Robotica (2018) volume 36, pp. 787–808. © Cambridge University Press 2018
doi:10.1017/S0263574718000048

Optimal planning in robotized cladding processes on generic surfaces

Paolo Magnoni†‡*, Nicola Pedrocchi†, Sebastian Thieme§, Giovanni Legnani‡ and Lorenzo Molinari Tosatti†

†*Institute of Industrial Technologies and Automation, National Research Council, via Corti 12, 20133 Milan, Italy. E-mails: nicola.pedrocchi@itia.cnr.it, lorenzo.molinaritosatti@itia.cnr.it*
‡*Department of Mechanical and Industrial Engineering, University of Brescia, via Branze 39, 25123 Brescia, Italy. E-mail: giovanni.legnani@unibs.it*
§*Fraunhofer Institut fur Werkstoff und Strahltechnik, Winterbergstr 28, 01277 Dresden, Germany. E-mail: sebastian.thieme@iws.fraunhofer.de*

(Accepted January 2, 2018. First published online: January 21, 2018)

SUMMARY
Cladding through laser metal deposition is a promising application of additive manufacturing. On the one hand, industrial robots are increasingly used in cladding because they provide wide wrist reorientation, which enables manufacturing of complex geometries. On the other hand, limitations in robot dynamics may prevent cladding of sharp edges and large objects. To overcome these issues, this paper aims at exploiting the residual degrees of freedom granted by the cladding process for the optimization of the deposition orientation. The proposed method optimizes the robot head orientation along a predefined path while coping with kino-dynamic constraints as well as process constraints. Experimental tests and results are reported and used to validate the approach.

KEYWORDS: Robotic additive manufacturing; Laser cladding; Orientation planning.

1. Introduction

Additive manufacturing (AM) of metallic parts is expected to be one of the most rapidly growing manufacturing fields in the coming years.[1] Among the various AM processes, direct energy deposition (DED) techniques are the best candidates for the production of large metal parts, repair of damaged components, and cladding of free-form surfaces. In these processes, a deposition material (usually metallic powder or wire) is melted on to a base material either by the action of a high-powered laser beam, called laser metal deposition (LMD), or by an electron beam, called electron beam melting. These processes generate a single deposited track of solidified material and allow the creation of complex cladding surfaces or 3D parts by depositing layer after layer.[2,3] Many successful examples of LMD applications can be found in field-specific literature.[4–10] In using the LMD technique, efforts generally focus on ensuring that the translation speed (or feed rate) is kept constant, with little attention paid to the orientation of the deposition head with respect to the underlying surface. Nevertheless, the angle between the deposition head and the perpendicular to the surface is important to improve the laser energy absorption,[11] and to avoid the reflection problems.[12–14] Indeed, Lalas *et al.*[11] demonstrate that the optimal laser energy absorption is achieved when the deposition axis is perpendicular to the surface. Liu and Li[12] show that the reflection problems are minimal when the deposition axis is perpendicular to the surface; despite this results are not generally valid since the back-reflection phenomenon depends on the design of the head.[13] Remarkably, the works[11–14] highlight that the process is feasible, even if not optimal, for a large range of relative orientations between the deposition axis and the perpendicular to

* Corresponding author. E-mail: paolo.magnoni@itia.cnr.it

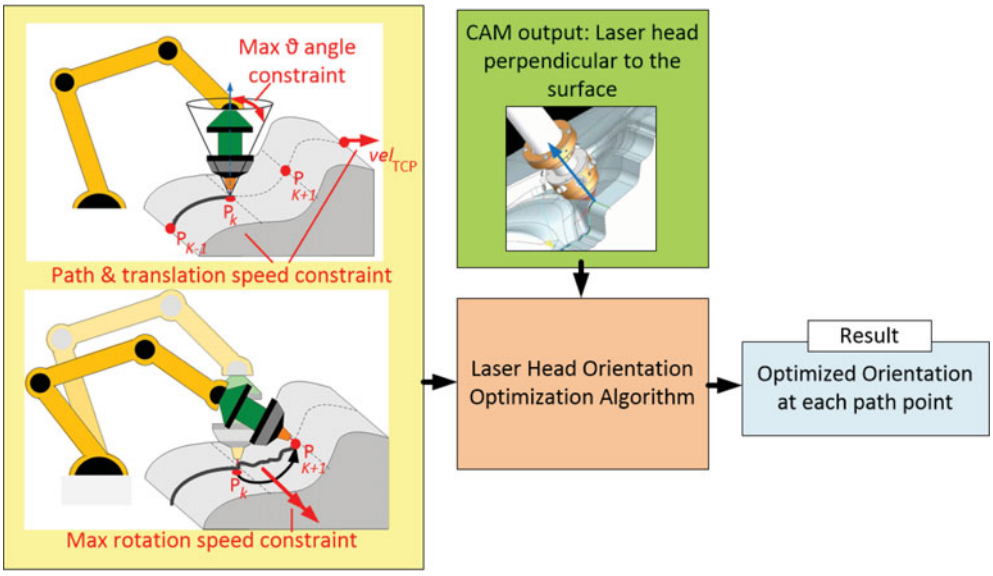


Fig. 1. Scheme of kino-dynamic constraints and desired targets for the residual degrees-of-freedom optimization in defining a laser head orientation.

the surface. As a matter of example, Liu[12] shows an application where the angle between the deposition axis and the perpendicular to the surface is about 45 deg. Therefore, for a wide class of cladding processes, it is possible to define the "feasibility cone" as the cone containing all the orientations of the deposition axis for which the process is feasible (even if not optimal).[(1)] The optimization of the angle between the deposition axis and the perpendicular to the surface is poorly addressed also in computer-aided manufacturing software (CAM). In fact, available CAM software simply generates paths where the deposition axis is always orthogonal to the surface and provides simple filtering methods to smooth the profile. Recently, some CAM software applications released specific modules for AM with LMD technique.[15] However, these modules allow the setting and the control of process parameters without offering algorithms for the orientation management of the deposition head. The path smoothness is essential to ensure that the trajectory is actually executable, by either CNC workstations or by robots. In fact, in the cladding of complex surfaces, sharp edges, and sudden modifications of the surface curvature require fast reorientation and high-dynamics performance, often not feasible due to the robot kinematics and dynamics constraints (see Fig. 1). Industrial practice generally solves process planning issues through hand-made programming and trial-error optimization.[16] This practice is burdensome to be applied to industrial anthropomorphic robots since it is time consuming: with robots, orientation planning is fundamental, and small errors in orientation planning may make the entire operation infeasible. However, despite the complexity in using industrial robots without the support of specific CAM software tools, they are considered the most promising platforms for AM cladding,[17] since industrial robots have better flexibility, reachability, and workspace dimension than standard 3D-printing machines and CNC workstations.

1.1. Contribution

Therefore, this paper focuses on the cladding processes performed with six degrees of freedom (DOFs) manipulators, and proposes a novel method to compute the "optimal" deposition head orientations for each node of the path exploiting the residual DOF given by the "feasibility cone". The optimality problem is to find the path that ensures that all technological constraints (tool center velocity, kinematic and dynamics constraint, and feasibility cone constraints) are met, while at the same time optimizing an objective function designed to maintain the deposition head in a pose orthogonal to the surface and simultaneously smoothing the reorientation (see Fig. 1). In this paper, an exhaustive mathematical

[(1)]The most common laser deposition heads have a "feasibility cone" that is centered around the perpendicular to the component surface.

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

formulation of the orientation-smoothing problem is reported. A "cost function" is proposed, and the optimal plan is defined as the plan that minimizes such function. The paper is structured as follows: Section 2 analyses state-of-the-art methodologies in robot motion planning; Section 3 is dedicated to the mathematical formulation of the problem; Section 4 describes the optimization method used to solve the planning problem; and finally, Section 5, an experimental section, describes the results achieved both in simulation and in a set of experiments.

2. Related Works

The laser deposition process is usually carried out using 3- or 3+2-axis kinematics, but the increasing demand for parts with complex geometry requires the process to be performed in five continuous kinematic axes or more.[13,18] Therefore, 5-axis and 6-axis laser cladding technologies represent a research challenge, especially given the lack of specific process-oriented CAM software to generate optimal orientation planning.[19] Most of the motion planning literature for AM and cladding processes focuses on the travel-path topology of the laser head's tool center point (TCP)[(2)].[20] Despite laser cladding may be performed using different strategies, Calleja in ref. [13] has observed that research into deposition strategies primarily focuses on the 3-axis laser cladding process. The importance of extending the research to applications that require five or more continuous axes is therefore clear. Calleja experimentally compares several TCP deposition paths with different 5-axis interpolations using an optimal set of process parameters to achieve a good quality result (i.e., laser power, feed rate, and powder flow rate), but does not provide an optimization algorithm. One of the few works exploiting reorientation of the laser head to improve deposition quality is ref. [12]. In that case, however, the inclination is not pre-optimized, but is changed in real time in response to online monitoring of the deposited shape. For industrial purposes, the well-known and widely adopted CAMs for laser cladding are the Siemens NX[21] and the DCAM.[22] These CAM programs calculate the path with the head axis perpendicular to the object's surface as the best cladding path, and they allow the operator to manually change the orientation where needed in case of non-smooth rotations. The most common practical solution to profile smoothing consists in reducing the translation speed and consequently the synchronous orientation speed. This, however, gives non-optimal results; the speed constraints may be satisfied, but the orientation can be far from the proper perpendicularity to the surface. Hence, a complete solution for the problem of optimal orientation planning in cladding processes using industrial robots is poorly considered in Literature, commercial software, and industrial practice. The only field in which a similar problem can be found is CNC-machining of free-form surfaces. In generic CAMs for subtractive processes, advanced algorithms are available.[23] These types of software implement heuristic methods to define the tool orientation in order to avoid collision with gauges, preserve the position within joint limits, and get smooth speed profiles. These smooth profiles, which are required to avoid a reduction in the surface quality, are often achieved using three common techniques: (i) they minimize the distance between successive tool orientations (with forward schemes and forward+backward schemes); (ii) they use mixed manual and automatic algorithms to lower the computation time of optimization; and (iii) they use the concept of the discrete visibility map. These methods are intended to optimized the orientation in 5-axis CNC machining, and thus modifying tilt and inclination angles.[23] However, in the cladding task with robots, the optimization will involve all the three angles describing a complete orientation in free space and novel-specific algorithms are required. Concluding, a recent work by Sellmann[24] is worth of note. It investigates the problem and proposes a method to optimize the orientation of cutting tools on 5-axis machines. It introduces three main assumptions: (i) the 5-axis machines are fast enough to achieve high dynamics, and the quality improvement is related to the jerk smoothing; (ii) the tolerance on quasi-redundant axes is limited to ± 5 deg; and (iii) the rotation coupling[(3)] is neglected since the machine orientation is limited. The orientation smoothing can be therefore modeled as a quadratic problem linearized around the surface normal axis. However, these assumptions are not realistic for robotic cladding: the robot limitations are related to maximum angular velocity and acceleration rather than jerk, and the angular coupling is a fundamental phenomenon when robot orientations have to be controlled. These aspects

[(2)]which corresponds with the laser focal point.
[(3)]The first derivative of the Cardan/Euler representation is coupled with respect to the angular velocity by a matrix function of the actual orientation.

