\hat{\boldsymbol{p}}$) has a unique, strongly regular local minimum

$$(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{\mu}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{\lambda}}(\hat{\boldsymbol{p}})) \in \mathbb{B}_\delta(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\lambda}})$$

for each $\boldsymbol{p} \in \mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$ and $\mathbb{A}(\hat{\boldsymbol{z}}, \hat{\boldsymbol{p}}) = \mathbb{A}(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}})$.

The result of the Sensitivity Theorem is that for a perturbed task parameter $\boldsymbol{p} \neq \hat{\boldsymbol{p}}$, in some neighborhood $\mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$ of the nominal parameter $\hat{\boldsymbol{p}}$, there exists a perturbed solution $\boldsymbol{z}(\boldsymbol{p})$ which is also a strongly regular local solution. The theorem indicates that the neighborhoods $\mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$ and $\mathbb{B}_\delta(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\lambda}})$ exist, that a differentiable vector function $[\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{\mu}(\boldsymbol{p}), \boldsymbol{\lambda}(\boldsymbol{p})]$ which satisfies Def. 1 can be determined in a neighborhood $\mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$ of $\hat{\boldsymbol{p}}$, and that there exist uniquely defined functions

$$(\boldsymbol{z}(\cdot), \boldsymbol{\mu}(\cdot), \boldsymbol{\lambda}(\cdot)) : \mathbb{B}_\epsilon(\hat{\boldsymbol{p}}) \to \mathbb{B}_\delta(\hat{\boldsymbol{z}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\lambda}}).$$

Important facts which come as consequence of this are that these mappings are required to be continuously differentiable (therefore Lipschitz continuous), and that the application of the theorem requires that the index set of the active inequality constraints is fixed within $\mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$.

The above leads to a method which allows the approximation of an optimal solution when the task parameter $\boldsymbol{p}$

is perturbed: Part (a) of Theorem 1 yields the sensitivity $\frac{d\boldsymbol{z}}{d\boldsymbol{p}}$ from the linear equation (4); Part (b) indicates the continuous differentiability of $\boldsymbol{z}(\boldsymbol{p})$ while $\boldsymbol{p}$ remains in some neighborhood of $\hat{\boldsymbol{p}}$, where [5]

$$\bar{\boldsymbol{z}}(\boldsymbol{p}) := \hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}) + \frac{d\boldsymbol{z}}{d\boldsymbol{p}}(\hat{\boldsymbol{p}})(\boldsymbol{p} - \hat{\boldsymbol{p}}) \tag{5}$$

provides an approximation of the optimal solution of the perturbed task parameter.

Equation (5) forms the basis for open-loop real-time optimal control via sensitivity analysis [5] [9], which allows for an online approximation of the NLP($\boldsymbol{p}$) based on a predetermined optimal solution to a nominal NLP($\hat{\boldsymbol{p}}$). The calculation of $\bar{\boldsymbol{z}}$ requires only a matrix-vector product and two vector summations. The time taken to compute (5) is therefore negligible in comparison to that of the computation of NLP($\boldsymbol{p}$), and the update rule is real-time-capable.

B. Uncertain Task Parameter Neighborhood Approximation

The approximation based on (5) suffers from the consequence that it is only valid locally - for task parameters $\boldsymbol{p}$ in some neighborhood $\mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$ of $\hat{\boldsymbol{p}}$. The Sensitivity Theorem assures that this neighborhood exists, but neglects to provide a clear description of its size. A conservative approximation of the size of this neighborhood has been developed [6], which is based on the requirements that the resultant approximated solutions remain Lipschitz continuous and are within a radius of the nominal solution wherein the set of active constraints does not change.

A consequence of Def. 1 and Thm. 1, is that the solution $\boldsymbol{z}(\boldsymbol{p})$ and the mappings $\boldsymbol{p} \mapsto \boldsymbol{z}(\boldsymbol{p})$, $\boldsymbol{z} \mapsto \boldsymbol{G}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p})$, and $\boldsymbol{p} \mapsto \boldsymbol{G}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p})$ are continuously differentiable, and therefore locally Lipschitz continuous, within the neighborhood $\mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$ of $\hat{\boldsymbol{p}}$. A similar statement can be made about $\boldsymbol{z}(\boldsymbol{p}) \mapsto \boldsymbol{H}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p})$. However, equality constraints are always active for solutions belonging to $\mathbb{S}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p})$, while the set of active inequality constraints $\mathbb{A}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p})$ can vary across $\mathbb{S}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p})$. As only the inequality constraints affect a change in the local active constraint set $\mathbb{A}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p})$, the equality constraints need not be considered in the remainder.

Exploiting Lipschitz continuity, and the dependence of $\boldsymbol{G}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p})$ on $\boldsymbol{z}(\boldsymbol{p})$ and its possible dependence on $\boldsymbol{p}$, the relationship of the constraint space to the solution and task parameter spaces can be written as

$$\begin{aligned}\left\|\boldsymbol{G}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p}) - \boldsymbol{G}(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}})\right\|_{\mathrm{q}} \\ \leq L_{\mathrm{G}} \left\|\boldsymbol{z}(\boldsymbol{p}) - \hat{\boldsymbol{z}}(\hat{\boldsymbol{p}})\right\|_{\mathrm{q}} + L_{\mathrm{p}} \left\|\boldsymbol{p} - \hat{\boldsymbol{p}}\right\|_{\mathrm{q}},\end{aligned} \tag{6}$$

where $\|\cdot\|_{\mathrm{q}}$ denotes any q-norm, with $q \in \mathbb{N} \cup \{\infty\}$; L_{G} is the Lipschitz constant (LC) corresponding to the mapping $\boldsymbol{z} \mapsto \boldsymbol{G}(\boldsymbol{z}, \boldsymbol{p})$; and L_{p} corresponds to the mapping $\boldsymbol{p} \mapsto \boldsymbol{G}(\boldsymbol{z}, \boldsymbol{p})$. A general q-norm is indicated as the Lipschitz properties hold for any norm, so long as the norm is used consistently.

