Task-oriented optimal dimensional synthesis of robotic manipulators with limited mobility

Matteo Russo[1], Luca Raimondi[1], Xin Dong[1], Dragos Axinte[1]*, James Kell[2]

[1]Faculty of Engineering, University of Nottingham, Nottingham NG8 1BB, UK
*Corresponding Author: dragos.axinte@nottingham.ac.uk

[2]Repair Technology, Rolls-Royce plc, PO Box 31, Derby DE24 8BJ, UK

Abstract: In this article, an optimization method is proposed for the dimensional synthesis of robotic manipulators with limited mobility, i.e. with less than 6 degrees-of-freedom ("DoF"), with a prescribed set of tasks in a constrained environment. Since these manipulators cannot achieve full 6-DoF mobility, they are able to follow only certain paths with prescribed position and orientation in space. While the most common approach to this problem employs pure path-planning algorithms, operations in narrow and complex environments might require changes to the robot design too. For this reason, this paper presents an improved approach which aims to minimize position and orientation error with a dimensional synthesis. First, a novel methodology that combines a path planning algorithm and dimensional synthesis has been proposed in order to optimize both robot geometry and pose for a given set of points. Then, the method is validated with a 4-DoF robot for high-precision laser operations in aeroengines as a case study. The example shows that the proposed procedure provides a stable algorithm with a high convergence rate and a short time to solution for robots with limited mobility in highly constrained scenarios.

Keywords: Optimization; Dimensional Synthesis; Kinematics; Manipulators.

1. Introduction

Despite having a reduced number of degrees of freedom ("DoF"), robotic manipulators with limited mobility are usually characterized by simple structure, lower cost, easy control and higher performance when compared to standard 6-DoF or redundant manipulators. A seminal example is given by the DELTA robot [1], a 3-DoF translational robot developed in the late 1980s that is widely used for industrial pick & place operations that require both high precision and high speed [2]. Furthermore, their lower mobility can be often compensated by a reconfigurable or modular design that can be adapted to the prescribed task, such as in [3-4]. Despite their advantages, manipulators with less than 6-DoF require a more careful design. In fact, standard 6-DoF and redundant manipulators only need to solve to a path-planning problem to perform most operations, whereas lower-mobility manipulators also require a task-oriented mechanism and dimensional synthesis.

In the last decades, many research groups worked on the geometrical optimization of robots with limited mobility, focusing on three main challenges: performance evaluation, optimization techniques and dimensional synthesis for a given workspace. However, defining a representative index for robot benchmarking is extremely difficult. Generic kinematic indices, such as the Global Conditioning Index [5], manipulability [6-7] and force transmission index [8-9] have been widely used as objective functions in robot optimization with many examples in literature [10-12]. Standard multi-objective optimization techniques such as weighted sum, constraint methods, goal attainment and genetic algorithms are usually employed [13], even if specific techniques tailored to robot designs have been studied, such as exhaustive map search [14] and game algorithms [15]. Overall, however, the optimization of generic objective functions can lead to results that are too general to significantly improve performance in a specific process or operation. On the other extreme, a task-oriented index can be developed to solve a one-off engineering problem but does not benefit a wider research field and cannot be used for performance comparison.

Dimensional synthesis, also known as kinematic synthesis, is an inverse problem in robot kinematics that aims to calculate the geometrical parameters of a manipulator to achieve a prescribed workspace [16]. While analytical methods can be used for simple robots with a limited number of bodies and DoF, a common approach to the dimensional synthesis is the definition of an optimization procedure with the robot geometry as only parameter and one or more of the aforementioned indices as objective function(s) [2, 16]. An extensive literature can be found on the dimensional synthesis of lower-mobility manipulators, with a focus on DELTA architectures [17-19] and novel designs [20-23]. Most of these works optimize robot geometry to either reach a prescribed workspace or to maximise performance in a given workspace with one or more kinetostatic indices as objective function.

Path planning involves an optimization procedure similar to the one used for dimensional synthesis, with joint values rather than robot geometry as parameters. In this case, the performance indices range from kinematics (e.g. time [24, 25], position/velocity [26], obstacle avoidance [27]) to dynamics (e.g. force and torque [28], energy [29, 30]). This optimization procedure is almost always performed after dimensional synthesis. Thus, a prescribed workspace volume is usually evaluated from operational requirement, then the robot geometry is optimized to reach the whole workspace volume, and finally the robot motion is computed for optimal performance around obstacles and along given path points. However, a robot geometry that is designed to achieve the best performance in a given volume might not be the optimal one to operate on a limited set of paths within the same workspace. In critical high-precision operations with strict constraints, this difference can be extremely significant.

For this reason, this work introduces a novel procedure for task-oriented dimensional synthesis of robotic manipulators with limited mobility. The proposed method computes robot geometry and joint values for a given set of paths while minimizing positioning and orientation error along them, solving both dimensional synthesis and path planning (with obstacle avoidance) at the same time. Then, this approach is validated on a relevant case study: the optimal dimensional synthesis of a lower-mobility manipulator for laser operations in aeroengines. This operational scenario requires extremely high precision (< 100 μm) along a complex path defined by the edge of an aerofoil, as well as strict geometrical constraints given by end-effector size, an access port with a diameter smaller than 9 mm and nearby aerofoil as obstacles. A single robot design cannot fulfil all the requirement; therefore, a modular design is proposed, made of a robot base with the actuators and the control hardware, and of a replaceable/disposable cable-driven end-effector mechanism with 2 degrees of freedom that is customized for a single operation on a given aerofoil and with a given laser tool. Thus, both motion and design parameters can change between different operations and applications. The modelling process needed to define the proposed optimization problem is described for the case study, and results are discussed in terms of accuracy, processing time and algorithm limitations.

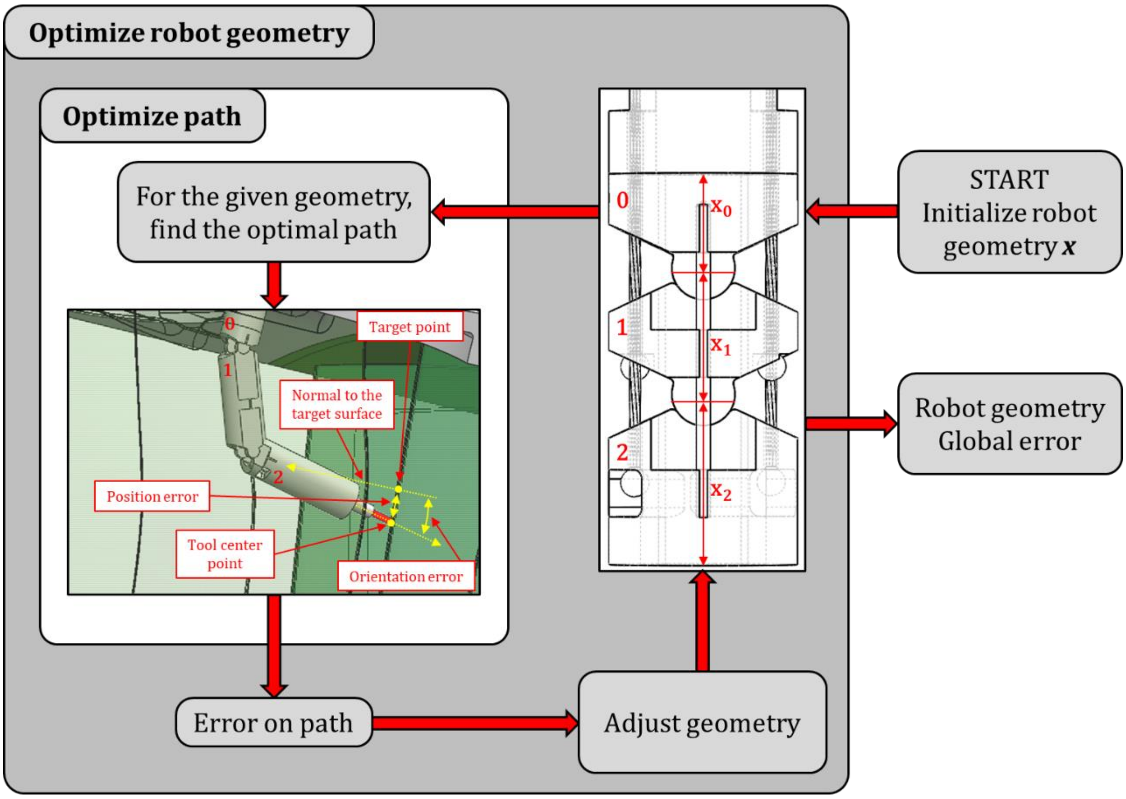
2. An optimization method for task-oriented dimensional synthesis

A procedure for the task-oriented dimensional synthesis of robotic manipulators with limited mobility is here proposed as based on an optimization problem that finds the best parameters to minimize the pose error over the prescribed task paths. In this section, the two main challenges in solving this problem are outlined: (i) the definition of an overall pose error that includes both position and orientation error, and (ii) the design of an algorithm that optimizes both geometrical parameters and motion variables, which can be summarized by the workflow in Fig. 1.