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

are critical on industrial robots, where high-rotation speed leads to high-dynamic effort with resulting motion inaccuracies and vibrations, which cause infeasibility. Furthermore, as shown in refs. [12,18], and exploiting new laser-head designs,[25] cladding displays a larger orientation tolerance around the normal axis of the surface than do the cutting processes studied in ref. [24]. The identification problem of the best motion plan that satisfies task, kinematic and dynamic constraints[26] is PSPACE-hard,[27] and far to be solved. Sampling-based planners, such as probabilistic roadmaps (PRM)[28] and rapidly-exploring random trees (RRT)[29] are the most used, since they solve motion planning problems in high-dimensional state spaces in a reasonable computational time. RRT*,[30] an extension of RRT, goes beyond RRT and PRM always guaranteeing convergence in finite time. However, it is applicable only to systems with simple dynamics with linear constraints. Webb *et al.*[31] enable RRT* to solve systems with linear differential constraints. The method works well in motion planning at the robot joint levels, while it is not easy to be applicable in Cartesian space since the dynamic constraints are extremely non-linear. The implementation of ref. [31] for the planning of the laser cladding path is difficult. Indeed, the method work at joint level, and accepts constraints in the Cartesian space, but does not manage a cost function where some DOF are totally constrained (i.e., Cartesian position) and some DOF must be optimized (i.e., orientations), as requested by cladding planning problem. In addition, an extension of ref. [31] would be needed since the research domain in the joint space results in a non-convex partially-constrained domain.

3. Problem Formulation

A generic cladding trajectory generated by commercial off-the-shelf CAM software can be represented as an ordered set of N nodes, where the kth node can be expressed by the Cartesian point $\mathbf{P}_k$ lying on the object surface and the Cardan ZYX angles (A_k, B_k, C_k) of a frame that originates in $\mathbf{P}_k$, with the x-axis directed as the tangent to the trajectory and the z-axis normal to the surface. Furthermore, since the linear deposition velocity $\mathbf{v}$ is a hard constraint for the process, each kth node is also characterized by an execution time t_k.

Therefore, denoting the workpiece frame with $\{w\}$ and the CAM-calculated frame for the kth node with $\{c_k\}$, it is possible to introduce the following unit vectors:

$$ {}^{w}\mathbf{t}^{c_k} := \mathbf{t}(A_k, B_k, C_k), \quad {}^{w}\mathbf{n}^{c_k} := \mathbf{n}(A_k, B_k, C_k), \quad {}^{w}\mathbf{b}^{c_k} := \mathbf{b}(A_k, B_k, C_k), $$

where ${}^{w}\mathbf{n}^{c_k}$, ${}^{w}\mathbf{t}^{c_k}$, and ${}^{w}\mathbf{b}^{c_k}$ are, respectively, the unit vector normal to the surface, the tangent to the trajectory, and the binormal to the trajectory, expressed in the object reference frame $\{w\}$.

Then, it is easy to define the rotation matrix ${}^{w}\mathbf{R}_{c_k}$ as

$$ {}^{w}\mathbf{R}_{c_k}(A_k, B_k, C_k) := \left[{}^{w}\mathbf{t}^{c_k} \quad {}^{w}\mathbf{b}^{c_k} \quad {}^{w}\mathbf{n}^{c_k} \right]. $$

Consider now the robot end-effector when the tool is in $\mathbf{P}_k$, and denote $(\alpha_k, \beta_k, \gamma_k)$ and ${}^{w}\mathbf{R}_{ee_k} := {}^{w}\mathbf{R}_{ee_k}(\alpha_k, \beta_k, \gamma_k)$ as the Cardan ZYX angles and the rotation matrix of a frame $\{ee\}$ fixed on the robot end-effector (see Fig. 2).

Like ${}^{w}\mathbf{R}_{c_k}$, it is possible to expand ${}^{w}\mathbf{R}_{ee_k}$ as

$$ {}^{w}\mathbf{R}_{ee_k}(\alpha_k, \beta_k, \gamma_k) := \left[{}^{w}\mathbf{t}^{ee_k} \quad {}^{w}\mathbf{b}^{ee_k} \quad {}^{w}\mathbf{n}^{ee_k} \right], $$

where ${}^{w}\mathbf{t}^{ee_k}$, ${}^{w}\mathbf{b}^{ee_k}$, and ${}^{w}\mathbf{n}^{ee_k}$ are the three axes of $\{ee\}$ such that ${}^{w}\mathbf{n}^{ee_k}$ is parallel to the laser head deposition direction.

As discussed in Section 1, a DED-based cladding process allows higher DOF in defining a feasible head orientation and therefore in defining the robot pose. In DED-based cladding processes, ${}^{w}\mathbf{R}_{ee_k}$ can differ from the CAM frame ${}^{w}\mathbf{R}_{c_k}$ calculated in $\mathbf{P}_k$.

Consequently, the path-planning problem consists in the definition of an optimal set of Cardan angles $\left(\alpha_k^o, \beta_k^o, \gamma_k^o\right)$ for the end-effector pose, which satisfy kinematic, dynamic, and technological constraints and optimize a user-defined objective function.

It is worth noting that the formulation of the problem relies on the assumption that ${}^{w}\mathbf{R}_{ee_k}$ is correctly tracked by the robot during the execution (e.g., the maximum trajectory-tracking error during the execution is bounded and small). This assumption will be further examined in the present work.

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

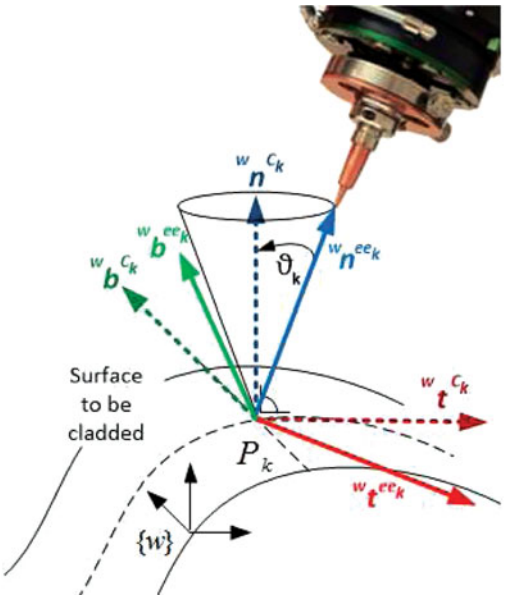


Fig. 2. Definition of frames and angles on a kth node of the cladding path.

3.1. Technological constraints

As discussed in Section 1, the material deposition axis must lie within a feasibility cone whose revolution axis is normal to the surface (see Fig. 2), and the translation velocity must be considered a hard constraint.

Denoting ϑ_k as the relative angle between the normal to the surface ${}^{w}\mathbf{n}^{c_k}$ and the laser head deposition axis ${}^{w}\mathbf{n}^{ee_k}$,

$$\vartheta_k := \operatorname{acos}\left({}^{w}\mathbf{n}^{c_k}\left(A_k, B_k, C_k\right) \cdot {}^{w}\mathbf{n}^{ee_k}\left(\alpha_k, \beta_k, \gamma_k\right)\right),$$

and denoting $\vartheta_{k_{\max}}$ as the maximum relative angle allowed by the process properties, it is possible to write that

$$\vartheta_k\left(\alpha_k, \beta_k, \gamma_k\right) < \vartheta_{k_{\max}}, \ \forall k = 1, \ldots, n. \tag{1}$$

3.2. Kino-dynamic constraints

In robotics and motion planning, kino-dynamic planning problems are a class of problems for which velocity, acceleration, and force/torque bounds must be satisfied at the same time that kinematic constraints, such as obstacle avoidance, are satisfied.[26]

Remark. Without loss of generality, the kino-dynamic constraints will be identified assuming that the orientation $(\alpha_k, \beta_k, \gamma_k)$ is close to the subsequent $(\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1})$, and a small time difference $t_{k+1} - t_k$ is between the two nodes. This assumption is feasible considering that the set of points $\mathbf{P}_k$ generated from CAM is dense.