The first right-hand term of (6) describes the relationship of the constraint space to the solution space. Through properties of Lipschitz continuity, this relationship can be extended to indicate that of the solution space to task parameter space

$$\begin{aligned}\left\|\boldsymbol{G}(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p}) - \boldsymbol{G}(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}})\right\|_{\mathrm{q}} \\ \leq L_{\mathrm{G}} L_{\mathrm{z}} \left\|\boldsymbol{p} - \hat{\boldsymbol{p}}\right\|_{\mathrm{q}} + L_{\mathrm{p}} \left\|\boldsymbol{p} - \hat{\boldsymbol{p}}\right\|_{\mathrm{q}},\end{aligned} \tag{7}$$

with L_{z} being the LC corresponding to the mapping $\boldsymbol{p} \mapsto \boldsymbol{z}(\boldsymbol{p})$. For the ith constraint, this implies

$$\left|G_i(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p}) - G_i(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}})\right| \leq (L_{\mathrm{G}} L_{\mathrm{z}} + L_{\mathrm{p}}) \left\|\boldsymbol{p} - \hat{\boldsymbol{p}}\right\|_{\mathrm{q}}. \tag{8}$$

Imposing on (8) the invariance requirements of $\mathbb{A}(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}})$ within $\mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$ and requiring that $\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}})$ and $\boldsymbol{z}(\boldsymbol{p})$ be strongly regular in the neighborhood of the $\hat{\boldsymbol{p}}$, and that they are therefore feasible, with $G_i(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p}) \leq 0$ and $G_i(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}}) \leq 0$ for $i \notin \mathbb{A}(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}}))$, it follows

$$\begin{aligned}G_i(\boldsymbol{z}(\boldsymbol{p}), \boldsymbol{p}) &\leq G_i(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}}) + (L_{\mathrm{G}} L_{\mathrm{z}} + L_{\mathrm{p}}) \left\|\boldsymbol{p} - \hat{\boldsymbol{p}}\right\|_{\mathrm{q}} \\ &\overset{!}{<} 0, i \notin \mathbb{A}(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}}).\end{aligned} \tag{9}$$

It is then finally possible to describe a conservative upper bound of $\mathbb{B}_\epsilon(\hat{\boldsymbol{p}})$ by

$$\left\|\boldsymbol{p} - \hat{\boldsymbol{p}}\right\|_{\mathrm{q}} \leq \frac{1}{L_{\mathrm{G}} L_{\mathrm{z}} + L_{\mathrm{p}}} \cdot \min_{i \notin \mathbb{A}(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}})} \{-G_i(\hat{\boldsymbol{z}}(\hat{\boldsymbol{p}}), \hat{\boldsymbol{p}}))\}. \tag{10}$$

Note: in the absence of at least one inactive inequality constraint dependent on $\boldsymbol{p}$, $L_{\mathrm{p}} = 0$.

Typically, these LCs must be approximated - a commonly non-trivial and costly task [20]. Endeavors have been made to compute these constants more efficiently [21] [22] - however the time and computational effort increase drastically with problem size and complexity. Therefore, a conservative but effective (see Sec. IV-C) compromise [6] [12] is made to approximate the LCs to the maxima of the discretized functions governing the respective Jacobians $\mathrm{G}'_{\boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{p})$, $\frac{d\boldsymbol{z}}{d\boldsymbol{p}}$, and $\mathrm{G}'_{\boldsymbol{p}}(\boldsymbol{z}, \boldsymbol{p})$ on each discretized active constraint set $\mathbb{A}(\boldsymbol{z}, \boldsymbol{p})$.

C. Analysis of the Workspace of a Robotic Manipulator in a given Mission

In the early stages of mission planning, the particular properties of the workspace of the robot with respect to a given optimal control problem - e.g. where and how constraint boundaries will be approached or a best approach corridor - may not be clear. Understanding these properties allows the uncertainty to be handled appropriately, and can also be an invaluable tool in the planning stages of a given mission. In such an analysis, only the task parameters $\boldsymbol{p}$ which admit a solution to $\mathbb{S}(\boldsymbol{z}, \boldsymbol{p})$ need be considered. This is termed the *task workspace* and is of dimension $\mathrm{n_p}$. By expanding a grid on the task workspace, such that the neighborhoods of the neighboring grid points overlap, the guarantee of existence of a strongly regular solution is provided throughout it.

Algorithm 1, in similar fashion to [6], considers an equidistant grid of points which can be used as an initial coarse parameterization of the task parameter space. The grid points at which the resolution of the NLP results in a strongly regular solution yields a discretized form of the task workspace. At each grid point of the task workspace, the sensitivity of the solution to $\boldsymbol{p}$ is determined, and the neighborhood of validity at that grid point is estimated. The result is referred to here as an initial *neighborhood map*, which gives a first indication of the robustness distribution on the task workspace.

If, in a given active constraint set, each grid point is not contained within the neighborhoods of its neighboring grid points, the grid on this set can be refined to include new grid points between the already-evaluated grid points to produce a refined neighborhood map. In contrast to the equidistant grid method adopted in [6], the method outlined in Alg. 2

Algorithm 1 Task space grid analysis

Require: Discretized grid of task parameters $\hat{\boldsymbol{p}}_i, i = 1$: total number of grid points
  for all i
    Resolve NLP($\hat{\boldsymbol{p}}_i$)
    if NLP($\hat{\boldsymbol{p}}_i$) yields strongly regular solution, **then**
      Conduct Sensitivity Analysis for solution $\hat{\boldsymbol{z}}_i$
  Sort set of strongly regular solutions $\hat{\boldsymbol{z}}$ into sets $\mathbb{A}_j(\hat{\boldsymbol{z}}, \hat{\boldsymbol{p}})$
  for all j
    Approximate L_{G}, $L_{\mathrm{z}}[$, $L_{\mathrm{p}}]$
  for all i
    Estimate neighborhood size of $\hat{\boldsymbol{p}}_i$ using (10)
Output: Neighborhood map

Algorithm 2 Grid refinement within a given $\mathbb{A}(\boldsymbol{z}, \boldsymbol{p})$

Require: $\hat{\boldsymbol{p}}_i, i = 1$: total number of grid points; initial neighborhood map
  for all i
    for all k grid points adjacent to $\hat{\boldsymbol{p}}_i$
      if $\hat{\boldsymbol{p}}_k$~$\hat{\boldsymbol{p}}_i \notin \mathbb{B}_\epsilon(\hat{\boldsymbol{p}}_i)$ **then**
        Select a new grid point $\hat{\boldsymbol{p}}_j \in \mathbb{B}_\epsilon(\hat{\boldsymbol{p}}_i)$
        Resolve NLP($\hat{\boldsymbol{p}}_j$)
        if NLP($\hat{\boldsymbol{p}}_j$) yields strongly regular solution **then**
          Conduct Sensitivity Analysis for solution $\hat{\boldsymbol{z}}_j$
          Estimate neighborhood size of $\hat{\boldsymbol{p}}_j$ using (10)
Output: Refined neighborhood map

leverages the varying neighborhood size across the active constraint set, which results from the proximity of a grid point to its nearest constraint boundary (see (10)). Only if $\hat{\boldsymbol{p}}_k$ adjacent to a given $\hat{\boldsymbol{p}}_i$ does not lie in the neighborhood of $\hat{\boldsymbol{p}}_i$, a new grid point $\hat{\boldsymbol{p}}_j$ located between the grid points $\hat{\boldsymbol{p}}_i$ and $\hat{\boldsymbol{p}}_k$ is selected from within the neighborhood of $\hat{\boldsymbol{p}}_i$. This method ensures that the neighborhood of $\hat{\boldsymbol{p}}_j$ overlaps with that of the existing $\hat{\boldsymbol{p}}_i$ by definition, while extending the neighborhood map coverage in the direction of $\hat{\boldsymbol{p}}_k$. The refinement can occur iteratively - adding new points until the desired neighborhood overlap is achieved.