2.1 Problem definition

The standard form of an optimization problem is given by the minimization of an objective function $\boldsymbol{F}(\boldsymbol{r}) = [f_1(\boldsymbol{r})\ f_2(\boldsymbol{r}) \cdots f_n(\boldsymbol{r})]^T$ (with $\boldsymbol{F}: \mathbb{R}^m \to \mathbb{R}^n$), subject to disequality constraints $\boldsymbol{g}(\boldsymbol{r}) = \left[g_1(\boldsymbol{r})\ g_2(\boldsymbol{r}) \cdots g_p(\boldsymbol{r})\right]^T$ and equality constraints $\boldsymbol{h}(\boldsymbol{r}) = [h_1(\boldsymbol{r})\ h_2(\boldsymbol{r}) \cdots h_t(\boldsymbol{r})]^T$. The outcome of this minimization is given by the optimal values for parameter vector $\boldsymbol{r} = [r_1\ r_2 \cdots\ r_m]^T$. The numbers n, m, p and t describe respectively the number of objective functions, parameters, inequality constraints and equality constraints [13]. The optimization problem can be written as

$$\min \boldsymbol{F}(\boldsymbol{r}) = \min[f_1(\boldsymbol{r})\ f_2(\boldsymbol{r}) \cdots f_n(\boldsymbol{r})]^T \ \textit{subject to}\ \boldsymbol{g}(\boldsymbol{r}) \leq 0,\ \boldsymbol{h}(\boldsymbol{r}) = 0 \tag{1}$$



Optimize robot geometry
Optimize path
For the given geometry, find the optimal path
Target point
Normal to the target surface
Position error
Tool center point
Orientation error
0
1
2
x0
x1
x2
START Initialize robot geometry x
Robot geometry Global error
Error on path
Adjust geometry

Fig. 1: Diagram of the optimization workflow, with CAD models from the case study.

A multi-objective optimization problem is characterized by $n > 1$, and it often leads to contradictory results, since a single best solution that minimizes simultaneously all the objective functions does not exist. Thus, the optimal solutions (also known as Pareto front) can be computed as those solutions that cannot be improved in any of the objectives without degrading at least another one [13]. A solution $\boldsymbol{r_1}$ dominates another solution $\boldsymbol{r_2}$ when

$$f_i(\boldsymbol{r_1}) \leq f_i(\boldsymbol{r_2})\ \textit{for}\ i \in \{1, \dots, n\}\ \cap\ \exists\ k: f_k(\boldsymbol{r_1}) < f_k(\boldsymbol{r_2}) \tag{2}$$

A solution is optimal when no other solution dominates it. The proposed task-oriented dimensional synthesis is, in its most general form, a multi-objective optimization problem that minimizes the pose error function $\boldsymbol{e_{tot}}(\boldsymbol{r}) = \left[e_{pos}(\boldsymbol{r})\ e_{or}(\boldsymbol{r})\right]^T$, where $e_{pos}(\boldsymbol{r})$ is the position error function and $e_{or}(\boldsymbol{r})$ is the orientation error function. The robot parameter vector $\boldsymbol{r}$ contains both the geometrical parameters, defined as robot geometry vector $\boldsymbol{x}$, and the motion variables (such as joint angles) for the prescribed task waypoints, given by actuation vector $\boldsymbol{q}$. The disequality constraints are given by:

- Actuation limits on $\boldsymbol{q}$, due to minimum and maximum joint angles and collision avoidance between different bodies of the robot.
- Physical limits on $\boldsymbol{x}$, due to manufacturing and assembly variables such as maximum hole depth for milling and drilling operations.
- Environmental limits on $\boldsymbol{r}$, such as obstacle geometry for collision avoidance.

Equality constraints are applied to robot geometry $\boldsymbol{x}$ to represent arbitrary design choices or fixed dimensions, such as the size of commercial components.

2.2 Objective function

The most critical step in an optimization problem is the identification of an objective function able to model and evaluate the behaviour of the system while performing the prescribed tasks. Since this work focuses on high-precision operations, an index that represents overall pose error is used as objective function. However, the generality of the proposed procedure makes it suitable to be used with any performance index that is relevant to the task under analysis. The overall pose error of a robotic manipulator can be defined as the displacement between the actual pose of the manipulator, which can be defined as either a pose vector containing position and orientation coordinates or as the transformation matrix of the end-effector of the robot, and the target pose. Any path for the prescribed tasks can be discretized in a finite number of points, and the pose error for each of these can be evaluated. Then, a global error can be defined as the maximum error along the whole path as per Algorithm 1. By evaluating the worst performance of the manipulator in the provided tasks, the global error can be used as objective function for the proposed dimensional synthesis.

Algorithm 1: Global pose error

```
load path_points, joint_positions, robot_geometry
for i = 1 : path_point_number
    robot_pose_i=  forward_kinematics (robot_geometry, joint_position_i)
    pose_error_i = error_function(robot_pose_i, path_point_i)
end
global_pose_error = max (pose_errors)
```

When position-only or orientation-only errors are considered, the absolute spatial or angular displacement can be easily evaluated and used as objective functions. When both errors have to be minimized, two different procedures can be followed:

- Perform a multi-objective optimization with position error and orientation error as objective functions. A multi-objective optimization simplifies the definition of an objective function but poses further challenges: the procedure to solve the problem becomes significantly more complex, and the solution itself is given by multiple parameter sets on the Pareto front rather than a single one. Thus, further evaluations and data analysis would be needed to identify one optimal solution.
- Solve a standard optimization problem with an index that combines orientation and position errors as a single objective function. Despite the challenge of defining a single function that embeds both errors, this procedure results in a single optimal set of

parameters and allows for a more efficient and performing algorithm. For these reasons, a standard optimization is preferred in this research.

In the scientific literature on optimization, several methods have been proposed to combine multiple objective functions into one. Among the most successful ones, weighted sum, goal attainment and constraint method, [13], can be adapted to this case study as follow:

- Weighted sum is a scalar method that associates a weight w_i to each objective function f_i. The weight function is arbitrarily defined and can be written as:

  $$\boldsymbol{w} = [w_1, w_2]^T \ subject\ to\ w_i \geq 0\ \forall\ i \in [1,2],\ \sum_{i=1}^{2} w_i = 1 \tag{3}$$

  Then, a global objective function $\boldsymbol{E_{WS}}$ is obtained by summing the weighted objective functions as:

  $$E_{WS} = \boldsymbol{w}^T \cdot \boldsymbol{e_{tot}}(\boldsymbol{r}) = [w_1, w_2] \cdot \left[e_{pos}(\boldsymbol{r})\ e_{or}(\boldsymbol{r})\right]^T \tag{4}$$

  This method is very popular for its simplicity and it guarantees to find solutions on the Pareto front if the problem is convex.

- In the constraint method [13], the most relevant objective function is selected as global objective function, while the other objective functions are converted into constraint functions. The problem can be formulated for orientation as a constraint as:

  $$\min e_{pos}(\boldsymbol{r})\ subject\ to\ e_{or}(\boldsymbol{r}) \leq \epsilon \tag{5}$$

  where ϵ is the threshold value of the orientation error.

- Goal attainment consists in the definition of a target vector that contains the ideal values for all the objective functions. The purpose of this method is to optimize a coefficient that measures the deviation of each solution from the target vector. The method can be divided into three steps: definition of the vector with the desired position and orientation values; weighting of each value (as per weighted sum method); minimization of scalar deviation coefficients δ [13]. The optimization problem is mathematically defined as:

  $$\min \delta\ \ subject\ to\ e_i(\boldsymbol{r}) - w_i\delta \leq E_{GA,i}\ for\ i \in \{1, 2\} \tag{6}$$

  where $\boldsymbol{E_{GA}}(\boldsymbol{r}) = \left[E_{GA,pos}(\boldsymbol{r})\ E_{GA,or}(\boldsymbol{r})\right]^T$ is the vector with the ideal values and $\boldsymbol{w} = [w_1, w_2]^T$ is the weight function. The main advantage of this method is that it is able to generate solutions for a non-convex surface domain, whereas the previous one can fail.

According to task requirement, any of these methods can be selected. Three distinct cases can be identified: (i) when the task demands both errors to be within given limits, a goal attainment method is preferable, using the limits as target vector; (ii) if either orientation or position have a given limit and the other value should only be minimized, the constraint method is easier to define, with the known limit as constraint function; (iii) for the remaining cases, a weighted sum should be considered.