3.2.1. Kinematic reachability. Since it is possible to modify the robot orientation for each node of the trajectory $\mathbf{P}_k$, it is necessary to calculate for each node the maximum range for the orientation in order to guarantee reachability and avoid collisions with external objects. The orientation configuration space mapped at the joint level may not result convex nor connected. Fortunately, the configuration space analysis is a well-established problem, and many simulation software tools are available for industrial robots, such as CimStation,[32] Robcad,[33] RobotStudio,[34] 3DAutomate,[35] Delmia,[36] and Industrial Path Planner.[37] Each of these software tools handles one or more robotics tasks. Some handle collision detection and/or trajectory optimization for the robots themselves; others allow users to program the robots off-line, design robot work cells, and perform other user-defined tasks through the development of user macros. In addition, many of these tools allow also a partially automated repositioning of the work object to identify the best position for robot dexterity. To ensure that the solution we are

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

proposing is as generalizable as possible, we have developed a C++ executable based on the Flexible Collision Library (FCL[38]) that describes the work scene using the Unified Robot Description Format standard.[39] The algorithm assumes that the orientation configuration space $\{ee_k\}$ is the local convex hull of all the reachable configurations starting from the nominal $\{c_k\}$. The local configuration span around the starting position has been analyzed by exploiting the Halton sequences,[40] which allow uniform mapping of high-dimensional spaces with a limited number of points. Therefore, once the work scene is given, the reachability constraints can be simply expressed as

$$\alpha_k^{\min} \leq \alpha_k \leq \alpha_k^{\max}, \quad \beta_k^{\min} \leq \beta_k \leq \beta_k^{\max}, \quad \gamma_k^{\min} \leq \gamma_k \leq \gamma_k^{\max}, \tag{2}$$

where $\left(\alpha_k^{\max}, \beta_k^{\max}, \gamma_k^{\max}\right)$ and $\left(\alpha_k^{\min}, \beta_k^{\min}, \gamma_k^{\min}\right)$ are the boundaries of the convex hull of the orientation space in each $\mathbf{P}_k$.

3.2.2. Speed constraints. In order to express the speed constraints, consider the well-known relation between the first derivative of the Cardan angles and the angular velocity ${}^k\boldsymbol{\omega}$:

$${}^k\boldsymbol{\omega} = \begin{bmatrix} 1 & 0 & -\sin(\beta_k) \\ 0 & \cos(\alpha_k) & \sin(\alpha_k)\cos(\beta_k) \\ 0 & -\sin(\alpha_k) & \cos(\alpha_k)\cos(\beta_k) \end{bmatrix} \begin{bmatrix} \dot{\alpha}_k \\ \dot{\beta}_k \\ \dot{\gamma}_k \end{bmatrix} = \mathbf{T}_{ee_k}(\alpha_k, \beta_k, \gamma_k) \begin{bmatrix} \dot{\alpha}_k \\ \dot{\beta}_k \\ \dot{\gamma}_k \end{bmatrix}.$$

Then, consider the well-known relation between the tool angular velocity ${}^w\boldsymbol{\omega}$ expressed in the world frame $\{W\}$, and the joint velocities $\dot{\mathbf{q}}$:

$${}^w\boldsymbol{\omega} = \mathbf{J}_{ornt}(\mathbf{q}_k)\,\dot{\mathbf{q}}, \quad \text{with} \quad \mathbf{q}_k = inverse\_kinematic(\mathbf{P}_k, \alpha_k, \beta_k, \gamma_k), \tag{3}$$

where $\mathbf{J}_{ornt}$ is the orientation Jacobian of the robot. Therefore, the relation between the relative orientation and the joint positions and velocities results in

$$\begin{bmatrix} \dot{\alpha}_k \\ \dot{\beta}_k \\ \dot{\gamma}_k \end{bmatrix} = \mathbf{T}_{ee_k}^{-1}\,\mathbf{J}_{ornt}\,\dot{\mathbf{q}}.$$

This relation is coupled in $(\alpha_k, \beta_k, \gamma_k)$, but is linear with respect to the joint velocities $\dot{\mathbf{q}}$. Therefore, the speed constraints are analytically defined by an upper-bound surface,

$$\begin{bmatrix} \dot{\alpha}_k^b \\ \dot{\beta}_k^b \\ \dot{\gamma}_{k^b} \end{bmatrix} := \begin{bmatrix} \dot{\alpha}_k^b(\alpha_k, \beta_k, \gamma_k) \\ \dot{\beta}_k^b(\alpha_k, \beta_k, \gamma_k) \\ \dot{\gamma}_k^b(\alpha_k, \beta_k, \gamma_k) \end{bmatrix} := \mathbf{T}_{ee_k}^{-1}\,\mathbf{J}_{ornt}\,\dot{\mathbf{q}}_{\max}, \tag{4}$$

and the first derivative of the Cardan angles must satisfy the relation

$$\begin{bmatrix} \dot{\alpha}_k \\ \dot{\beta}_k \\ \dot{\gamma}_k \end{bmatrix} \leq \begin{bmatrix} \dot{\alpha}_k^b \\ \dot{\beta}_k^b \\ \dot{\gamma}_{k^b} \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} \alpha_{k+1} \\ \beta_{k+1} \\ \gamma_{k+1} \end{bmatrix} \leq \begin{bmatrix} \alpha_k \\ \beta_k \\ \gamma_k \end{bmatrix} + \begin{bmatrix} \dot{\alpha}_k^b(\alpha_k, \beta_k, \gamma_k) \\ \dot{\beta}_k^b(\alpha_k, \beta_k, \gamma_k) \\ \dot{\gamma}_k^b(\alpha_k, \beta_k, \gamma_k) \end{bmatrix} (t_{k+1} - t_k). \tag{5}$$

3.2.3. Torque constraints definition. The definition of torque-level constraints is slightly more complex than that for speed-level constraints. In fact, considering the first derivative of Eq. (3) and the well-known equations of robot dynamics,[41] it is possible to write the following system of equations:

$$\begin{aligned} \begin{bmatrix} \ddot{\alpha}_k \\ \ddot{\beta}_k \\ \ddot{\gamma}_k \end{bmatrix} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{T}_{ee_k}^{-1}\right){}^k\boldsymbol{\omega} + \mathbf{T}_{ee_k}^{-1}\,{}^k\dot{\boldsymbol{\omega}}, \\ {}^w\dot{\boldsymbol{\omega}} &= \dot{\mathbf{J}}_{ornt}\dot{\mathbf{q}} + \mathbf{J}_{ornt}\,\mathbf{B}^{-1}\left(\boldsymbol{\tau} - \mathbf{C}\,\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q})\right), \end{aligned}$$

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

where $\boldsymbol{\tau}$ is the torque at joint level, $\mathbf{B}$ is the mass matrix, $\mathbf{C}$ is the Coriolis matrix, and $\mathbf{G}(\mathbf{q})$ is the gravity vector. When the equations are combined, the relation between the second derivative of the Cardan angles and the robot kinetic state is

$$\begin{bmatrix} \ddot{\alpha}_k \\ \ddot{\beta}_k \\ \ddot{\gamma}_k \end{bmatrix} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{T}_{ee_k}^{-1}\right)\mathbf{J}_{ornt}\,\dot{\mathbf{q}} + \mathbf{T}_{ee_k}^{-1}\left(\dot{\mathbf{J}}_{ornt}\,\dot{\mathbf{q}} + \mathbf{J}_{ornt}\,\mathbf{B}^{-1}\left(\boldsymbol{\tau} - \mathbf{C}\,\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q})\right)\right). \tag{6}$$

The complexity comes from the fact that the upper bound for $\left(\ddot{\alpha}_k,\ \ddot{\beta}_k,\ \ddot{\gamma}_k\right)$ is not linearly dependent on $\dot{\mathbf{q}}_{\max}$ and $\boldsymbol{\tau}_{\max}$; it can be found in a combination of the terms. Therefore, it is not possible to define analytically a hyper-surface as the upper boundary for the second derivative of the Cardan angles. Indeed, the right-hand term may be a non-convex, unconnected hull. The maximum acceleration can often be computed only through proper linearization of the system around the working conditions (see Section 3.3.2).

3.3. Problem reduction

The kinematic and dynamic constraints on the Cardan angles in each robot trajectory node $\mathbf{P}_k$ result in non-linear coupled functions that may increase the identification complexity of the optimal pose set for the deposition head. Furthermore, common optimization algorithms require that the state configuration be decoupled from the state constraints. Therefore, a complete solution to the optimization problem would necessitate developing an *ad-hoc* method. However, some simplifications can be done without loss of generality for a wide class of tasks. Indeed, when the robot dexterity is rather homogeneous in all movement directions, we can assume that, given a configuration $\overline{\mathbf{q}}$, the maximum velocity and acceleration can be considered relatively constant in a large range around the given configuration. This assumption loses its validity when the robot is close to the workspace boundaries, or to its singular configuration. Under bounded conditions, the assumption could also be reasonable for path planning in the robotized cladding of parts whose dimension is smaller than the robot workspace. The complete solution to the general problem will be investigated in a future work.

3.3.1. Speed constraints simplification. Consider (5) and the reachability limits calculated in the node $\mathbf{P}_k$ as defined in (2).

Recalling that $\dot{\alpha}_{k,\,\max}$, $\dot{\beta}_{k,\,\max}$, and $\dot{\gamma}_{k^{\max}}$ are functions of α_k, β_k, and γ_k, respectively, it is possible to assume that the hull is connected and convex around the nominal orientation. This convex hull can be simply approximated by a parallelepiped. Therefore, it is possible to compute the three bound conditions one for each Cardan angle:

$$\begin{bmatrix} \alpha_k^1 \\ \beta_k^1 \\ \gamma_k^1 \end{bmatrix} = \underset{\alpha_k,\beta_k,\gamma_k}{\arg\max}\left(\dot{\alpha}_k^b\right), \quad \begin{bmatrix} \alpha_k^2 \\ \beta_k^2 \\ \gamma_k^2 \end{bmatrix} = \underset{\alpha_k,\beta_k,\gamma_k}{\arg\max}\left(\dot{\beta}_k^b\right), \quad \begin{bmatrix} \alpha_k^3 \\ \beta_k^3 \\ \gamma_k^3 \end{bmatrix} = \underset{\alpha_k,\beta_k,\gamma_k}{\arg\max}\left(\dot{\gamma}_k^b\right). \tag{7}$$