III. Mission Description

In this section, the 7 DOF free-floating robotic task, motion planning NLP, and SA are formulated.

A. The 7 Degree of Freedom Free-floating Robot Task

The robotic task to be considered in this paper is taken from the DEOS [23] On-Orbit Servicing scenario, see Fig. 1. Two satellites are involved - a free-floating chaser, consisting of a 7 DOF robot arm mounted to an unactuated base body, and a non-cooperative tumbling target positioned to be within reach of the robot gripper.

Specifically, the approach of the robotic arm from an *initial approach joint configuration* (IC) to a *grasping joint configuration* (GC) is considered, similar to the scenario in [2]. In this GC, the end-effector (EE) of the robot arm will be in a relative pose which is suitable for grasping the target satellite at a pre-defined point, termed the *grasping point* (GP), which is located at a position ${}^{\mathrm{T}}\boldsymbol{r}_{\mathrm{GP}}$ that is fixed in the body-frame of the target satellite.

The free-floating robot begins at rest. In Sections III-A-C, the target satellite has a general angular velocity $\boldsymbol{\omega}_{\mathrm{T}}$. The ultimate position of the GP at the end of the maneuver, and subsequently the necessary final pose of the EE, is determined based on the estimated rotational motion of the target. The energy-optimal motion of the robot required to move from its IC to the GC in a fixed time is determined through the solution of a constrained NLP in joint space [2].

It is possible for the goal GP to be positioned at any point on the sphere with radius $\left|{}^{\mathrm{T}}\boldsymbol{r}_{\mathrm{GP}}\right|$ and centered on the center of mass of the target object. A given estimated initial target satellite pose and angular velocity are inputs to the optimization problem. The position of the GP in the inertial frame evolves from these initial parameters following the rotational dynamics of the target and assuming that there is no translational relative motion. The trajectory resulting from the resolution of the NLP with this input will be nominally optimal to this GP. However, if there is an uncertainty in the estimated pose and angular velocity of the target, and therefore in the position of the GP, the resolved trajectory will need to be departed from [4]. The new trajectory will be obtained using the methods of Sec. II, for the NLP and $\boldsymbol{p}$ described in the following.

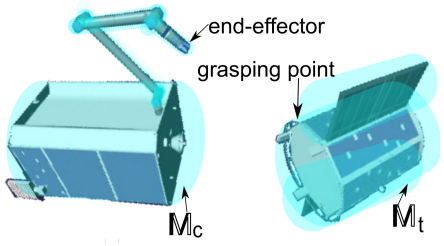

Fig. 1. The target satellite and a chaser spacecraft with attached robot manipulator [23]. The translucent capsules indicate the collision geometry of the chaser $\mathbb{M}_{\mathrm{c}}$ and target $\mathbb{M}_{\mathrm{t}}$.

B. Equations of Motion

The configuration of the space robot can be described by joint positions $\boldsymbol{q}$ with base pose $\boldsymbol{x}_0 = [\boldsymbol{r}_0^\top \ \boldsymbol{\phi}_0^\top]^\top$. The pose of the EE is defined by $\boldsymbol{x}_{\mathrm{ee}} = [\boldsymbol{r}_{\mathrm{ee}}^\top \ \boldsymbol{\phi}_{\mathrm{ee}}^\top]^\top$. Their time derivatives are $\dot{\boldsymbol{q}}$, $\dot{\boldsymbol{x}}_0 = [\boldsymbol{v}_0^\top \ \boldsymbol{\omega}_0^\top]^\top$, and $\dot{\boldsymbol{x}}_{\mathrm{ee}} = [\boldsymbol{v}_{\mathrm{ee}}^\top \ \boldsymbol{\omega}_{\mathrm{ee}}^\top]^\top$. Note that $[\boldsymbol{r}_{\mathrm{ee}}^\top, \ \boldsymbol{\phi}_{\mathrm{ee}}^\top]^\top = [{}^{\mathrm{IT}}\mathrm{A}(\phi_{\mathrm{T}})^{\mathrm{T}}\boldsymbol{r}_{\mathrm{GP}}, \ \phi_{321}(\mathrm{A}(180_{\mathrm{y}}) \ {}^{\mathrm{IT}}\mathrm{A}(\phi_{\mathrm{T}}))]^\top$, where $\mathrm{A}(180_{\mathrm{y}})$ ensures that the EE frame z-axis points toward the target (see Fig. 2).

As the space robot is initially set at rest and free-floating dynamics are assumed, the CoM of the system does not move and the total momentum is constantly zero. The conservation of linear $\boldsymbol{P}$ and angular $\boldsymbol{L}$ momentum relates the velocity of the chaser base body to the joint velocities through

$$[\boldsymbol{P}^\top \ \ \boldsymbol{L}^\top]^\top = \mathrm{H}_0\dot{\boldsymbol{x}}_0 + \mathrm{H}_{0\mathrm{m}} \ \dot{\boldsymbol{q}} \tag{11}$$

where H_0 is the locked inertia of the multi-body system and $\mathrm{H}_{0\mathrm{m}}$ is the dynamic-coupling inertia matrix.

In the case of the robot and scenario described above, the system is kinematically redundant and non-holonomic. The time derivatives of the pose of the EE and the configuration of the robot are related through the generalized Jacobian J^*, seen in the inertial frame as

$$\dot{\boldsymbol{x}}_{\mathrm{ee}} = \mathrm{J}^* \ \dot{\boldsymbol{q}}. \tag{12}$$

The free-floating system's generalized Jacobian J^* is therefore a $[6 \times 7]$ matrix, and depends not only on the kinematics

of the robot, but also on the configuration-dependent inertia of the robot.

The resulting generalized equations of motion for the free-floating robot are given by

$$\mathrm{H}^*(\boldsymbol{q})\ \ddot{\boldsymbol{q}} + \mathrm{C}^*(\boldsymbol{q}, \dot{\boldsymbol{q}})\ \dot{\boldsymbol{q}} = \boldsymbol{\tau}, \tag{13}$$

where $\mathrm{H}^*(\boldsymbol{q}) \in \mathbb{R}^{7\times7}$ is the generalized inertia matrix, $\mathrm{C}^*(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{7\times7}$ contains the nonlinear and Coriolis terms, and $\boldsymbol{\tau} \in \mathbb{R}^{7\times1}$ includes the manipulator joint torques [24].