2.3 Optimization algorithm

Rather than running a single optimization with a large number of parameters that include both robot geometry and motion variables, this study proposes a more efficient approach in which the problem is broken down into two distinct optimization loops. First, a low-level loop evaluates the optimal actuation values for a given robot geometry. Then, a high-level loop performs the dimensional synthesis. As also outlined in [31-33], the advantages of a decomposition into two separated optimization problems can be summarized as:

- Separation of motion variables and geometrical parameters: the proposed two-level strategy allows for different requirements and tolerances for different kind of variables, improving the quality of the outcomes. The algorithm of each loop can be optimized for a different problem (low-level for path planning, high-level for dimensional synthesis) to improve its efficiency.
- Reduced number of optimization parameters: a single optimization would optimize a large number of parameters at the same time. Running two separate loops each with a different subset of those parameter increases code stability and robustness.

The pseudo-code for the low-level optimization is reported in Algorithm 2, while the high-level code is described in Algorithm 3.

Algorithm 2: Internal loop – path planning

```
load path_points, robot_geometry, actuation_constraints, obstacle_constraints
initialize joint_positions
optimization
    find joint_positions to minimize global_pose_error
    subject to actuation_constraints, obstacle_constraints
```

Algorithm 3: External loop – dimensional synthesis

```
load path_points, geometry_constraints, actuation_constraints, obstacle_constraints
initialize robot_geometry
optimization
    find robot_geometry  to minimize global_pose_error
    subject to joint_positions, geometry_constraints
```

3. Case study: a robotic system for laser processes

In this section, the optimization of a 4-DoF robotic manipulator for in-situ laser processing of the aerofoils of an aeroengine is reported as an example of the proposed methodology; in this application the robot needs to enter the engine via a narrow access port, navigate to the required workspace and perform an accurate path against the target aerofoil. The whole operation needs to be performed without disassembling the engine from the aircraft frame. This application is characterized by a complex environment with narrow access and limited mobility due to the need to navigate and operate in an aeroengine. The laser process requires the tip of the robot to follow a path along an aerofoil edge with high precision (position error < 100 μm). During the operation, the distance between the robot and the aerofoil edge must be constant and equal to the focal length of the laser. Furthermore, the laser should be normal to the edge, with a maximum allowed orientation error of approximately 10 deg. Conventional optimization methods try to maximize the performance of the given kinematic architecture in a given workspace, which results in a robot with overall better performance but worse accuracy on the prescribed path. The addition of more degrees of freedom would make the robot too bulky to perform the task, whereas the proposed method converges to a solution.

3.1 Background

Laser processing for the generation of small and accurate geometries in narrow workspaces is characterized by a complex set of constraints, given by the process itself, the environment and robot specifications. The main requirements of the robot for this case study are:

- **Position (process):** the maximum position error is 100 μm.
- **Alignment (process):** the maximum orientation error is 10 deg between the laser beam direction and the normal to the aerofoil edge.
- **Laser focus (process):** the distance between the aerofoil edge and the end-effector of the robot must be equal to the focal distance of the laser, so that the laser beam is focused onto the target surface during the operation (e.g. cutting, drilling, engraving).
- **Speed (process):** the robot must be able to follow the prescribed path with a constant velocity. The velocity value depends on the operation, but it is usually in the range from 0.1 to 10 mm/s, which can be easily achieved by commercial actuators.
- **Access (environment):** the robot must be able to access the engine through a circular port with a diameter of 9 mm.
- **Collisions (environment):** the robot must be able to navigate and operate with no collisions.
- **End-effector size (robot):** the robot must be able to operate an end-effector of cylindrical shape with a length of 20 mm.
- **Robot shape (robot):** the centreline of the robot body has a minimum bending radius of 10 mm, in order to avoid damage to the cable of the laser end-effector.
- **Other:** Further constraints are given by materials, components and manufacturing techniques for the chosen design (e.g. minimum wall thickness, actuator torque/force, minimum hole size, maximum hole lengths, manufacturing and assembly tolerances).

In order to achieve the desired performance, a 4-DoF robot with a tendon-driven compliant serial-parallel architecture has been selected. The proposed design is an adaptation of the REINER robot in Fig. 2a, developed for a similar application (boreblending) on gas turbine blades [34-35]. REINER is composed of a 2-DoF compliant tip mechanism (Fig.2b) and a 2-DoF actuation pack (Fig. 2c). The actuation pack is equipped with five linear motors and a rotational motor to control the stages of the base of the manipulator and the four steel cables that actuate the compliant tip mechanism. To perform the desired operation in-situ, the actuation pack can be attached to aeroengines as per Figs. 2c and 2d.

The proposed design combines the original actuation pack with a new modular tip mechanism design, which can be optimized for operations on a single aerofoil geometry. In this way, a different interchangeable tip can be designed for different geometries, to ensure the best performance on each task. Thus, the robot can be described as a hybrid serial-parallel mechanism with a $\underline{\text{RP}}$(2S$\underline{\text{P}}$S-C)(2S$\underline{\text{P}}$S-C) architecture, with the rotary (R) and linear (P) stages of the actuation pack, as per Fig. 3a, in series with two (2S$\underline{\text{P}}$S-C) tendon-driven compliant (C) sections, where each cable in tension is modelled as an (S$\underline{\text{P}}$S) serial chain, with the varying length represented by an actuated prismatic (P) joint between two spherical (S) joints at the cable routing points of each disk, as per Fig. 3b. The tip mechanism design is explained in Section 3.2, while further details on the kinematic chain can be found in Section 3.3. This solution is able to perform high-precision operations as well as navigate through the narrow access port and constrained environment of an aeroengine.