The identification of such global maxima in the configuration space is generally guaranteed, and it is simple to arrive at the global maxima since the space being bounded has only three dimensions. Given the bound conditions, it is possible therefore to compute the corresponding velocities as

$$\begin{bmatrix} \dot{\alpha}_k^1 \\ \dot{\beta}_k^1 \\ \dot{\gamma}_k^1 \end{bmatrix} = \begin{bmatrix} \dot{\alpha}_k^b\left(\alpha_k^1, \beta_k^1, \gamma_k^1, \right) \\ \dot{\beta}_k^b\left(\alpha_k^1, \beta_k^1, \gamma_k^1, \right) \\ \dot{\beta}_k^b\left(\alpha_k^1, \beta_k^1, \gamma_k^1, \right) \end{bmatrix}, \quad \begin{bmatrix} \dot{\alpha}_k^2 \\ \dot{\beta}_k^2 \\ \dot{\gamma}_k^2 \end{bmatrix} = \begin{bmatrix} \dot{\alpha}_k^b\left(\alpha_k^2, \beta_k^2, \gamma_k^2, \right) \\ \dot{\beta}_k^b\left(\alpha_k^2, \beta_k^2, \gamma_k^2, \right) \\ \dot{\beta}_k^b\left(\alpha_k^2, \beta_k^2, \gamma_k^2, \right) \end{bmatrix},$$

$$\begin{bmatrix} \dot{\alpha}_k^3 \\ \dot{\beta}_k^3 \\ \dot{\gamma}_k^3 \end{bmatrix} = \begin{bmatrix} \dot{\alpha}_k^b\left(\alpha_k^3, \beta_k^3, \gamma_k^3, \right) \\ \dot{\beta}_k^b\left(\alpha_k^3, \beta_k^3, \gamma_k^3, \right) \\ \dot{\beta}_k^b\left(\alpha_k^3, \beta_k^3, \gamma_k^3, \right) \end{bmatrix}.$$

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

Finally, the maximum first derivative of the Cardan angles $\dot{\alpha}_{k,\max}$, $\dot{\beta}_{k,\max}$, and $\dot{\gamma}_{k,\max}$ can be computed as

$$\begin{bmatrix} \dot{\alpha}_{k,\max} \\ \dot{\beta}_{k,\max} \\ \dot{\gamma}_{k,\max} \end{bmatrix} = \begin{bmatrix} \min\left(\dot{\alpha}_k^1, \dot{\alpha}_k^2, \dot{\alpha}_k^3\right) \\ \min\left(\dot{\beta}_k^1, \dot{\beta}_k^2, \dot{\beta}_k^3\right) \\ \min\left(\dot{\gamma}_k^1, \dot{\gamma}_k^2, \dot{\gamma}_k^3\right) \end{bmatrix}. \tag{8}$$

These constraints, which differ with respect to the different nodes k considered, can be easily computed off-line, before optimization.

Two remarks are as follows:

1. The approximation of the boundaries through a parallelepiped is conservative, and it could over-limit the optimization procedure (see Section 4). In this case, relaxing these constraints may be the simplest heuristic to guarantee a solution.
2. It is important to analyze the constraints of the speed configuration space in order to check that the assumption of the robot's quasi-homogeneous dexterity along different movement directions is a valid assumption.

3.3.2. Dynamics constraints simplification. Consider (6) and the reachability limits calculated in the node $\mathbf{P}_k$ as defined in (2). As above mentioned, the dynamics constraints, are not described by hyper-surfaces. Therefore, the computation of the three bound conditions, one for each Cardan angle, is complex: the equations to be solved are defined in a non-convex domain, and the dimension of the exploration space is significantly high: $3 \times 6 \times 6$

$$\begin{bmatrix} \alpha_k^1 \\ \beta_k^1 \\ \gamma_k^1 \\ \dot{\mathbf{q}}^1 \\ \boldsymbol{\tau}^1 \end{bmatrix} = \underset{\alpha_k, \beta_k, \gamma_k, \dot{\mathbf{q}}, \boldsymbol{\tau}}{\arg\max} \left(\ddot{\alpha}_k\right), \quad \begin{bmatrix} \alpha_k^2 \\ \beta_k^2 \\ \gamma_k^2 \\ \dot{\mathbf{q}}^2 \\ \boldsymbol{\tau}^2 \end{bmatrix} = \underset{\alpha_k, \beta_k, \gamma_k, \dot{\mathbf{q}}, \boldsymbol{\tau}}{\arg\max} \left(\ddot{\beta}_k\right), \quad \begin{bmatrix} \alpha_k^3 \\ \beta_k^3 \\ \gamma_k^3 \\ \dot{\mathbf{q}}^3 \\ \boldsymbol{\tau}^3 \end{bmatrix} = \underset{\alpha_k, \beta_k, \gamma_k, \dot{\mathbf{q}}, \boldsymbol{\tau}}{\arg\max} \left(\ddot{\gamma}_k\right),$$

A further constraint simplification is needed. Under the hypothesis that the velocity must remain bounded and small, we can ignore the term that linearly depends on the joint velocity in (6), resulting in:

$$\begin{bmatrix} \ddot{\alpha}_k \\ \ddot{\beta}_k \\ \ddot{\gamma}_k \end{bmatrix} \approx \mathbf{J}_{ornt}\, \mathbf{B}^{-1} \left(\boldsymbol{\tau} - \mathbf{G}(\mathbf{q})\right).$$

This assumption allows a dramatic simplification of the problem, making the representation of the dynamic constraints similar to (4):

$$\begin{bmatrix} \ddot{\alpha}_k^b \\ \ddot{\beta}_k^b \\ \ddot{\gamma}_{k^b} \end{bmatrix} := \begin{bmatrix} \ddot{\alpha}_k^b\left(\alpha_k, \beta_k, \gamma_k\right) \\ \ddot{\beta}_k^b\left(\alpha_k, \beta_k, \gamma_k\right) \\ \ddot{\gamma}_k^b\left(\alpha_k, \beta_k, \gamma_k\right) \end{bmatrix} := \mathbf{J}_{ornt}\, \mathbf{B}^{-1} \boldsymbol{\tau}^{\max}, \tag{9}$$

where $\ddot{\alpha}_k^b$, $\ddot{\beta}_k^b$, and $\ddot{\gamma}_k^b$ are the upper-bound surfaces. Therefore, the second derivative of the Cardan angles must satisfy the relation

$$\begin{bmatrix} \ddot{\alpha}_k \\ \ddot{\beta}_k \\ \ddot{\gamma}_k \end{bmatrix} \leq \begin{bmatrix} \ddot{\alpha}_k^b \\ \ddot{\beta}_k^b \\ \ddot{\gamma}_{k^b} \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} \dot{\alpha}_{k+1} \\ \dot{\beta}_{k+1} \\ \dot{\gamma}_{k+1} \end{bmatrix} \leq \begin{bmatrix} \dot{\alpha}_k \\ \dot{\beta}_k \\ \dot{\gamma}_k \end{bmatrix} + \begin{bmatrix} \ddot{\alpha}_k^b\left(\alpha_k, \beta_k, \gamma_k\right) \\ \ddot{\beta}_k^b\left(\alpha_k, \beta_k, \gamma_k\right) \\ \ddot{\gamma}_k^b\left(\alpha_k, \beta_k, \gamma_k\right) \end{bmatrix} \left(t_{k+1} - t_k\right).$$

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

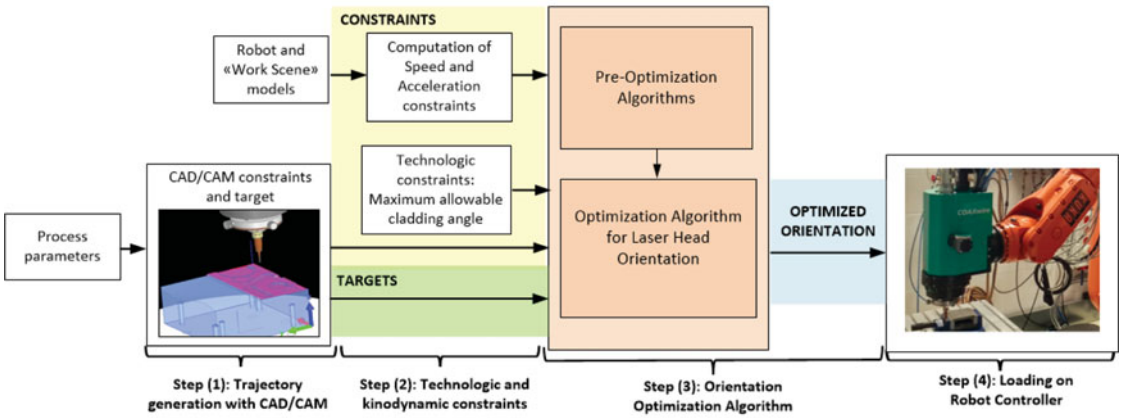


Fig. 3. Scheme of proposed planning algorithm.

Similarly to what obtained in (5), the acceleration constraints results in

$$\begin{bmatrix}\alpha_{k+1}\\ \beta_{k+1}\\ \gamma_{k+1}\end{bmatrix} \leq \left(1+\frac{(t_{k+1}-t_k)}{(t_k-t_{k-1})}\right)\begin{bmatrix}\alpha_k\\ \beta_k\\ \gamma_k\end{bmatrix} - \frac{(t_{k+1}-t_k)}{(t_k-t_{k-1})}\begin{bmatrix}\alpha_{k-1}\\ \beta_{k-1}\\ \gamma_{k-1}\end{bmatrix} + \begin{bmatrix}\ddot{\alpha}_k^b\\ \ddot{\beta}_k^b\\ \ddot{\gamma}_k^b\end{bmatrix}(t_{k+1}-t_k)^2 . \qquad (10)$$

Finally, under the same assumptions of the speed constraints simplification, the hull results connected and convex around the nominal orientation, and can be approximated by a parallelepiped. Therefore, following the procedure in Section 3.3.1 to pass from (7) to (8) can be implemented to calculate the global maximum $(\ddot{\alpha}_{k,\max}, \ddot{\beta}_{k,\max}, \ddot{\gamma}_{k,\max})$.

Remark. The assumption that velocity can be neglected in (6) is conservative. However, it is worth underlining that if the kinematic state of the robot does not satisfy the conditions of this assumption, the tracking error of the robot will probably not be bounded and small. Indeed, often the dynamics effects are not properly managed by industrial robot controllers (i.e., fast trajectory tracing is challenging for standard industrial robots).