C. The Parametric Optimization Problem

The path to be followed in the joint space of the robot is parameterized by a set of order-4 uniform b-splines - one for each joint - with fixed boundary conditions, except for the final position, and with one free b-spline vertex to determine the intermediate curve. The pair of these free parameters for all of the robot joints comprises the set of optimization solution parameters $\boldsymbol{z} \in \mathbb{R}^{14}$.

Let $q_i(t) \in \mathbb{R}, i = 1:7$, denote the position of one of the robot joints on a given discretized time interval, as in (1). The pose $\boldsymbol{x}_{\mathrm{ee}}$ and torques $\boldsymbol{\tau}$ are obtained at each via point from the joint positions and derivatives sampled from the b-splines using (11), (12), and (13).

The NLP is then formulated as

$$\begin{aligned}
&\min_{\boldsymbol{z}} \quad J = \Sigma(\boldsymbol{\tau}\cdot\dot{\boldsymbol{q}})^2 \qquad (14)\\
&s.t.\ \ c_{\mathrm{position}}(\boldsymbol{q}(\boldsymbol{z}), t) \le 0,\\
&\quad\ c_{\mathrm{velocity}}(\dot{\boldsymbol{q}}(\boldsymbol{z}), t) \le 0,\\
&\quad\ c_{\mathrm{torque}}(\boldsymbol{q}(\boldsymbol{z}), \dot{\boldsymbol{q}}(\boldsymbol{z}), \ddot{\boldsymbol{q}}(\boldsymbol{z}), t) \le 0,\\
&\quad\ c_{\mathrm{collision}}(\boldsymbol{q}(\boldsymbol{z}), \boldsymbol{x}_0(\boldsymbol{q}(\boldsymbol{z})), \mathbb{M}_{\mathrm{t}}, \mathbb{M}_{\mathrm{c}}, t) \le 0,\\
&\quad\ c_{\mathrm{finalpose}}(\boldsymbol{q}(\boldsymbol{z}), \boldsymbol{x}_0(\boldsymbol{q}(\boldsymbol{z})), \boldsymbol{r}_{\mathrm{T}}, \boldsymbol{\phi}_{\mathrm{T}}(\mathrm{t_f})) = \boldsymbol{x}_{\mathrm{ee}}(\mathrm{t_f}),\\
&for\ \ t = 0, ..., \mathrm{n}.
\end{aligned}$$

The constraints c_{position}, c_{velocity}, and c_{torque} are box constraints which limit the motion of the robot joints. Collision of the robot with itself and collision of the robot with the target are mitigated through $c_{\mathrm{collision}}$. The presence of collisions is evaluated using ODE [25], following which $c_{\mathrm{collision}}$ is computed as a penetration depth of the geometries $\mathbb{M}_{\mathrm{t}}$ and $\mathbb{M}_{\mathrm{c}}$ [3]. The geometry $\mathbb{M}_{\mathrm{c}}$ consists of capsules fit over the chaser base and each link of the robot, while the target geometry $\mathbb{M}_{\mathrm{t}}$ is comprised of capsules which encapsulate the central cylinder and each of the two solar panels which protrude perpendicularly from the central body (see Fig. 1). The geometry is chosen to avoid the sharp corners of more tightly fitting rectangular polytopes. This does result in slightly oversized geometry. To prevent spurious collision detections, the collision constraint is relaxed to ignore the protruding endcaps of the capsule polytopes. The equality constraint $c_{\mathrm{finalpose}}$ indicates the final pose of the robot EE $\boldsymbol{x}_{\mathrm{ee}}(t_{\mathrm{f}})$. The required pose $\boldsymbol{x}_{\mathrm{ee}}(t_{\mathrm{f}})$ is defined by the 3 angles describing the orientation of the target $\boldsymbol{\phi}_{\mathrm{T}}$, the fixed ${}^{\mathrm{T}}\boldsymbol{r}_{\mathrm{GP}}$, and the position of the CoM of the target $\boldsymbol{r}_{\mathrm{T}}$. The NLP has been solved using the *slsqp* algorithm provided by NLopt [26], utilizing SpaceDyn [27] for dynamics and kinematics computations.

D. Sensitivity of Solutions to End-Effector Pose

Assuming a fixed relative distance between the CoM of the spacecraft (see above), the minimal set of parameters defining the task in question is $\boldsymbol{p}_{\boldsymbol{\phi}_{\mathrm{T}}} = \boldsymbol{\phi}_{\mathrm{T}}(t_{\mathrm{f}}) \in \mathbb{R}^3$. For the purposes of analysis, the remainder of this section addresses the parametric sensitivity for the condition $\boldsymbol{\omega}_{\mathrm{T}} = \mathbf{0}$ [rad/s].

To determine the sensitivities $\frac{d\boldsymbol{z}}{d\boldsymbol{p}}$ using (4), it is necessary to evaluate the RHS of (4). The derivatives with respect to $\boldsymbol{z}$, i.e. $\mathrm{L}''_{\boldsymbol{zz}}$, $\mathrm{G}'_{\boldsymbol{z}}$, and $\mathrm{H}'_{\boldsymbol{z}}$, are provided by the result of the optimization. The derivative $\mathrm{L}''_{\boldsymbol{zp}}$ requires $\mathrm{J}'_{\boldsymbol{p}}$, $\mathrm{G}'_{\boldsymbol{p}}$, and $\mathrm{H}'_{\boldsymbol{p}}$ - which also provide the remaining RHS terms:

First, given the cost function $\boldsymbol{J}$ of the NLP defined in (14),

$$\boldsymbol{J}'_{\boldsymbol{p}_{\boldsymbol{\phi}_{\mathrm{T}}}} = \frac{\partial J}{\partial \boldsymbol{\phi}_{\mathrm{T}}} = \frac{\partial J}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{\mathrm{ee}}}\frac{\partial \boldsymbol{x}_{\mathrm{ee}}}{\partial \boldsymbol{\phi}_{\mathrm{T}}} \tag{15}$$

$$= \sum_{k=0}^{N}\left[\sum_{i=1}^{7}\frac{\partial}{\partial \boldsymbol{q}}[\tau_i^2\dot{q}_i^2] + 2\sum_{i=1}^{6}\sum_{j=i+1}^{7}\frac{\partial}{\partial \boldsymbol{q}}[\tau_i\tau_j\dot{q}_i\dot{q}_j]\right]_k \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{\mathrm{ee}}}\frac{\partial \boldsymbol{x}_{\mathrm{ee}}}{\partial \boldsymbol{\phi}_{\mathrm{T}}},$$