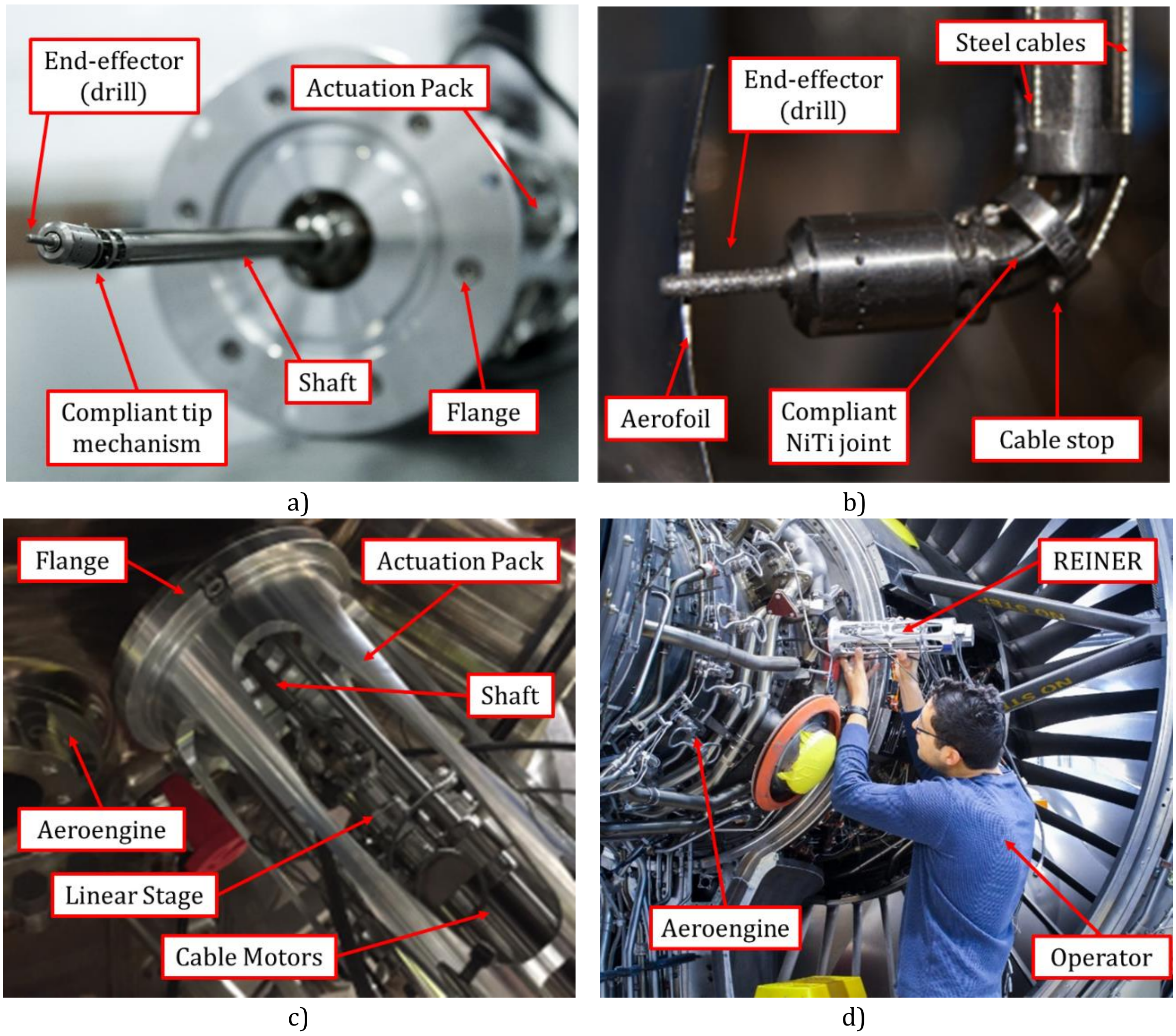
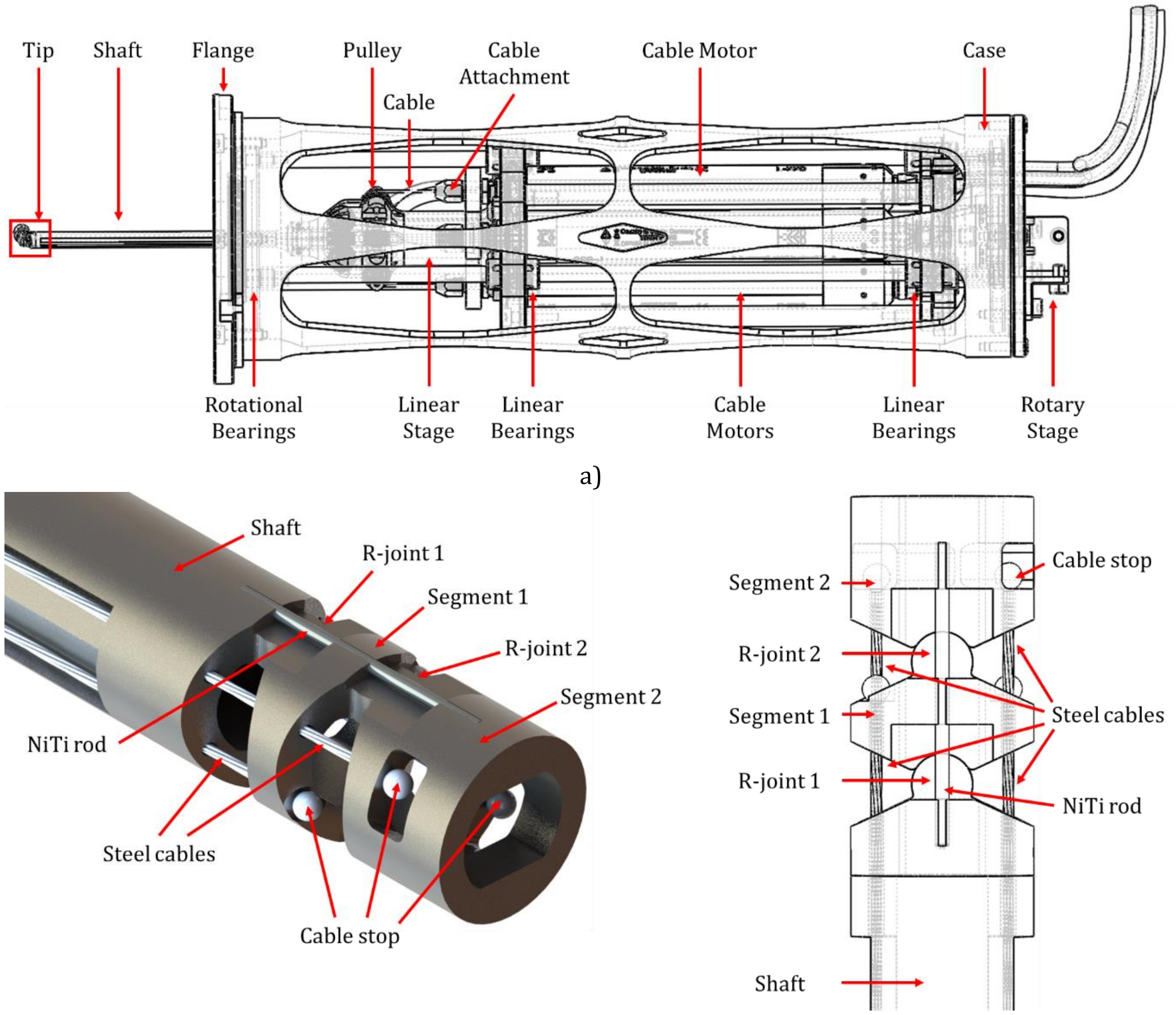


Fig. 2: The REINER robot for in-situ teleoperated boreblending of gas turbine blades [34-35]: a. Robot prototype; b. Tip mechanism and end-effector; c. Actuation pack; d. Installation on an aeroengine.

3.2 Design solution

A design solution for the interchangeable tip design is shown in Fig. 3b. The rigid bodies (shaft, segment 1 and segment 2) are manufactured in titanium as hollow cylinders. The central hole is used for the delivery of cables for the laser-equipped end-effector. While in the REINER design the bending of the robot was controlled by the compliance of the nickel-titanium (NiTi) backbone as in Fig. 2b, sliding surfaces kinematically equivalent to rotational joints (R-joint) are used in this case study to guide the relative motion of the segments. This solution improves the accuracy of the kinematic model, since the real bending of the NiTi rods is characterized by a non-constant curvature [36-37]. Despite this change, the NiTi backbone has been kept in this case study for robot safety: in case of cable breakage, the stiffness of the NiTi rod ensures that the manipulator returns to its straight configuration rather than moving in an uncontrolled way in the aeroengine, risking to damage the aerofoil being processed by the laser. Furthermore, they also dampen dynamic irregularities in the motion and support the mechanical assembly when the cables are not tensioned. The steel cables are routed through the segments up as shown in Fig. 3 and each of them ends with a spot-welded bead with a diameter larger than the hole in the disk, which enables the actuation of the segments without additional locking systems.



a)

b)

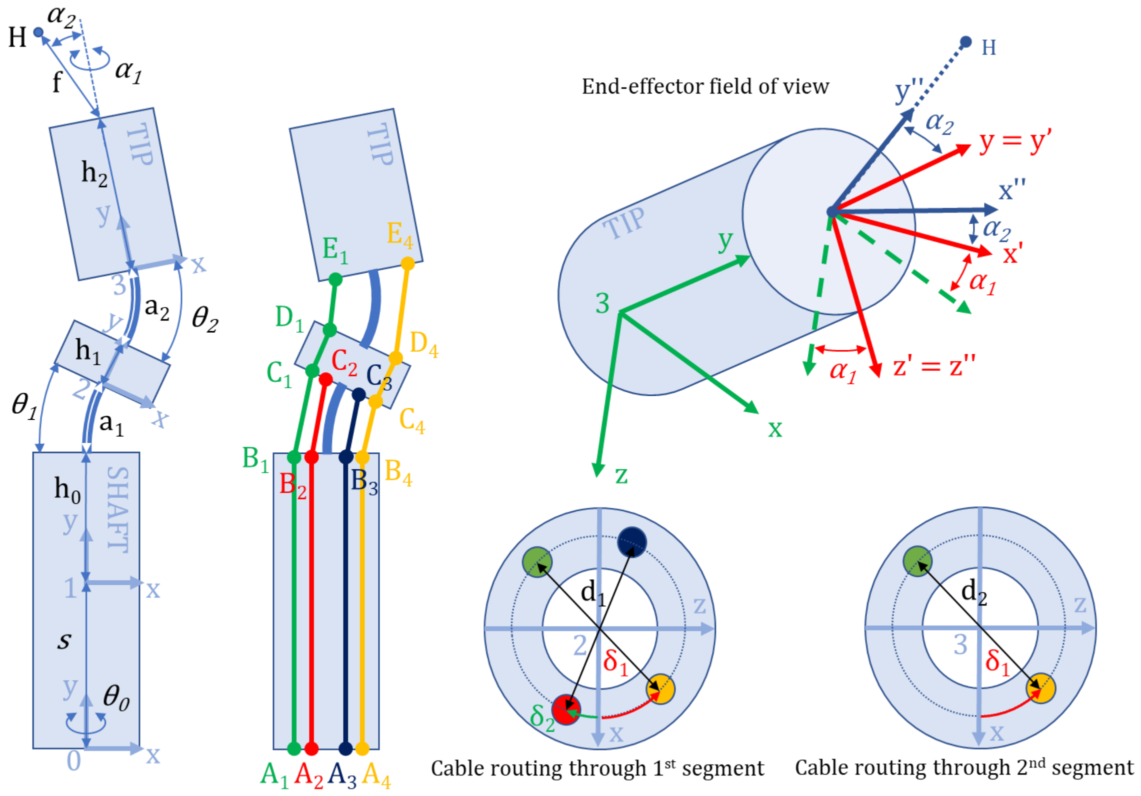
Fig. 3: Proposed design solution: a) Actuation pack design; b) Tip mechanism.