4. Optimal Path-Planning Calculus

The method presented here, which was developed to solve the planning problem of optimizing a cladding trajectory, consists of four steps:

1. A trajectory is generated with existing off-the-shelf CAM software (SKM-DCAM). As already noted in the Introduction, in order to obtain good energy absorption during the LMD process, the deposition head should be maintained perpendicular to the underlying surface (the optimum pose for the LMD technology). This functionality is already provided by all the considered CAMs. Therefore, given a surface to be cladded (or to be used as a base from which the object will be built up), off-the-shelf software can be used to compute a tool path in which the laser head axis is always perpendicular to the underlying surface. In addition, the CAM also solves for the definition of the normal axis in correspondence of vertexes and edges.
2. The technological and kino-dynamic constraints are calculated according to Eqs. (1), (8), and (10) for each point of the path generated from the CAM.
3. A solver is deployed to identify the path that meets all constraints and optimizes the objective function defined in Section 4.1.
4. Finally, the optimized output is loaded on the robot controller.

4.1. Cost function definition

From the technological point of view, the optimal solution will maintain the robot tool orientation $(\alpha_k, \beta_k, \gamma_k)$ in a position as close as possible to the normal axis of the surface. Therefore, the optimum

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

condition would be

$$\alpha_k = A_k, \quad \beta_k = B_k, \quad \gamma_k = C_k, \quad \forall k.$$

As a consequence, denoting $\mathbf{x}$ and $\mathbf{x}^*$ as

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \alpha_1 & \dots & \alpha_n & \beta_1 & \dots & \beta_n & \gamma_1 & \dots & \gamma_n \end{bmatrix}^{\mathrm{T}}, \\ \mathbf{x}^* &= \begin{bmatrix} A_1 & \dots & A_n & B_1 & \dots & B_n & C_1 & \dots & C_n \end{bmatrix}^{\mathrm{T}}, \end{aligned}$$

a feasible cost function to be minimized can be the sum of square errors:

$$f(\mathbf{x}) = \left(\mathbf{x} - \mathbf{x}^*\right)^{\mathrm{T}} \left(\mathbf{x} - \mathbf{x}^*\right). \tag{11}$$

4.2. Constraint representation

The speed constraints (linearized as shown in Section 3.3) can be written as

$$\mathbf{M}_{vel} \cdot \mathbf{x} < \mathbf{b}_{\mathbf{vel}}, \tag{12}$$

where $\mathbf{M}_{vel}$ is a $3(n-1) \times 3n$ matrix defined as

$$\mathbf{M}_{vel} = \begin{bmatrix} \mathbf{M}_{vel,1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{vel,2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{vel,3} \end{bmatrix} \quad \text{with} \quad \overbrace{\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}}^{\boldsymbol{M_{vel,i}} \;\; (n-1)\times n, \;\; i=1,\dots,3}$$

and where $\mathbf{b}_{vel}$ is a $3(n-1) \times 1$ vector defined as

$$\begin{aligned} \mathbf{b}_{vel_\alpha} &= \left[\dot{\alpha}_{1,\max} \cdot \Delta t_1 \dots \dot{\alpha}_{k,\max} \cdot \Delta t_k \dots \dot{\alpha}_{\max,n-1} \cdot \Delta t_{n-1}\right] \\ \mathbf{b}_{vel_\beta} &= \left[\dot{\beta}_{1,\max} \cdot \Delta t_1 \dots \dot{\beta}_{k,\max} \cdot \Delta t_k \dots \dot{\beta}_{\max,n-1} \cdot \Delta t_{n-1}\right] \\ \mathbf{b}_{vel_\gamma} &= \left[\dot{\gamma}_{1,\max} \cdot \Delta t_1 \dots \dot{\gamma}_{k,\max} \cdot \Delta t_k \dots \dot{\gamma}_{\max,n-1} \cdot \Delta t_{n-1}\right] \\ \mathbf{b}_{vel,\max} &= \left[\mathbf{b}_{vel_\alpha}, \; \mathbf{b}_{vel_\beta}, \; \mathbf{b}_{vel_\gamma}\right]^{\mathrm{T}}. \end{aligned}$$

On the basis of considerations expressed in Section 3, the torque constraints can be written as linearized acceleration constraints according to the following equation:

$$\mathbf{M}_{\mathbf{acc}} \cdot \mathbf{x} < \mathbf{b}_{\mathbf{acc}}, \tag{13}$$

where $\mathbf{M}_{\mathbf{acc}}$ is a $3(n-2) \times 3n$ matrix defined as

$$\mathbf{M}_{\mathbf{acc}} = \begin{bmatrix} \mathbf{M}_{\mathbf{acc},1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\mathbf{acc},2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_{\mathbf{acc},3} \end{bmatrix} \quad \text{with} \quad \overbrace{\begin{bmatrix} \Delta t_2 & -(\Delta t_1+\Delta t_2) & \Delta t_1 & & \\ & \Delta t_3 & -(\Delta t_2+\Delta t_3) & \Delta t_2 & \\ & \ddots & \ddots & \ddots & \\ & & \Delta t_{n-1} & -(\Delta t_{n-2}+\Delta t_{n-1}) & \Delta t_{n-2} \end{bmatrix}}^{\mathbf{M_{acc,i}} \;\; (n-2)\times n, \;\; i=1,\dots,3}$$

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

and where $\mathbf{b_{acc}}$ is a $3(n-2)\times 1$ vector defined as

$$\mathbf{b}_{\mathbf{acc}_\alpha} = \left[\ddot{\alpha}_{1,\max}\cdot\Delta t_1\cdot\Delta t_2^2\ \ldots\ \ddot{\alpha}_{n-2,\max}\cdot\Delta t_{n-2}\cdot\Delta t_{n-1}^2\right]$$
$$\mathbf{b}_{\mathbf{acc}_\beta} = \left[\ddot{\beta}_{1,\max}\cdot\Delta t_1\cdot\Delta t_2^2\ \ldots\ \ddot{\beta}_{n-2,\max}\cdot\Delta t_{n-2}\cdot\Delta t_{n-1}^2\right]$$
$$\mathbf{b}_{\mathbf{acc}_\gamma} = \left[\ddot{\gamma}_{1,\max}\cdot\Delta t_1\cdot\Delta t_2^2\ \ldots\ \ddot{\gamma}_{n-2,\max}\cdot\Delta t_{n-2}\cdot\Delta t_{n-1}^2\right]$$
$$\mathbf{b}_{acc} = \left[\,\mathbf{b}_{acc_\alpha}, \mathbf{b}_{acc_\beta}, \mathbf{b}_{acc_\gamma}\,\right]^{\mathrm{T}}.$$

To limit negative speed and acceleration values, it is possible to double the equations for the $(n-1)$ speed constraints and the $(n-2)$ acceleration constraints, defining a single linearized system as

$$\mathbf{M}_{tot} = \begin{bmatrix}\mathbf{M}_{vel}\\ -\mathbf{M}_{vel}\\ \mathbf{M}_{acc}\\ -\mathbf{M}_{acc}\end{bmatrix} \text{ and } \mathbf{b}_{tot} = \begin{bmatrix}\mathbf{b}_{vel,\max}\\ \mathbf{b}_{vel,\max}\\ \mathbf{b}_{acc,\max}\\ \mathbf{b}_{acc,\max}\end{bmatrix}. \tag{14}$$

The problem formulation, according to Eqs. (1), (14), and (2), can be stated as

$$\min_{\mathbf{x}} f(\mathbf{x}) : \begin{cases}\cos(\vartheta_k(\mathbf{x})) < \cos(\vartheta_{k,\max}(\mathbf{x})), & \forall k = 1,\ldots,n,\\ \mathbf{M}_{tot}\cdot\mathbf{x} < \mathbf{b}_{tot},\\ \mathbf{x}_{\min} \le \mathbf{x} \le \mathbf{x}_{\max}.\end{cases} \tag{15}$$

4.3. Optimization problem solver

The system of Eq. (15) is highly non-linear with non-convex constraints in Cartesian space (i.e., there are trigonometric constraints). In addition, the number of free variables is extremely high for the solver, since for each point of the trajectory there are three free variables. On the one hand, the problem optimization through the adoption of evolutionary algorithms, such as genetic algorithms and particle swarm optimization, is infeasible due to the excessive number of free variables. Indeed, despite they are the most common instrument for non-convex domain optimization problem, they need extremely high time to achieve a convergence that is not guaranteed. On the other hand, quad-programming methods, such as the *interior point algorithm*, are particularly suitable for large-scale non-linear programming, as stated in ref. [42], even if they could converge towards local minimum solutions. The authors ran an extensive experimental campaign with the aim to identify the best solver: the Matlab® `fmincon` function that implements *interior point algorithm* was finally selected. This solver demonstrated its superiority for the solution of (15) in terms of less failures with respect to the previously mentioned optimization methods. Furthermore, this method allowed an easier tuning of solver parameters. In order to increase computation speed, two routines post-process the path generated from the CAM before running the optimization solver:

1. Subdivision of the tool path in sub-tracks. A common solution to completing a cladding/AM infill is a zig-zag pattern with different tracks distanced by an offset. At the end of each track, some CAM software packages (i.e., SKM-DCAM) allow the user to define a lead-out space followed by a new lead-in distance. During lead-out and lead-in, the laser is turned off. Consequently, the bend between two contiguous tracks is performed without the material accumulation that is caused by the robot's deceleration and acceleration profiles during direction changes. As a consequence, every track of a zig-zag infill strategy path can be considered a different optimization problem. This routine dramatically eases optimization by reducing the number of points in each path.
2. Feasibility check before the application of the algorithm. Consider a pair of subsequent nodes in a generic path: for each pair, $\vartheta_{k,\max}$ is defined, and it is therefore possible to verify the minimum relative angle between the two deposition axes (Fig. 4). If the maximum allowable speed is too low to span such angle, the process becomes infeasible. To make the cladding feasible in such cases, a higher maximum speed must be imposed, or the orientation constraint $\vartheta_{k,\max}$ must be relaxed.