as $\frac{\partial \dot{\boldsymbol{q}}}{\partial \boldsymbol{q}} = 0$; $\frac{\partial \ddot{\boldsymbol{q}}}{\partial \boldsymbol{q}} = 0$, given (13); and ($i, j$, and k as above)

$$\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{q}} = \frac{\partial}{\partial \boldsymbol{q}}[\mathrm{H}^*(\boldsymbol{q})\ \ddot{\boldsymbol{q}} + \mathrm{C}^*(\boldsymbol{q}, \dot{\boldsymbol{q}})\ \dot{\boldsymbol{q}}]$$

$$\frac{\partial}{\partial \boldsymbol{q}}[\tau_i^2\dot{q}_i^2] = 2\tau_i\frac{\partial \tau_i}{\partial \boldsymbol{q}}\dot{q}_i^2$$

$$\frac{\partial}{\partial \boldsymbol{q}}[\tau_i\tau_j\dot{q}_i\dot{q}_j] = \frac{\partial \tau_i}{\partial \boldsymbol{q}}\tau_j\dot{q}_i\dot{q}_j + \tau_i\frac{\partial \tau_j}{\partial \boldsymbol{q}}\dot{q}_j\dot{q}_j.$$

For small variances, $\delta\boldsymbol{x}_{\mathrm{ee}} = \mathrm{J}^*\delta\boldsymbol{q}$. To determine $\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{\mathrm{ee}}}$, the inverse of J^* is needed. As the robot is redundant and J^* is not square, the pseudoinverse $\mathrm{J}^{\#}$ is employed to derive $\dot{\boldsymbol{q}} = \mathrm{J}^{\#}\dot{\boldsymbol{x}}_{\mathrm{ee}} + (\mathrm{I} - \mathrm{J}^{\#}\mathrm{J})\dot{\boldsymbol{q}}_0$, where $\dot{\boldsymbol{q}}_0$ is a $[7\times1]$ vector. In this work, $\|\dot{\boldsymbol{q}}\|$ should be minimized, so that the objective is also minimized. Solutions are therefore chosen such that the second term tends to zero, and $\partial\boldsymbol{q}/\partial\boldsymbol{x}_{\mathrm{ee}} \approx \mathrm{J}^{\#}$.

The sensitivity of the solution with respect to $\boldsymbol{p}_{\boldsymbol{\phi}_{\mathrm{T}}}$ depends on the associated sensitivity with respect to the final pose of the EE, as shown in (15). Consideration of the sensitivity of the cost J, and by extension the sensitivity of (14), to changes in $\boldsymbol{x}_{\mathrm{ee}}(t_{\mathrm{f}})$ requires less computational effort and provides a more intuitive insight to a prospective operator in the selection of a trustworthy motion plan (see Sec. IV-B). In the remainder of this work, the uncertainty is therefore parameterized in terms of $\boldsymbol{p}_{\mathrm{pose}} = \boldsymbol{x}_{\mathrm{ee}}(t_{\mathrm{f}}) \in \mathbb{R}^6$, and the template of (15) is suitably adjusted.

The above partial derivatives are also used in determining $\mathrm{G}'_{\boldsymbol{p}_{\boldsymbol{\phi}_{\mathrm{T}}}} = \frac{\partial \mathrm{G}}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{\mathrm{ee}}}$ and $\mathrm{H}'_{\boldsymbol{p}_{\boldsymbol{\phi}_{\mathrm{T}}}} = \frac{\partial \mathrm{H}}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{\mathrm{ee}}}$, which are comparatively trivial to derive and further discussion is neglected here. *Note:* the derivatives of $c_{\mathrm{collision}}$ must be determined numerically, but it is noted that $\frac{\partial c_{\mathrm{collision}}}{\partial \boldsymbol{x}_{\mathrm{ee}}} = \frac{\partial c_{\mathrm{collision}}}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}_{\mathrm{ee}}}$.

IV. Scenario Simulation and Analysis

In this section, the scenario described in Sec. III will be analyzed using the methods described in Sec. II. Additionally, the efficacy of the sensitivity-based update as a foundation for online re-planning is explored.

A. Simulation Framework

The solution to the NLP in (14) describes the motion of the space robot. The base is initially located at [0 0 0] m, with the x-axis oriented towards the CoM of the target, see Fig. 1. The target CoM is located at [0.928 0.1 0.057] m. In the IC, the joints are set to $[0, -0.1745, 0, 1.0472, 0, -1.2217, 0]$ rad, and the EE of the robot is located at [0.74 0.1 1.04] m, with orientation $[0, \pi/2, 0]$ rad.

The set of all possible GP positions is described as a function of the target orientation by a sphere centered at the CoM of the target with a radius of 0.52 m. For practical reasons, the investigated task parameter sub-space is limited to those corresponding to orientations of the target where the GP is located on the hemisphere closest to the initial position of the chaser base. For purpose of methodological discussion, $\boldsymbol{\omega}_{\mathrm{T}} = \mathbf{0}$, without loss of generality to the demonstration. The set of task parameters $\mathbb{P}$ for the subsequent analysis is then defined by discretizing the subset of possible grasping positions and determining the corresponding possible EE poses at each of these discretized points, where the orientation of the EE is required to be perpendicular to the sphere at the GP (see Fig. 2), fixing $\phi_{\mathrm{ee},1}$ and $\phi_{\mathrm{ee},2}$. Multiple target orientations produce GP at the same location, for different values of $\phi_{\mathrm{ee},3}$, but the geometry of the target will alter the motion requirements to satisfy motion constraints, see Fig. 3.

For this analysis, 25266 target satellite orientations were selected at regular intervals on the range of orientations for which corresponding $\boldsymbol{p}_{\mathrm{pose}} = \boldsymbol{x}_{\mathrm{ee}}$ was determined to lie in the region of interest. Then, for each given $\boldsymbol{p}_{\mathrm{pose}}$ on this task parameter space, a Monte Carlo search was conducted to determine the near-globally optimal solution to the NLP, which was then selected as the nominal solution for this $\boldsymbol{p}_{\mathrm{pose}}$. Finally, the sensitivity of the solutions with respect to $\boldsymbol{p}_{\mathrm{pose}}$ was obtained using (5) and Sec. III-D. The neighborhood of each nominal pose was estimated as described in Sec. II.

B. Task-Workspace Analysis

This section analyzes the resulting task workspace with the goal to suggest which of its regions may provide suitable options for the GP in a mission context.