3.3 Robot model

A simplified kinematic scheme for the (2S$\underline{\text{P}}$S-C)(2S$\underline{\text{P}}$S-C) tip design is shown in Fig. 4. A complete list of the models' parameters is reported in Table 1. The mechanism is made of three hollow cylinders (shaft, first segment and second segment) connected in series by twin compliant backbones that allow bending on one plane only. The shaft is directly driven by the actuation pack, which controls its rotation around the y-axis ("twist angle") with the rotational stage (θ_0) and its displacement on the y-axis with the linear stage (s). The orientation of the first segment with respect to the shaft is described by bending angle θ_1, while the orientation of the second segment with respect to the first one by θ_2. The lengths of the segments are given by geometrical parameters h_0, h_1 and h_2, while the diameters of the cable guide centres on each of them are respectively d_0, d_1 and d_2. The lengths of the compliant backbones are a_1 and a_2. The curvature of each backbone is controlled by a pair of cables, cables 2 and 3 for the first joint, cables 1 and 4 for the second joint, whose routing can be seen in Fig. 4 and is determined by a diameter and phase for each segment. Each cable is modelled as S$\underline{\text{P}}$S serial kinematic chains with a variable length controlled by the virtual prismatic actuator. This assumption requires the cables to always be tensioned. For this optimization, the laser beam is modelled as a straight virtual link that starts at the end of the second segment with an orientation defined by α_1 and α_2 and a length equal to the laser focal distance. Point H represents the laser focal point and should follow the aerofoil edge during the laser operation, while the orientation of the virtual link (also called "laser field of view") should match the normal of the aerofoil edge to meet the orientation requirements.

Table 1: List of kinematic parameters of the proposed case study

Par	Description	Par	Description
a_1	Arc length of 1st compliant joint	d_0	Diameter of cable guide on shaft
a_2	Arc length of 2nd compliant joint	d_1	Diameter of cable guide on 1st segment
h_0	Length of shaft	d_2	Diameter of cable guide on 2nd segment
h_1	Length of 1st segment	δ_1	Phase of cables 1-4 wrt bending plane
h_2	Length of 2nd segment	δ_2	Phase of cables 2-3 wrt bending plane
α_1	1st laser orientation parameter	l_1	Length of 1st cable (green)
α_2	2nd laser orientation parameter	l_2	Length of 2nd cable (red)
f	Laser focal length	l_3	Length of 3rd cable (blue)
θ_0	Twist angle	l_4	Length of 4th cable (yellow)
s	Linear displacement	A_i	Cable attachment point to the motor
θ_1	Bending angle of 1st compliant joint	B_i	Cable exit point from base disk
θ_2	Bending angle of 2nd compliant joint	C_i	Cable entrance point into 1st disk
H	Laser focal point	D_i	Cable exit point from 1st disk
O	Base reference point	E_i	Cable attachment point to 2nd disk



Cable routing through 1st segment

Cable routing through 2nd segment

Fig. 4: Kinematic scheme of the proposed robotic system

With reference to the scheme in Fig. 4, the kinematic problem can be solved with a continuous curvature approach by analysing the motion of the robot in cartesian, configuration and actuation space [36-37]. The pose of the robot can be defined in the cartesian space by the position and orientation of point H with respect to base frame 0xyz of Fig. 4. In configuration space, each pose is defined by motion variables θ_0, s, θ_1 and θ_2, linked to the 4 DoFs of the robot. The Forward Kinematics from configuration to cartesian space can be written as

$$ {}^{\mathbf{0}}\boldsymbol{H} = {}^{\mathbf{0}}\boldsymbol{T}_{\mathbf{1}}\, {}^{\mathbf{1}}\boldsymbol{T}_{\mathbf{2}}\, {}^{\mathbf{2}}\boldsymbol{T}_{\mathbf{3}}\, {}^{\mathbf{3}}\boldsymbol{T}_{\boldsymbol{H}}(0 \quad 0 \quad 0 \quad 1)^T \tag{7} $$

The mathematical expression for the transformation matrices in Eq. (7) are reported in Table 2. These equations can be adapted to the R-joint design presented in Section 3.2 with the assumption a_1 = 0; a_2 = 0. This choice improves the efficiency of the optimization and control algorithms by simplifying the equations in Table 2 and avoiding some singularities in the Inverse Kinematics from cartesian to configuration space.

When defining a pose in the actuation space, the first two variables can be defined again by θ_0 and s, since the twist angle and the linear displacement are controlled directly by the rotational and linear stage motors. The remaining actuation variables are the length of the four steel cables l_1, l_2, l_3 and l_4, controlled by the remaining four linear motors in the actuation pack, for a total of six actuation variables. With reference to the cable routing geometry in Fig. 4, when the configuration variables are known, the length of the cables can be computed as

$$ l_i = \begin{cases} \overline{A_\iota B_\iota} + \overline{B_\iota C_\iota} + \overline{C_\iota D_\iota} + \overline{D_\iota E_\iota}\, for\ i = \{1, 4\} \\ \overline{A_\iota B_\iota} + \overline{B_\iota C_\iota}\, for\ i = \{2, 3\} \end{cases} \tag{8} $$

The cable layout parameters for the evaluation of Eq. (8) are provided in Table 3.

Table 2: Transformation matrices for Forward Kinematics (configuration to cartesian space).

Matrix	Equation
${}^{\mathbf{0}}\boldsymbol{T}_{\mathbf{1}}$	$\begin{bmatrix} \cos\theta_0 & 0 & \sin\theta_0 & 0 \\ 0 & 1 & 0 & s \\ -\sin\theta_0 & 0 & \cos\theta_0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
${}^{\mathbf{1}}\boldsymbol{T}_{\mathbf{2}}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & h_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & \theta_1{}^{-1}a_1(cos\,\theta_1 - 1) \\ \sin\theta_1 & \cos\theta_1 & 0 & \theta_1{}^{-1}a_1\,sin\,\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
${}^{\mathbf{2}}\boldsymbol{T}_{\mathbf{3}}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & h_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & \theta_2{}^{-1}a_2(cos\,\theta_2 - 1) \\ \sin\theta_2 & \cos\theta_2 & 0 & \theta_2{}^{-1}a_2\,sin\,\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
${}^{\mathbf{3}}\boldsymbol{T}_{\boldsymbol{H}}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & h_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha_1 & 0 & \sin\alpha_1 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha_1 & 0 & \cos\alpha_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha_2 & -\sin\alpha_2 & 0 & 0 \\ \sin\alpha_2 & \cos\alpha_2 & 0 & f \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

3.4 Optimization setup

Eight kinematic parameters have been selected for the optimization: four of them are the main geometrical variables of the robot (h_1, h_2, α_1, α_2), given by the length of the segments and the field of view of the laser, and they are optimized by the high-level loop; the remaining four are the motion variables in configuration space (θ_0, s, θ_1, θ_2), optimized by the low-level loop. These parameters are summarized in Table 4. All the remaining kinematic parameters have been chosen according to design and manufacturing. The focal distance of the laser has been chosen according to the available devices. Environmental constraints have been extracted from a CAD model of an aeroengine.

Table 3: Position vectors for Inverse Kinematics (configuration to actuation space)