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

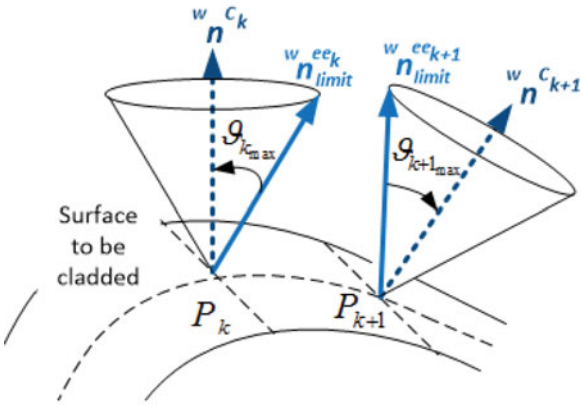


Fig. 4. Feasibility check between two contiguous nodes of the cladding path: an infeasibility is detected if the closest possible orientations require too high a rotation speed.

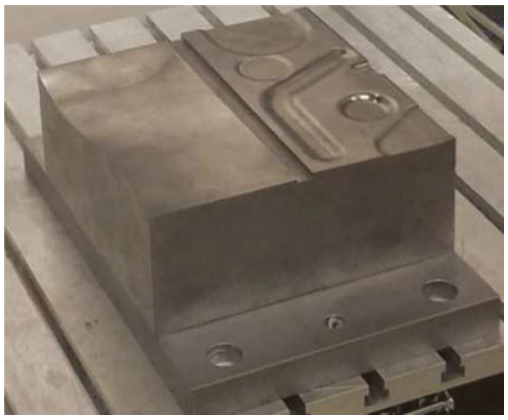


Fig. 5. Object with a generic, irregular surface to be cladded with NiFe 6040 alloy.

Table I. Process parameters for NiFe 6040 alloy.

Power:	1.9 kW	⊘ Wire:	1.2 mm
TCP speed:	0.01 m/s	Cladding rate:	1.25 kg/h
Wire speed:	0.7 m/s	Layer height:	1 mm
Focal height:	199.3 mm		

5. Results

5.1. Setup

This section describes both a simulated and an experimental example. The robot used in the experiments is a KUKA KR30. Figure 5 displays the object to be cladded, and Fig. 6 shows the COAXwire head (a processing optic for laser-wire cladding, developed in Fraunhofer IWS[25]) that was used in the experiments. The COAXwire head is characterized by a wide ϑ_{max} angle because of minor energy absorption problems (common in powder-based solutions). This laser head may therefore derive great benefit from the proposed algorithm. The experiments have been performed using using NiFe 6040 alloy wires. The CAM software used to generate the nominal path for the robot was SKM-DCAM.[22]

5.2. Parameter-setting

The methodology in ref. [13] has been adapted to set the process parameters; the values are summarized in Table I. The value of ϑ_{max} has been chosen equal to 60 deg for each point of the path, since the

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

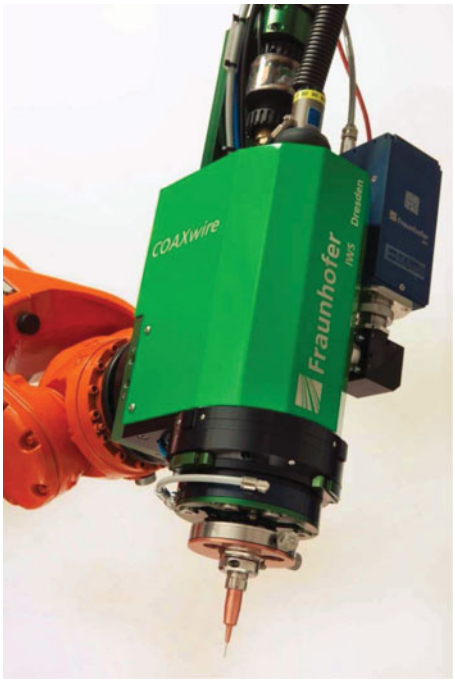


Fig. 6. COAXwire laser head mounted on KR30 KUKA Robot (courtesy of Fraunhofer IWS).

wire-feed system can guarantee good deposition quality under those working conditions. Joint speed and torque limits were determined experimentally. When high-dynamics performance is requested, the robot displays large vibrations. This is reflected in poor tracking performance, i.e., when the robot moves fast, the following error is measurable in millimeters. Experimentally, the maximum velocity and torques that give good tracking performance are about 50% of nominal maximum motor speeds and torques (maximum motor speed and torque data available on KUKA data sheets). However, when these low maximum values are imposed, the "feasibleness check" procedure often raises infeasible path nodes. Therefore, the maximum speed has been increased to 60% of the nominal maximum speed in order to make the cladding of the object feasible. In this way, the residual vibrations at the maximum velocity were in the tenths of millimetres and therefore have been considered negligible.

5.3. Simulation results

Figure 7 compares the path generated from SKM-DCAM and from the optimization algorithm. The cladding tracks computed by SKM-DCAM display discontinuous orientations, and the resulting path does not respect the speed constraints. The optimization algorithm, in contrast, slightly smooths the DCAM output, granting the imposed velocity and torque limits thanks to its exploitation of all the available range of ϑ.

Figures 8–10 compare the optimization algorithm with two other methods: (i) a simple method based on rotation-speed saturation and (ii) a simple method whereby the laser head uses a constant orientation along the entire path (the angle imposed is the mean of the CAM surface angles evaluated along the path). These two methods, which are extremely simple, are often the heuristic solutions implemented in different CAMs to solve the smoothness problem, but they are not optimal. Indeed, Fig. 8(b) shows that the path calculated from such heuristics is far from optimal along the considered trajectory. Figures 8(a), 9, and 10 show that the two heuristics are not able to ensure either the technological constraint $\vartheta_{\max}$ or the kino-dynamics constraints (angles and speeds).

The optimally computed path is superior to the paths computed using other methods because of the convergence of the `fmincon` function. Indeed, although the optimization solver guarantees that the hard limits are always respected (thanks to the pre-optimization feasibility check algorithm), the convergence cannot be ensured; convergence depends on the number of points, pattern, constraints,

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

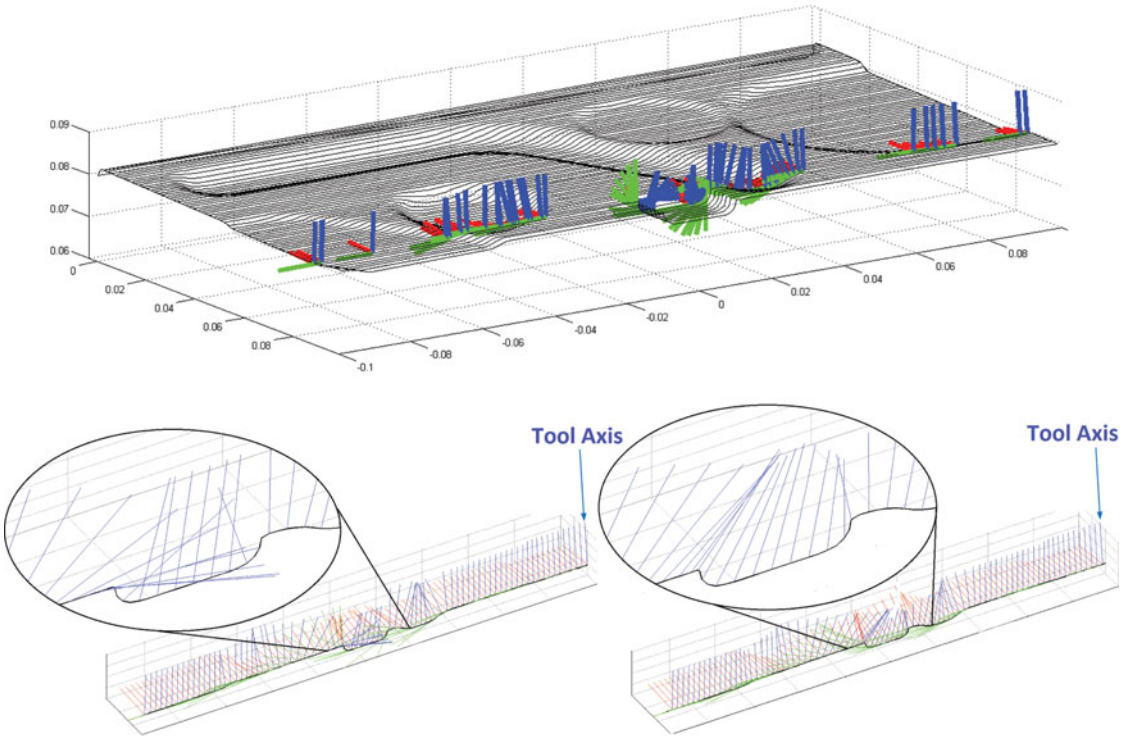


Fig. 7. A cladding track. Left: Original CAM raw orientations. Right: Smooth orientations after optimization algorithm. The time execution of the path is 23 s, corresponding to a linear TCP velocity of 10 mm/s.

etc. In the case of convergence, the minimum found by the interior point algorithm may not be the global minimum, especially on paths with a great number of nodes.

Therefore, `fmincon` is unsatisfactory in order to guarantee the convergence of the constrained optimization for every possible cladding trajectory, and future research will improve the optimization solver method.

5.4. Experimental results

A KUKA Robot Language (KRL) file was generated based on the output path produced by the optimization method. The post-processor that generated the KRL file was developed internally.