First, the activity of the constraints of the problem is analyzed. The set of active inequality constraints $\mathbb{A}(\boldsymbol{z}, \boldsymbol{p}_{\mathrm{pose}})$ was determined for the solution at each $\boldsymbol{p}_{\mathrm{pose}}$. The grid points were then sorted into active constraint sets, where the solutions of each neighboring grid point have the same set of active constraints. Of the 2062 tallied sets $\mathbb{A}(\boldsymbol{z}, \boldsymbol{p}_{\mathrm{pose}})$ on the task workspace, approximately 10% of the sets each contain at least 200 of the sample $\boldsymbol{p}_{\mathrm{pose}}$ on the regular grid.

The neighborhoods $\mathbb{B}_{\epsilon}(\hat{\boldsymbol{p}}_{\mathrm{pose}})$ across the task workspace were then estimated. Using the constraint values and gradients obtained from the resolution of the NLP($\hat{\boldsymbol{p}}_{\mathrm{pose}}$) at each grid point and the associated sensitivities of the solutions, the LCs L_{G} and L_{z} were approximated by the largest respective gradient present in each active constraint set. Note: in the presented scenario, $L_{\mathrm{p}} = 0$. For the estimate (10), the 2-norm was selected [28].

A neighborhood map for two of the largest active constraint sets on the task workspace is developed here. No constraints are active in set $\mathbb{A}_1$. A sample trajectory is provided in Fig. 2(a). In set $\mathbb{A}_2$, the collision constraint governing the capsules for robot link 7 and the target body are active over the last four via points. This does not mean that the bodies are in collision, but rather that the optimizer is actively steering the solution to avoid the condition. The motion resulting in this active constraint is depicted in Fig. 3. The neighborhood map, which provides a first indication of the robustness distribution on the task workspace, is presented in Fig. 4.

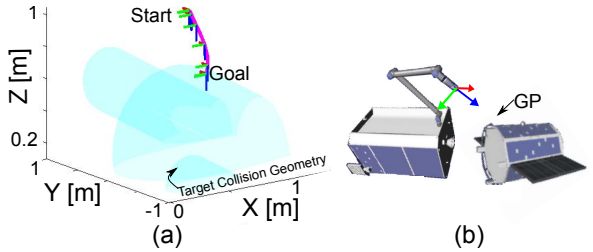

Fig. 2. The approach trajectory (a) in pink of the robot end-effector (EE) to enable grasp at GP (b) for a sample grid point in active constraint set $\mathbb{A}_1$. The orientation of the EE is indicated at via points by the corresponding EE frame depicted in (b).

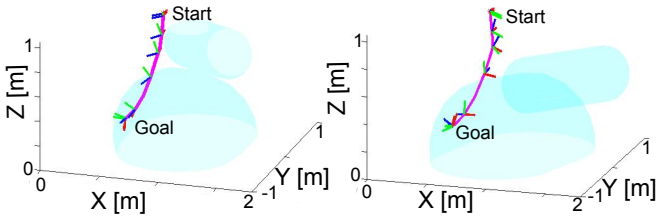

Fig. 3. The approach trajectories for two sample grid points from active constraint set $\mathbb{A}_2$ with the same end-effector final position, but differing target satellite orientations. The blue capsules correspond to the geometry depicted in Fig.1, oriented to the target orientation query.

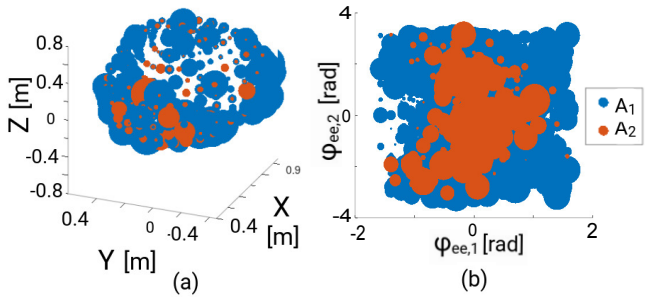

Fig. 4. Task workspace and neighborhood map: Estimated neighborhoods for grid points on the discretized task workspace of the two largest active constraint sets in (a) position and (b) orientation. These maps provide an indication of the robustness distribution on the task workspace.

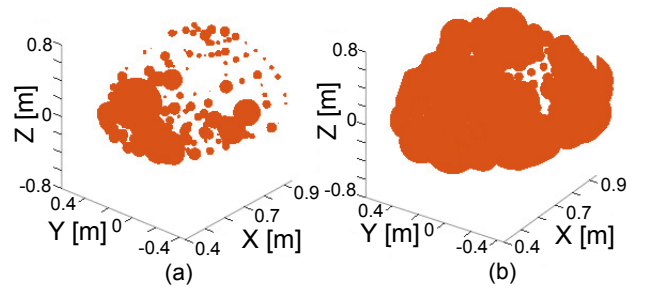

Fig. 5. (a) Initial neighborhood map: Estimated neighborhood sizes for set $\mathbb{A}_2$ on the initial grid using Alg. 1. (b) Refined neighborhood map: Estimated neighborhood sizes for set $\mathbb{A}_2$ on the grid discretizing the task workspace, which has been refined using Alg. 2. This refinement extends the guarantees of robustness on the task workspace, as discussed in Sec.II-C.

The task parameter grid is now refined using Alg. 2, with the goal of maximizing the task workspace. Fig. 5 indicates, as an example, the achieved refined neighborhood map for set $\mathbb{A}_2$. For ease of reading, only the projection on the translational dimensions is depicted. This grid refinement, which introduced sample points between the existing grid points as described in Alg. 2, required approximately factor 5 fewer new points than would be required under a uniform grid

refinement to achieve the same neighborhood coverage. Note that it is expected for all of the new grid points to belong to $\mathbb{A}_2$, as they are selected from within the neighborhoods of the existing grid points.

The variance in neighborhood size across the task workspace reflects the proximity of the grid point to a constraint boundary. The refinement of Alg. 2 also highlights the boundaries of the connected sub-sets within the same set of active constraints, see Fig. 5(b).

The subsets of the task workspace which could be targeted in a mission for the EE pose are $\mathbb{P}_1 = \{\boldsymbol{p} \in \mathbb{P} | 0.4 < r_{\text{ee,x}} < 0.5, -0.1 < r_{\text{ee,y}} < 0.4, -0.2 < r_{\text{ee,z}} < 0.4, |\phi_{\text{ee},1}| < 0.4\pi, |\phi_{\text{ee},2}| < \pi\}$ and $\mathbb{P}_2 = \{\boldsymbol{p} \in \mathbb{P} | r_{\text{ee,z}} > 0.2\}$, where there are a large concentration of no-active-constraint solutions and the neighborhoods are largest, see Fig. 6.

C. Statistical Analysis of Re-planning Methods

While planning the motion for the approach task under nominal conditions is a good starting point, re-planning as efficiently as possible when the nominal conditions are departed from is necessary. A statistical comparison of the presented method to cold and warm-started local searches is conducted in this section.