Par	Position vector	Par	Position vector
$^{\mathbf{1}}\boldsymbol{A_1}$	$\left(-\frac{d_0\cos\delta_1}{2} \quad 0 \quad -\frac{d_0\sin\delta_1}{2} \quad 1\right)^T$	$^{\mathbf{2}}\boldsymbol{C_1}$	$\left(-\frac{d_1\cos\delta_1}{2} \quad 0 \quad -\frac{d_1\sin\delta_1}{2} \quad 1\right)^T$
$^{\mathbf{1}}\boldsymbol{A_2}$	$\left(\frac{d_0\cos\delta_2}{2} \quad 0 \quad -\frac{d_0\sin\delta_2}{2} \quad 1\right)^T$	$^{\mathbf{2}}\boldsymbol{C_2}$	$\left(\frac{d_1\cos\delta_2}{2} \quad 0 \quad -\frac{d_1\sin\delta_2}{2} \quad 1\right)^T$
$^{\mathbf{1}}\boldsymbol{A_3}$	$\left(-\frac{d_0\cos\delta_2}{2} \quad 0 \quad \frac{d_0\sin\delta_2}{2} \quad 1\right)^T$	$^{\mathbf{2}}\boldsymbol{C_3}$	$\left(-\frac{d_1\cos\delta_2}{2} \quad 0 \quad \frac{d_1\sin\delta_2}{2} \quad 1\right)^T$
$^{\mathbf{1}}\boldsymbol{A_4}$	$\left(\frac{d_0\cos\delta_1}{2} \quad 0 \quad \frac{d_0\sin\delta_1}{2} \quad 1\right)^T$	$^{\mathbf{2}}\boldsymbol{C_4}$	$\left(\frac{d_1\cos\delta_1}{2} \quad 0 \quad \frac{d_1\sin\delta_1}{2} \quad 1\right)^T$
$^{\mathbf{1}}\boldsymbol{B_1}$	$\left(-\frac{d_0\cos\delta_1}{2} \quad h_0 \quad -\frac{d_0\sin\delta_1}{2} \quad 1\right)^T$	$^{\mathbf{2}}\boldsymbol{D_1}$	$\left(-\frac{d_1\cos\delta_1}{2} \quad h_1 \quad -\frac{d_1\sin\delta_1}{2} \quad 1\right)^T$
$^{\mathbf{1}}\boldsymbol{B_2}$	$\left(\frac{d_0\cos\delta_2}{2} \quad h_0 \quad -\frac{d_0\sin\delta_2}{2} \quad 1\right)^T$	$^{\mathbf{2}}\boldsymbol{D_4}$	$\left(\frac{d_1\cos\delta_1}{2} \quad h_1 \quad \frac{d_1\sin\delta_1}{2} \quad 1\right)^T$
$^{\mathbf{1}}\boldsymbol{B_3}$	$\left(-\frac{d_0\cos\delta_2}{2} \quad h_0 \quad \frac{d_0\sin\delta_2}{2} \quad 1\right)^T$	$^{\mathbf{3}}\boldsymbol{E_1}$	$\left(-\frac{d_2\cos\delta_1}{2} \quad 0 \quad -\frac{d_2\sin\delta_1}{2} \quad 1\right)^T$
$^{\mathbf{1}}\boldsymbol{B_4}$	$\left(\frac{d_0\cos\delta_1}{2} \quad h_0 \quad \frac{d_0\sin\delta_1}{2} \quad 1\right)^T$	$^{\mathbf{3}}\boldsymbol{E_4}$	$\left(\frac{d_2\cos\delta_1}{2} \quad 0 \quad \frac{d_2\sin\delta_1}{2} \quad 1\right)^T$

By using Eqs. (1), two different optimization problems can be defined according to Algorithms 2 and 3 in Section 2.3. The optimization problem of the internal loop for path planning, introduced in Algorithm 2, can be written as

$$\min e_{pos}(\boldsymbol{\theta_0}, \boldsymbol{s}, \boldsymbol{\theta_1}, \boldsymbol{\theta_2}) \; subject \; to \; \boldsymbol{g}(\boldsymbol{\theta_0}, \boldsymbol{s}, \boldsymbol{\theta_1}, \boldsymbol{\theta_2}) \leq \mathbf{0} \tag{9}$$

with disequality constraints $\boldsymbol{g}(\boldsymbol{\theta_0}, \boldsymbol{s}, \boldsymbol{\theta_1}, \boldsymbol{\theta_2}) = [\boldsymbol{g_{ob}}, \boldsymbol{g_{act}}, e_{or} - \epsilon\,]^T$, where $\boldsymbol{g_{ob}}$ represents obstacle avoidance constraints, $\boldsymbol{g_{act}}$ contains the minimum and maximum values of each motion parameter, and the term $e_{or} - \epsilon$ requires the orientation error e_{or} to be lower than a given tolerance ϵ. Since the tolerance on the position error is stricter than the one on the orientation error, a constraint method as per Eq. (5) has been chosen to define position error as objective function and orientation error as inequality constraint.

The outer surface of the robot can be discretized as a point cloud, defined as a set S = {$\boldsymbol{P_i}$} of points $\boldsymbol{P_i}$. Similarly, the outer surfaces of the obstacles (e.g. aerofoil, walls of the work environment) can be defined as a point cloud O = {$\boldsymbol{Lj}$}. Thus, the obstacle avoidance constraint $\boldsymbol{g_{ob}}$ requires the distance between two generic points $\boldsymbol{P_i}$ and $\boldsymbol{L_j}$ to be equal to, or greater than, a prescribed distance d. This constraint can be expressed as

$$\left|\boldsymbol{P_i} - \boldsymbol{L_j}\right| \geq d \; \forall \; \boldsymbol{P_i} \in S, \boldsymbol{L_j} \in O \tag{10}$$

The actuation constraint $\boldsymbol{g_{act}}$ defines the limits of the motion variables in configuration space $(\theta_0, s, \theta_1, \theta_2)$ due to physical constraints such as actuator limits and collision between multiple bodies of the robot. These limits are defined by the minimum value $\left(\theta_{0,min}, s_{min}, \theta_{1,min}, \theta_{2,min}\right)$ and the maximum value $\left(\theta_{0,max}, s_{max}, \theta_{1,max}, \theta_{2,max}\right)$ of the motion variables. Therefore, $\boldsymbol{g_{act}}$ can be written as

$$\left(\theta_{0,min}, s_{min}, \theta_{1,min}, \theta_{2,min}\right) \leq (\theta_0, s, \theta_1, \theta_2) \leq \left(\theta_{0,max}, s_{max}, \theta_{1,max}, \theta_{2,max}\right) \tag{11}$$

The optimization problem of the external loop for dimensional synthesis, as per Algorithm 3, can be defined as

$$\min e_{pos}(h_1, h_2, \alpha_1, \alpha_2) \; subject \; to \; \boldsymbol{g}(h_1, h_2, \alpha_1, \alpha_2) \leq \mathbf{0}, \;\; \boldsymbol{h}(h_1, h_2, \alpha_1, \alpha_2) = \mathbf{0} \tag{12}$$

with equality constraints $\boldsymbol{h}$ determined by the motion parameters computed by the internal loop with Eq. (9), and disequality constraints $\boldsymbol{g}(h_1, h_2, \alpha_1, \alpha_2) = \left[\boldsymbol{g_{ob}}, \boldsymbol{g_{geom}}, e_{or} - \epsilon\,\right]^T$, where $\boldsymbol{g_{ob}}$

represents obstacle avoidance constraints, $\boldsymbol{g}_{geom}$ contains the minimum and maximum values of each design parameter, and the term $e_{or} - \epsilon$ requires the orientation error e_{or} to be lower than a given tolerance ϵ. The formulation of these constraints is equivalent to the one of the internal loop for path planning.

The code has been implemented in MATLAB on a Windows 10 computer with a 2.60 GHz quad-core CPU (Intel i7-6700HQ) by using the in-built functions of the MATLAB Optimization Toolbox. To verify its robustness, the algorithm was run for 10 different aerofoil geometries, each with 10 different values of the laser focal distance, for a total of 100 cases.

Table 4: List of parameters for the proposed optimization procedure

Par	Description	Par	Description
h_1	Length of 1st segment	θ_0	Twist angle
h_2	Length of 2nd segment	s	Linear displacement
α_1	1st laser orientation parameter	θ_1	Bending angle of 1st compliant joint
α_2	2nd laser orientation parameter	θ_2	Bending angle of 2nd compliant joint

4. Results and discussion

The algorithm converged to a desirable solution on 76% of the 100 cases, with a maximum estimated position error equal to $1.93 \cdot 10^{-6}$ mm. The longest time to convergence was equal to 588.72 s. The overall performance of the algorithm, as reported in Table 5, was satisfying, with most of the cases converging to a solution. Considering a variable laser focal length, 9 out of 10 aerofoil edges can be followed successfully for the laser operation under study.

Table 5: Algorithm performance

Overall Performance			
Algorithm convergence		**Causes of failure**	
Convergence ratio	92%	Collisions	11%
Success ratio	76%	Bending radius	8%
Feasible paths	9/10	Error	6%
Performance on Feasible Paths			
Parameter	**Minimum**	**Average**	**Maximum**
Solution time	178.37 s	272.79 s	588.72 s
Maximum error	$3.15 \cdot 10^{-7}$ mm	$1.93 \cdot 10^{-6}$ mm	$4.26 \cdot 10^{-5}$ mm

One of the cases is here reported as an example, with the numerical solution for both design and motion parameters shown in Table 6. The aerofoil edge has been discretized in 16 points, and the optimal motion parameters for each point are computed with a maximum estimated error of $5.51 \cdot 10^{-7}$ mm. Four screenshots of a CAD simulation of the outcome motion and geometry can be observed in Fig. 5, and the convergence of the algorithm in the example is illustrated in the objective function value over iteration plot in Fig. 6.