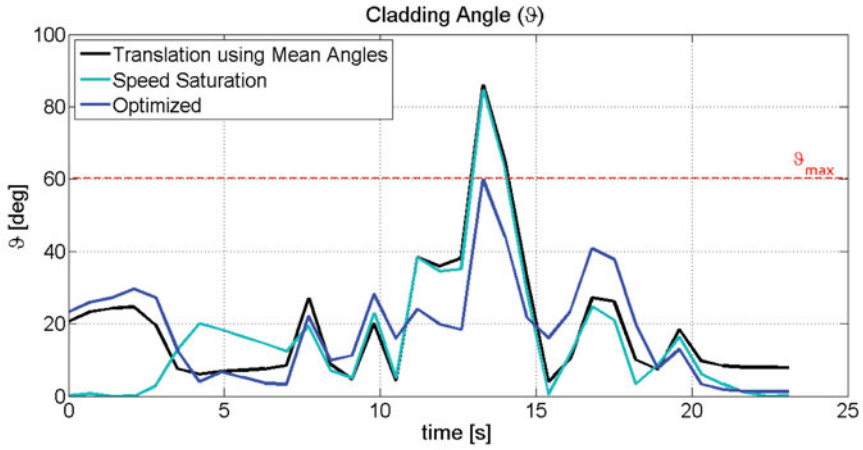
The experiments confirmed superior smoothness in the robot movement when the optimization algorithm is used. The execution of the path generated by the optimization method produces significantly lower robot vibrations than did the execution of the path generated by SKM-DCAM. A visual comparison of the original head orientation and the optimized orientation is in Fig. 11.

6. Conclusion and Future Works

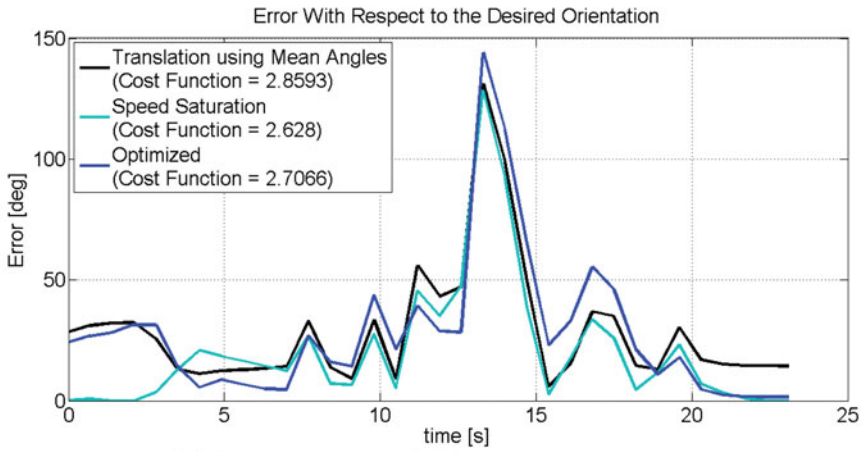
This paper focuses on managing the orientation for a deposition head mounted on an anthropomorphic robot during laser cladding of free-form surfaces. The best orientation possible has been found by solving a constrained optimization problem with an interior point algorithm implemented in the Matlab® `fmincon` function. The problem included the following technological and kino-dynamics constraints:

1. A maximum allowable deviation of the laser head deposition axis w.r.t. the normal direction of the underlying surface, expressed as a non-linear constraint.
2. A boundary for possible orientations, in order to ensure collision avoidance and joint limits avoidance. This boundary was computed using the Flexible Collision C++ Library (FCL).

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press



(a) Evolution of ϑ (relative angle between the normal to the surface and the laser head deposition axis) values over time



(b) Evolution of $\vartheta - \vartheta^{CAM}$ over the time. ϑ^{CAM} results from standard CAM computation. In the labels, the Cost Function values for each experiment is reported.

Fig. 8. Comparison of three methods in tracking the path of Fig. 7: (i) maintaining a constant orientation (black line); (ii) filtering the CAM Cardan angles imposing the re-orientation speed saturation (cyan line); (iii) optimizing the orientation (blue lines). (a) Evolution of ϑ (relative angle between the normal to the surface and the laser head deposition axis) values over time. (b) Evolution of $\vartheta - \vartheta^{CAM}$ over the time. ϑ^{CAM} results from standard CAM computation. In the labels, the cost function values for each experiment is reported.

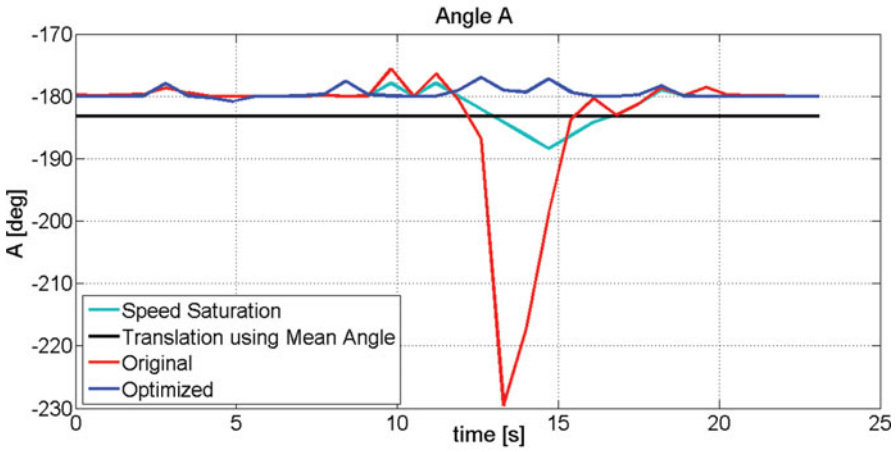
3. A maximum allowable joint speed and torque, in order to guarantee accurate trajectory tracking, without vibrations, by the robot's interpolated motion.

The cost function for the algorithm is proportional to the distance between the optimization variable and the desired orientation. This desired orientation is always perpendicular to the surface in order to have good energy absorption and laser reflection.

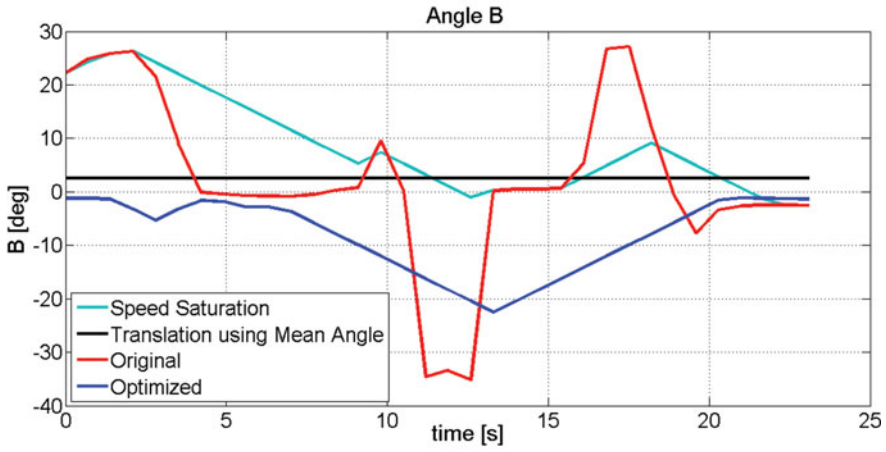
Simulated and real tests were performed, showing that the algorithm output can provide an optimal and automated way to generate robot cladding trajectories.

It is worth noting that the method presents some limitations when the robot is close to the workspace boundaries, and the wrist is close to a singular configuration, since the basic simplification assumptions lose their validity. In addition, "fmincon" displays convergence issues for long paths. Both these limitations will be addressed in future works. Finally, in future works, an extensive experimental campaign aiming at evaluating the obtained surface quality will be presented and a implementation within a commercial CAM software will be considered.

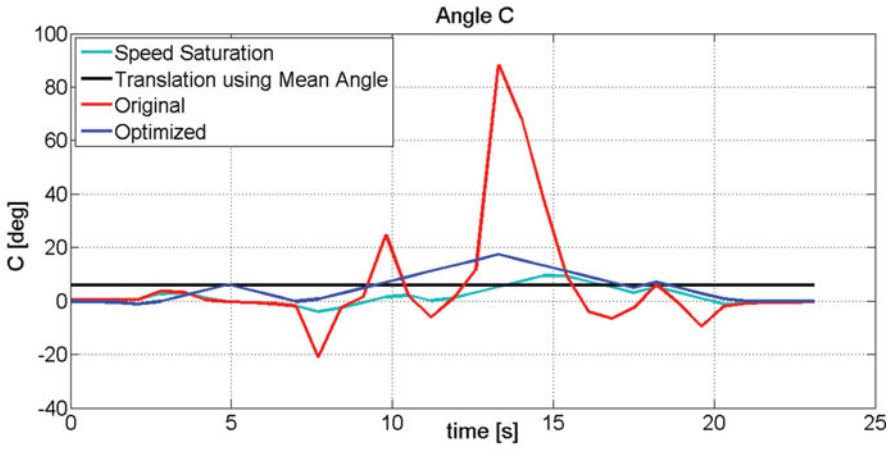
https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press