A statistical analysis of 1000 random query points selected from within the estimated neighborhoods of grid points belonging to $\mathbb{A}_2$ for optimization time and cost is presented in Table I. The results in the column 'Cold Start' refer to the case where a locally optimal solution was obtained for each random online query from a global search; 'Warm Start' uses the nominal solution of the grid point of the neighborhood within which the random online query lies as the initial guess for obtaining a locally optimal solution; and 'Sens.-based Update' uses the sensitivity-based approximation outlined in Sec.II-A.

Note that, while the trajectories obtained using the approximation (5) are sub-optimal compared to the other two methods, the solutions for all of the online queries selected within the neighborhood of the nominal grid point were feasible, in accordance with Thm. 1. These trajectories were also obtained in a fraction of the time required by the optimization-based methods. For comparison, the cold start had a 39.7% convergence rate.

A similar analysis was conducted at larger distances from the nominal grid point. For each of 10 selected nominal grid points, 100 random sample queries were selected from spherical shells of 0.2 neighborhood radius thickness and an inner radius ranging from one neighborhood radius to 4 radii. The sensitivities of the grid point were used to approximate solutions at the sample query using (5). The feasibility of the approximated solution was evaluated, see Table II. Thm. 1 provides guarantees of feasibility only within the neighborhood of the task parameter, but the neighborhood estimation (10) is conservative. It is therefore possible for feasible solutions to lie near, but outside of, the range of one neighborhood radius from the grid point. This is reflected in the overall 11.4% query feasibility up to the boundary of the shell with inner radius of 1.6 neighborhood radii (Table II). When combined with the 100% feasibility of the approximations analyzed from within the neighborhood, this would suggest that the neighborhoods have been conservatively, but well estimated.

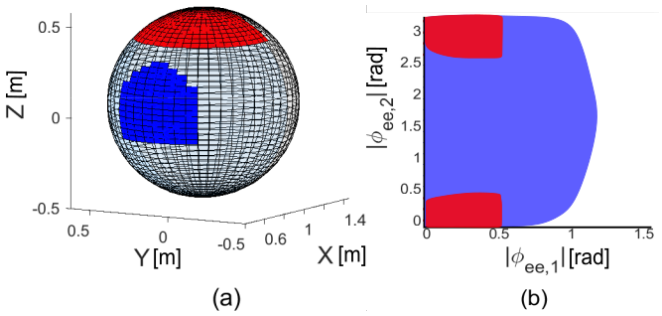

Fig. 6. For the purpose of mission analysis, two potential regions $\mathbb{P}_1$ (blue) and $\mathbb{P}_2$ (red) have been derived from the task workspace (see Figs. 4 and 5), from which the final end-effector (a) position and (b) orientation could be selected.

TABLE I
STATISTICAL ANALYSIS OF RE-PLANNING METHODS

		Cold Start	Warm Start	Sens.-based Update
Computation Time [s]	mean	14.053	0.6583	$1.42e-6$
	std	2.0765	0.4892	$2.36e-6$
Cost	mean	0.0008508	0.0024451	0.0085806
	std	0.001472	0.004359	0.002572

TABLE II
STATISTICAL ANALYSIS OF TASK PARAMETER PERTURBATIONS BEYOND NEIGHBORHOOD SIZE

Radial distance [radii]		$1-1.2$	$1.2-1.4$	$1.4-1.6$	>1.6
Warm Start	Cost	0.0011	0.0023	0.0057	0.0075
	Comp. Time [s]	1.41	1.62	1.82	2.63
Sens.-based Update	Feasibility [%]	25.8	8.2	0.32	0
	Cost deviation	0.0036	0.0059	0.077	-
	Comp. Time [s]	$1.4e-6$	$1.4e-6$	$1.4e-6$	-

For comparison, the updated solution was used to provide an initial guess to warm start re-plan the maneuver. The cost and time statistics of the warm start derived motion plan and the RMS deviation of the cost of the approximated solution from the warm start solution are included in Table II, which suggests that, even if the approximated solution is not itself feasible, it does provide a good initial guess for determining feasible solutions.

D. Discussion

Open questions from the work presented in [6] are those of the scalability of the SA task parameter space and the scalability of the robotic system to which the presented methodology is applied. SA has been conducted above for the 6 dimensional robot end-effector pose - a sensitivity parameter space which is 3 times larger than the 2 dimensional joint angle task parameter space explored in [6]. Additionally, the robot for which the motion is planned in this paper has 5 more joints and a free-floating base compared to the 2 link fixed-base planar arm discussed in that paper. The neighborhood estimation methodology has been shown to be scalable in both task parameter space and robot complexity.

When conducting the sensitivity-based updated at $\boldsymbol{p}$, it is necessary to select the nominal $\hat{\boldsymbol{p}}$ upon which to base the update such that $\boldsymbol{p}$ resides within the estimated neighborhood of $\hat{\boldsymbol{p}}$, as the resultant trajectory is only guaranteed to be feasible within this neighborhood. The outcome of the failure to do so was shown in Table II. It should be noted that, provided this requirement is upheld, under no

circumstances does the neighborhood approximation for $\hat{\boldsymbol{p}}$ cross a constraint boundary. If it is determined that $\boldsymbol{p}$ belongs to a different active constraint set, and therefore lies outside of the neighborhood of $\hat{\boldsymbol{p}}$, a more appropriate $\hat{\boldsymbol{p}}$ should be selected.

The interval on which the functions and mappings discussed in Sec. II are continuously differentiable is, necessarily, bounded. Conversely, boundaries are imposed on the task workspace when any requirement of Thm. 1 breaks down. For example, the assumption of Lipschitz continuity for robotic arms is strong, as it does not hold in the presence of Jacobian singularities. As such, it has been assumed in the preceding that Jacobian singularities can be avoided. A method for computing a Jacobian singularity map for a 6 DOF free-floating robot is presented in [29]. The singularity map is expressed in the robot joint space, with guarantees of completeness, and can be used to treat the singularity avoidance problem as a trivial collision avoidance problem. The involvement of such a singularity map in the analysis of a task parameter space therefore becomes trivial and the overall assumption of Lipschitz continuity remains valid.

V. Conclusions

This paper introduced the concept of a task workspace, which indicates where in the task parameter space a feasible solution can be found efficiently and provably robustly. A method to estimate the neighborhood of validity of the Sensitivity Theorem was presented and used to form neighborhood maps on a discrete grid of the task parameter space. In light of conservative Lipschitz constant approximations, these neighborhood maps indicate where in task parameter space soft guarantees of the existence of an optimal solution to the task can be provided - forming the task workspace. A dynamic exploration of the task workspace was presented which substantially reduced the requisite number of grid points compared to [6]. A methodological comparison of the sensitivity-based update to state-of-the-art optimization-based re-planning was presented.