The main scenarios in which the proposed algorithm converges to a technically unfeasible solution occur when the algorithm overrides a constraint to find a solution in one of the following scenarios: one of the joint is forced to move out of its limits (11% of the total cases), causing a collision between the two segments of the tip mechanism; the overall shape of the robot damages the laser cable due to a steep bending radius (8% of the total cases), as exemplified in Fig. 7. This kind of errors are not typical of standard optimization problems and are likely to be caused by the decomposition of the procedure in nested loops, which can decrease the reliability of the computed results by converging on unfeasible parameter sets.

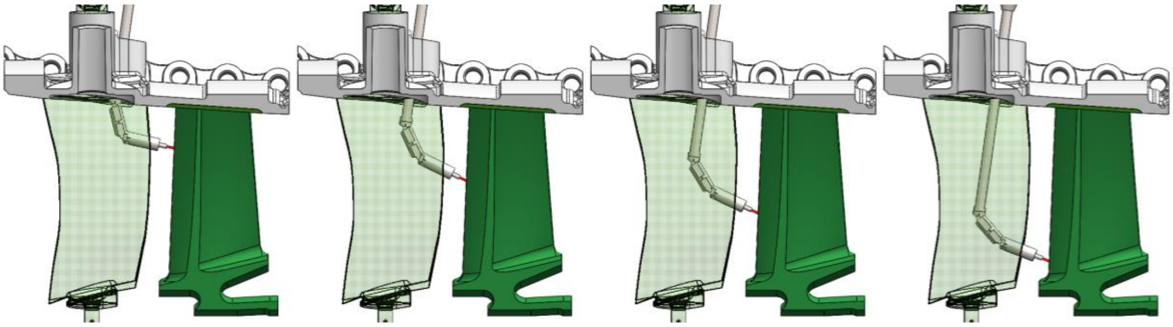


Fig. 5: The proposed robot following an aerofoil edge in a CAD simulation.

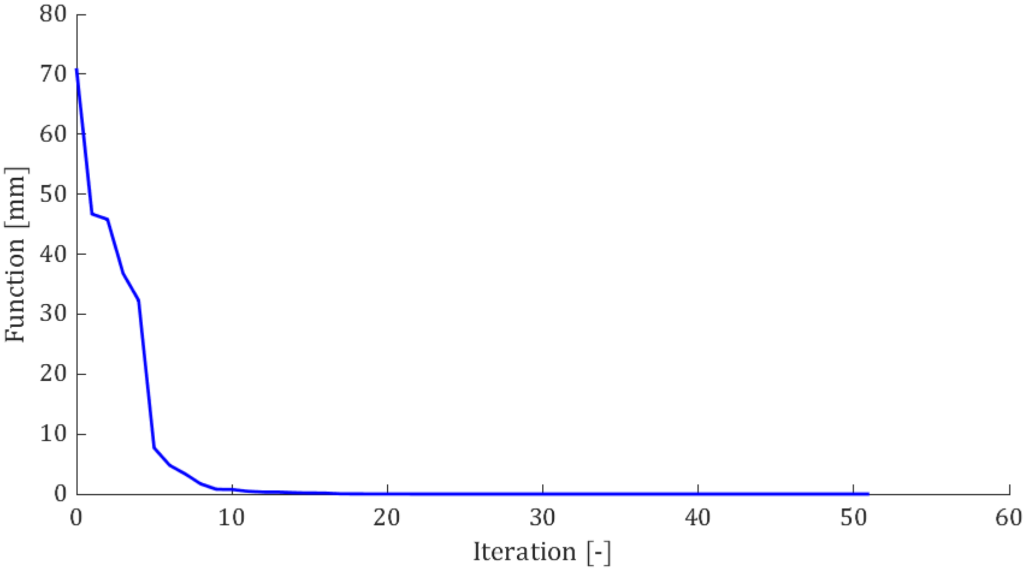


Fig. 6: Convergence of the proposed optimization algorithm in the case reported in Table 6: value of the objective function over number of iterations of the external loop (Algorithm 3).

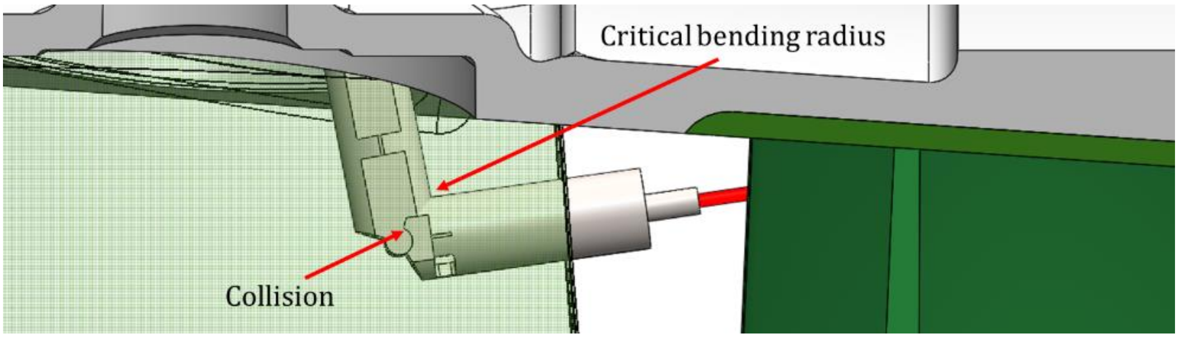


Fig. 7: An example of algorithm failure: collision between the segments and critical bending radius for the laser cables.

Another common cause of failure is given by a position error higher than the allowed tolerance along the path (6% of the total cases). Furthermore, in 8 cases the algorithm could not converge to a solution, due to the combination of either a long laser focal distance with a small aerofoil geometry, or a short focal distance with a large environment. However, by adopting a laser with a different focal distance, these algorithm failures can be circumvented to enable laser processing of all the aerofoil samples within the prescribed requirements.

Table 6: Example solution for the aerofoil in Fig. 5.

DESIGN PARAMETERS				
***h_1* [mm]**	***h_2* [mm]**	***α_1* [rad]**	***α_2* [rad]**	***f* [mm]**
13.96431	17.13120	0.21305	0.553706	10.00000
MOTION PARAMETERS				
Point	***θ_0* [rad]**	***s* [mm]**	***θ_1* [rad]**	***θ_2* [rad]**
1	-0.08434	0.151085	0.104741	1.567584
2	-0.08036	3.392616	0.11258	1.521151
3	-0.07643	6.483351	0.120522	1.468779
4	-0.07262	7.019522	0.105833	1.337489
5	-0.06896	5.173678	0.147325	1.052155
6	-0.06537	6.179534	0.2316	0.829653
7	-0.06182	8.637112	0.31293	0.663734
8	-0.05825	11.9714	0.380433	0.546941
9	-0.05461	15.96767	0.42714	0.482248
10	-0.05081	20.9731	0.429995	0.516349
11	-0.04682	25.20861	0.467919	0.47224
12	-0.04258	29.86218	0.48898	0.467329
13	-0.03809	34.7568	0.501854	0.483021
14	-0.03339	39.15063	0.53683	0.448246
15	-0.02873	43.9812	0.55623	0.452057
16	-0.02347	49.00459	0.602123	0.402437
PERFORMANCE				
Solution Time			395.52 s	
Maximum Error			$5.51 \cdot 10^{-7}$ mm	

Alternative optimization setups (multi-objective, weighted function, goal attainment) have been considered for performance comparison, but have proved unreliable for the case study. Multi-objective methods (e.g. genetic algorithms) prioritize either orientation or position, often leading to outcomes with good results on only one of the objective functions with long times to solution. Weighted function outcomes depend on the arbitrary definition of the weight vector. When a greater weight is given to the position error, the orientation error is neglected and vice versa, leading to a very low success ratio (12%) on the proposed cases. While more refined and robust solution searching methods can be used to improve convergence, the proposed optimization method can be employed independently from the kind of solver.

The proposed algorithm has been compared to a standard optimization procedure, which solves the problem by computing both design and motion parameters within a single optimization problem rather than by using nested loops with decoupled variables. This standard optimization can be expressed as

$$\min e_{pos}(\boldsymbol{\theta_0}, \boldsymbol{s}, \boldsymbol{\theta_1}, \boldsymbol{\theta_2}, h_1, h_2, \alpha_1, \alpha_2). \tag{13}$$

The significantly higher number of variables leads the conventional optimization to stop at local minima, often converging to a result with a position error higher than the allowed tolerance. Thus, by decoupling synthesis and path planning, the algorithm introduced in this paper is characterized by a better convergence rate: the same problem solved with a conventional minimization procedure converges to a feasible solution on 52% of the total cases, with significantly longer times to solution (more than 1000 s).