The development of a task workspace and its analysis are important tools for real-world mission planning. The usefulness of these tools in analyzing a robotic task, understanding constraint boundaries of the robot, and to suggest reasonable relative approach directions and grasping points was demonstrated through the approach maneuver of a 7 DOF free-floating robotic arm to a target satellite. A 3 dimensional task parameter space was considered, but was expressed in 6 dimensions for a more intuitive mission analysis.

Future work will apply the presented method to the capture of a tumbling target and explore the use of the neighborhood estimation as a bounded uncertainty definition for use in robust control.

References

[1] J. Virgili-Llop, C. Zagaris, I. Richard Zappulla, A. Bradstreet, and M. Romano, "A convex-programming-based guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator," *Int. J. of Robot. Research*, vol. 38, no. 1, 2019.

[2] R. Lampariello and G. Hirzinger, "Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time," in *IROS 13*, Tokyo, Japan, November 2013.

[3] C. Specht, A. Bishnoi, and R. Lampariello, "Autonomous spacecraft rendezvous using tube-based model predictive control: Design and application," *J. of Guid., Control, and Dyn.*, vol. 46, no. 7, 2023.

[4] R. Lampariello, H. Mishra, N. Oumer, P. Schmidt, M. De Stefano, and A. Albu-Schäffer, "Tracking control for the grasping of a tumbling satellite with a free-floating robot," *IEEE Robot. and Automat. Lett.*, vol. 3, no. 4, pp. 3638–3645, 2018.

[5] M. Gerdts, *Optimal Control of ODEs and DAEs*. de Gruyter, 2023.

[6] C. Specht, M. Gerdts, and R. Lampariello, "Neighborhood estimation in sensitivity-based update rules for real-time optimal control," in *2020 ECC*, 2020, pp. 1999–2006.

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

[8] T. S. Lembono, A. Paolillo, E. Pignat, and S. Calinon, "Memory of motion for warm-starting trajectory optimization," *IEEE RA-L*, vol. 5, no. 2, pp. 2594–2601, 2020.

[9] C. Büskens, "Optimierungmethoden und Sensitivitätsanalyse für optimale Steuerprozesse mit Steuer- und Zustands-Beschränkungen," Ph.D. dissertation, Westfällischen Wilhelms-Universität Münster, 1998.

[10] C. Büskens and H. Maurer, "Sensitivity analysis and real-time optimization of parametric nonlinear programming problems," *Online Optimization of Large Scale Systems*, 2001.

[11] ——, "SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control," *J. of Comput. and Appl. Math.*, vol. 120, pp. 85–108, 2000.

[12] Lemos-Paião, Ana P., Lopes, Sofia O., and de Pinho, M. D. R., "A parametric optimal control problem applied to daily irrigation," *Math. Model. Nat. Phenom.*, vol. 20, p. 2, 2025.

[13] A. Reiter, H. Gattringer, and A. Müller, *Real-Time Computation of Inexact Minimum-Energy Trajectories Using Parametric Sensitivities*, ser. Mechanisms and Machine Science, C. Farraresi and G. Quaglia, Eds. Springer, June 2018, vol. 49.

[14] A. Reiter, A. Müller, and H. Gattringer, *Rapid Nearly-Optimal Rendezvous Trajectory Planning Using Parameter Sensitivities*, ser. CISM Int. Centre for Mech. Sci. (Courses and Lectures). Springer, Cham, 2019, vol. 584.

[15] C. Büskens and M. Gerdts, "Emergency landing of a hypersonic flight system: A corrector iteration method for admissible real-time optimal control approximations," in *Optimalsteuerungsprobleme in der Luft- und Raumfahrt, Workshop in Greifswald des Sonderforschungsbereichs 255: Transatmospärische Flugsysteme*, Munich, Germany, 2003.

[16] De Marchi, A., Dreves, A., Gerdts, M., S. Gottschalk, and S. Rogovs, "A function approximation approach for parametric optimization," *J. of Optim. Theory and Appl.*, vol. 196, 2022.

[17] C. Specht and R. Lampariello, "Verfahren und system zur planung der bewegung eines kinematisch redundanten roboters," Patent pending Aktenzeichen 10 2025 126 770.7, 2025.

[18] ESA, "Invitation to Tender for Verification and Validation of Real-Time Optimised Safety-Critical GNC SW Systems," *ITT AO/1-10895/21/NL/CRS*, 2021.

[19] P. Lourenço, H. Costa, J. Branco, P. Garoche, A. Sadeghzadeh, J. Frey, and et al., "Verification and validation of optimisation-based control systems: methods and outcomes of vv4rtos," in *ESA GNC and ICATT Conf.*, Sopot, Poland, 2023.

[20] M. Darup and M. Mönnigmann, "Fast computation of lipschitz constants on hyperrectangles using sparse codelists," *Comput. and Chem. Eng.*, vol. 116, pp. 135–143, April 2018.

[21] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, "Efficient and accurate estimation of lipschitz constants for deep neural networks," in *NeurIPS 2019*, Vancouver, Canada, Dec 2019.

[22] D. Teichrib and M. Darup, "Efficient computation of lipschitz constants for mpc with symmetries," in *CDC 2023*, Singapore, Dec 2023.

[23] D. Reintsema, J. Thaeter, A. Rathke, W. Naumann, P. Rank, and J. Sommer, "Deos – the german robotics approach to secure and de-orbit malfunctioned satellites from low earth orbits," in *i-SAIRAS*, Sapporo, Japan, 2010.

[24] Y. Xu and T. Kanade, *Space Robotics: Dynamics and Control*, ser. The Springer International Series in Engineering and Computer Science. Springer, 1993.

[25] R. Smith, "Open dynamics engine (ode)," 2009. [Online]. Available: http://www.ode.org/

[26] S. G. Johnson, "The nlopt nonlinear-optimization package," 2011. [Online]. Available: http://github.com/stevengj/nlopt

[27] K. Yoshida, "The spacedyn: A matlab toolbox for space and mobile robots," *J. of the Soc. of Instrum. and Control Engineers*, vol. 38, no. 2, pp. 138–143, 1999.

[28] M. Kardynska and J. Smieja, "L1 and l2 norms in sensitivity analysis of signaling pathway models," in *2016 MMAR*, 2016, pp. 589–594.
[29] D. Calzolari, R. Lampariello, and A. Giordano, "Singularity maps of space robots and their application to gradient-based trajectory planning," in *Proc. of Robot.: Sci. and Syst.*, Oregon, USA, July 2020.