5. Conclusions

Dimensional synthesis and path planning are widely regarded as two distinct problems that are both required to optimize a robot's performance: the dimensional synthesis defines the geometry of the robot, while the path planning finds the optimal actuation value to perform a prescribed task. However, when operating in highly constrained scenarios and with limited degrees of freedom, dimensional synthesis alone might result in suboptimal parameters for path planning, and vice versa. To address this issue, this paper presented a novel approach for the task-oriented dimensional synthesis of lower-mobility robotic manipulators, which is based on a dual multi-objective optimization that minimizes position and orientation error on prescribed paths by solving a path planning problem with a low-level optimization loop and the dimensional synthesis of the robot with a high-level loop. The main results of this research can be summarized as:

- A novel optimization method to solve both dimensional synthesis and path planning, which focuses on robots with limited mobility in highly constrained scenarios and provides a stable algorithm with a high convergence rate and a short time to solution when compared to the alternatives.
- Validation of the proposed algorithm on a case study, where the novel method is successfully applied to a challenging aerospace scenario for the optimization of a mechanism for in-situ laser processing of aerofoil in aeroengine, characterized by strict process constraints, an extremely narrow and complex geometry and the need for high precision (0.1 mm).
- A comparative analysis of optimization setups, through which convergence was achieved 76% of the times, compared with a 12% success ratio of other setups.

Overall, the novel algorithm proved to be a fast and powerful tool to solve task-oriented dimensional synthesis problems, performing significantly better than other approaches on lower-mobility manipulators in highly constrained environments.

Acknowledgments: This work was funded by Rolls-Royce plc, by UK EPSRC project "Robotics and Artificial Intelligence for Nuclear" (RAIN), grant EP/R026084/1, and by UK EPSRC project "Through-life performance: From science to instrumentation", grant EP/P027121/1. The authors would like to acknowledge Dr. David Alatorre Troncoso and Dr. Amir Rabani for their support and work on the project.

6. References

[1] Rey L., Clavel R. (1999). The delta parallel robot. In Parallel Kinematic Machines (pp. 401-417). Springer, London.

[2] Merlet J.P. (2006). Parallel robots (Vol. 128). Springer Science & Business Media.

[3] Huang T., Li M., Zhao X.M., Mei J.P., Chetwynd D.G., Hu, S.J. (2005). Conceptual design and dimensional synthesis for a 3-DOF module of the TriVariant-a novel 5-DOF reconfigurable hybrid robot. IEEE Transactions on Robotics, 21(3), 449-456.

[4] Sun T., Song Y., Li Y., Zhang J. (2010). Workspace decomposition based dimensional synthesis of a novel hybrid reconfigurable robot. Journal of Mechanisms and Robotics, 2(3).

[5] Gosselin C., Angeles J. (1991). A global performance index for the kinematic optimization of robotic manipulators. Journal of mechanical design (1990), 113(3), 220-226.

[6] Yoshikawa T. (1985). Manipulability of robotic mechanisms. The international journal of Robotics Research, 4(2), 3-9.

[7] Doty K.L., Melchiorri C., Schwartz E.M., Bonivento C. (1995). Robot manipulability. IEEE Transactions on Robotics and Automation, 11(3), 462-468.

[8] Takeda Y., Funabashi H. (2001). A transmission index for in-parallel wire-driven mechanisms. JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing, 44(1), 180-187.

[9] Chen C., Angeles J. (2007). Generalized transmission index and transmission quality for spatial linkages. Mechanism and Machine Theory, 42(9), 1225-1237.

[10] Gao Z., Zhang D., Hu X., Ge Y. (2010). Design, analysis, and stiffness optimization of a three degree of freedom parallel manipulator. Robotica, 28(3), 349-357.

[11] Kelaiaia R., Company O., Zaatri A. (2012). Multiobjective optimization of a linear Delta parallel robot. Mechanism and Machine Theory, 50, 159-178.

[12] Russo M., Ceccarelli M., Takeda Y. (2018). Force transmission and constraint analysis of a 3-SPR parallel manipulator. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 232(23), 4399-4409.

[13] Marler R.T., Arora J.S. (2004). Survey of multi-objective optimization methods for engineering. Structural and multidisciplinary optimization, 26(6), 369-395.

[14] Russo M., Herrero S., Altuzarra O., Ceccarelli M. (2018). Kinematic analysis and multi-objective optimization of a 3-UPR parallel mechanism for a robotic leg. Mechanism and Machine Theory, 120, 192-202.

[15] Yang C., Li Q., Chen Q. (2019). Multi-objective optimization of parallel manipulators using a game algorithm. Applied Mathematical Modelling, 74, 217-243.

[16] Hartenberg R., Denavit J. (1964). Kinematic synthesis of linkages. New York: McGraw-Hill.

[17] Laribi M.A., Romdhane L., Zeghloul S. (2007). Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace. Mechanism and machine theory, 42(7), 859-870.

[18] Zhang L., Mei J., Zhao X., Huang T. (2012). Dimensional synthesis of the delta robot using transmission angle constraints dimensional synthesis of the delta robot using transmission angle constraints. Robotica, 30(3), 343-349.

[19] Brinker J., Corves B., Takeda Y. (2019). Kinematic and Dynamic Dimensional Synthesis of Extended Delta Parallel Robots. In Robotics and Mechatronics: Proceedings of the Fifth IFToMM International Symposium on Robotics & Mechatronics (ISRM 2017) (Vol. 72, p. 131). Springer.

[20] Huang T., Li Z., Li M., Chetwynd D.G., Gosselin C.M. (2004). Conceptual design and dimensional synthesis of a novel 2-DOF translational parallel robot for pick-and-place operations. J. Mech. Des., 126(3), 449-455.

[21] Zhao Y. (2013). Dimensional synthesis of a three translational degrees of freedom parallel robot while considering kinematic anisotropic property. Robotics and Computer-Integrated Manufacturing, 29(1), 169-179.

[22] Zhao Y., Cheng G. (2017). Dimensional synthesis of a 3UPS-PRU parallel robot. Robotica, 35(12), 2319-2329.

[23] Zhao X., Zhao T., Xu X., Bian H., Ding S. (2019). Kinematic analysis and dimensional synthesis of a three-degrees-of-freedom hybrid-drive parallel mechanism. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 233(8), 2728-2752.

[24] Zhang Q., Zhao M.Y. (2016). Minimum time path planning of robotic manipulator in drilling/spot welding tasks. Journal of Computational Design and Engineering, 3(2), 132-139.

[25] Wang W., Song H., Yan Z., Sun L., Du, Z. (2018). A universal index and an improved PSO algorithm for optimal pose selection in kinematic calibration of a novel surgical robot. Robotics and computer-integrated manufacturing, 50, 90-101.

[26] Cao X., Zou X., Jia C., Chen M., Zeng Z. (2019). RRT-based path planning for an intelligent litchi-picking manipulator. Computers and electronics in agriculture, 156, 105-118.

[27] Wei K., Ren B. (2018). A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm. Sensors, 18(2), 571.

[28] Garg D.P., Kumar M. (2002). Optimization techniques applied to multiple manipulators for path planning and torque minimization. Engineering applications of artificial intelligence, 15(3-4), 241-252.

[29] Mac T.T., Copot C., Tran D.T., De Keyser R. (2016). Heuristic approaches in robot path planning: A survey. Robotics and Autonomous Systems, 86, 13-28.

[30] Jørgensen T.B., Wolniakowski A., Petersen H.G., Debrabant K., Krüger, N. (2018). Robust optimization with applications to design of context specific robot solutions. Robotics and Computer-Integrated Manufacturing, 53, 162-177.

[31] Ruszczyński A. (1995). On convergence of an augmented Lagrangian decomposition method for sparse convex optimization. Mathematics of Operations Research, 20(3), 634-656.

[32] Zimmermann K. (2003). Disjunctive optimization, max-separable problems and extremal algebras. Theoretical Computer Science, 293(1), 45-54.

[33] Joo J.Y., Ilić M.D. (2013). Multi-layered optimization of demand resources using lagrange dual decomposition. IEEE Transactions on Smart Grid, 4(4), 2081-2088.

[34] Alatorre D., Nasser B., Rabani A., Nagy-Sochacki A., Dong X., Axinte D., Kell, J. (2018, October). Robotic Boreblending: The Future of In-Situ Gas Turbine Repair. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1401-1406). IEEE.

[35] Alatorre D., Nasser B., Rabani A., Nagy-Sochacki A., Dong X., Axinte D., Kell J. (2018). Teleoperated, in situ repair of an aeroengine: Overcoming the internet latency hurdle. IEEE Robotics & Automation Magazine, 26(1), 10-20.

[36] Jones B.A., Walker I.D. (2006). Kinematics for multisection continuum robots. IEEE Transactions on Robotics, 22(1), 43-55.

[37] Webster III R.J., Jones B.A. (2010). Design and kinematic modeling of constant curvature continuum robots: A review. The International Journal of Robotics Research, 29(13), 1661-1683.