(a) Cardan Angle A



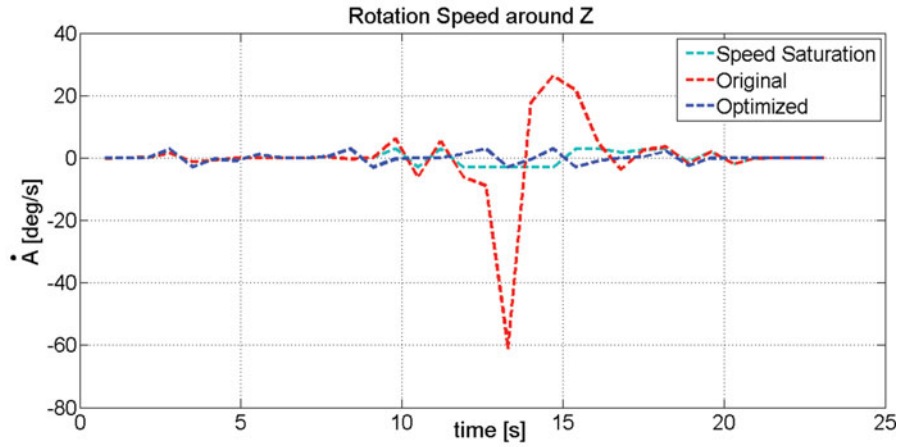
(b) Cardan Angle B



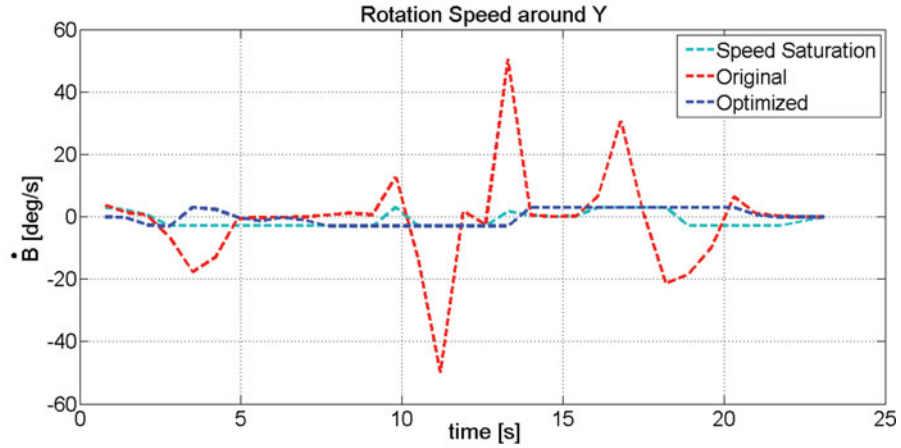
(c) Cardan Angle C

Fig. 9. Comparison of four methods in tracking the path of Fig. 7. (i) The red lines represent the Cardan angles as computed from the CAM; (ii) the cyan lines represent the Cardan angles when the nominal re-orientation velocities are saturated; (iii) the black line represents the Cardan angles when they are maintained constant over the experiment, and equal to the mean value (the angle imposed is the mean of the CAM surface angles evaluated along the path); (iv) the blue lines represent the Cardan angles computed by the optimization method. Particularly, methods (ii) and (iii) do not allow the correct execution of the movement since the constraint $\vartheta_{\max}$ is not respected (see Fig. 8). (a) Cardan angle A. (b) Cardan angle B. (c) Cardan angle C.

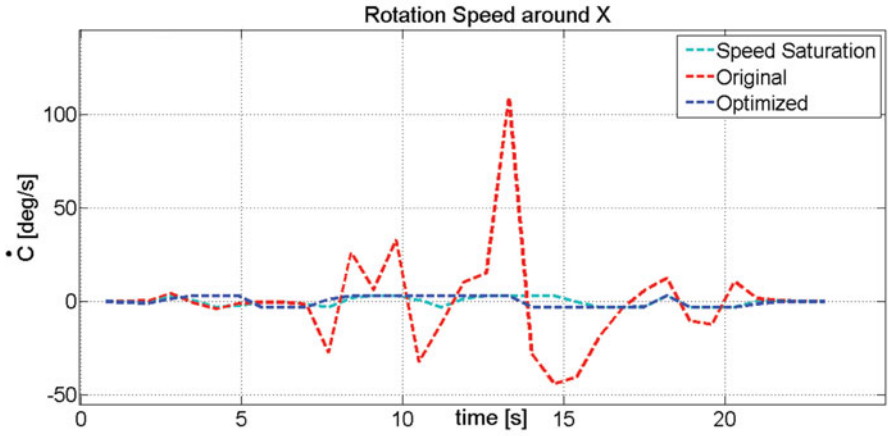
https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press



(a) Cardan Angle A



(b) Cardan Angle B



(c) Cardan Angle C

Fig. 10. First derivative of the Cardan angles in tracking the path in Fig. 7. The same methods described in Fig. 9 are compared. (a) Cardan angle A. (b) Cardan angle B. (c) Cardan angle C.

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

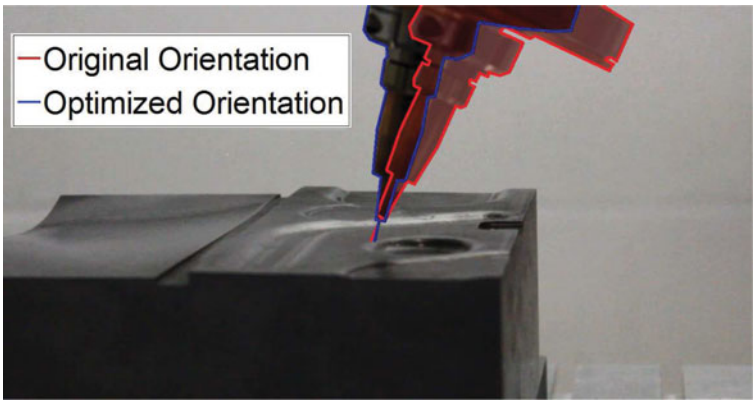


Fig. 11. Real robot motion test: comparison between COAXWire head orientations on a cladding trajectory.

Acknowledgements

This work has been carried out under a collaboration between the Institute of Industrial Technologies and Automation (ITIA) of the National Research Council of Italy (CNR) and the Fraunhofer Institute for Material and Beam Technology (IWS). The authors would like to thank M. Beschi (CNR-ITIA) for his help and support.

References

1. T. Wohlers, "Wohlers report 2014: 3D printing and additive manufacturing state of industry," *Wohlers Associates Inc* (2014).
2. G. Chryssolouris, S. Zannis, K. Tsirbas and C. Lalas, "An experimental investigation of laser cladding," *CIRP Ann. -Manuf. Technol.* **51**(1), 145–148 (2002). doi:10.1016/S0007-8506(07)61486-3.
3. I. Gibson, D. W. Rosen and B. Stucker, *Additive Manufacturing Technologies* (Springer, New York, 2010).
4. S. Nowotny, S. Scharek, E. Beyer and K. H. Richter, "Laser beam build-up welding: Precision in repair, surface cladding, and direct 3D metal deposition," *J. Therm. Spray Technol.* **16**(3), 344–348 (2007).
5. J. Gao, X. Chen, O. Yilmaz and N. Gindy, "An integrated adaptive repair solution for complex aerospace components through geometry reconstruction," *Int. J. Adv. Manuf. Technol.* **36**(11–12), 1170–1179 (2008).
6. B. Denkena, V. Boess, D. Nespor, F. Floeter and F. Rust, "Engine blade regeneration: A literature review on common technologies in terms of machining," *Int. J. Adv. Manuf. Technol.* **81**(5–8), 917–924 (2015).
7. Y. Y. Zhu Y, X. J Tian, J. Li and H. M. Wang, "The anisotropy of laser melting deposition additive manufacturing Ti-6.5Al- 3.5Mo-1.5Zr-0.3Si titanium alloy," *Mater. Des.* **67**, 538–542 (2015).
8. C. Hong *et al.*, "Laser additive manufacturing of ultrafine A-structural materials properties," *Microstruct. Process.* **635**, 118–128 (2015).
9. B. Nie, L. M. Yang, H. Huang, S. Bai, P. Wan and J. Liu, "Femtosecond laser additive manufacturing of iron and tungsten parts," *Appl. Phys. A—Mater. Sci. Process.* **119**(3), 1075–1080 (2015).
10. H. Bikas, P. Stavropoulos and G. Chryssolouris, "Additive manufacturing methods and modelling approaches: A critical review," *Int. J. Adv. Manuf. Technol.* **83,** 389–405 (2016).
11. C. Lalas, K. Tsirbas, K. Salonitis and G. Chryssolouris, "An analytical model of the laser clad geometry," *Int. J. Adv. Manuf. Tech.* **32**, 34–41 (2007).
12. J. Liu and L. Li, "In-time motion adjustment in laser cladding manufacturing process for improving dimensional accuracy and surface finish of the formed part," *Opt. Laser Technol.* **36**(6), 477–483 (2004).
13. A. Calleja, I. Tabernero, A. Fernández, A. Celaya, A. Lamikiz and L. N. Lopez de Lacalle, "Improvement of strategies and parameters for multi-axis laser cladding operations," *Opt. Lasers Eng.* **56**, 113–120 (2014).
14. M. Xu, L. Jianfeng, J. Jibin and B. Li, "Influence of powders and process parameters on bonding shear strength and micro hardness in laser cladding remanufacturing," *Proc. CIRP* **29**, 8049 (2015). doi:10.1016/j.procir.2015.02.088.
15. A Arntz, M Wegener and Y Liu, "Computer aided manufacturing supported process planning of additive manufacturing by laser deposition welding," *J. Laser Appl.* **27**, S14002 (2015) doi:https://doi.org/10.2351/1.4823748.
16. V. Ocelik, M. Eekma, I. Hemmati and J. D. Hosson, "Elimination of start/stop defects in laser cladding," *Surf. Coat. Technol.* **206**(8–9), 2403–2409 (2012).
17. G. Q. Zhang, X. Li, R. Boca, J. Newkirk, B. Zhang, T. A. Fuhlbrigge, H. K. Feng and N. J. Hunt, "Use of Industrial Robots in Additive Manufacturing – A Survey and Feasibility Study," *Proceedings of the 41st International Symposium on Robotics ISR/Robotik 2014, VDE* (2014) pp. 512–517.

18. J. Ruan, K. Eiamsa-ard and F. W. Liou, "Automatic process planning and toolpath generation of a multiaxis hybrid manufacturing system," *J. Manuf. Process.* **7**(1), 57–68 (2005).
19. A. Molina, A. H. Al-Ashaab, T. I. A. Ellis, R. I. M. Young and R. Bell, "A review of computer-aided simultaneous engineering systems," *Res. Eng. Des.* **7**(1), 38–63 (1995).
20. W. Sheng, N. X. Heping Chen, Y. Chen and M. Song, "Surface Partitioning in Automated CAD-Guided Tool Planning for Additive manufacturing," *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS2003* (Oct. 2, 2003) pp. 2072–2077.
21. Siemens. PLM NX Hybrid Additive Manufacturing Software (2016). [Online]. Available: www.plm.automation.siemens.com/en_us/products/nx/for-manufacturing/cam/. [Last accessed 14 Jan. 2018].
22. SKM-Informatik. DCAM (2016). [Online]. Available: www.skm-informatik.de/skm/cam/skm-dcam-at-a-glance.html. [Last accessed 14 Jan. 2018].
23. A. Lasemi, D. Xue and P. Gu, "Recent development in CNC machining of freeform surfaces: A state-of-the-art review," *Comput.-Aided Des.* **42**(7), 641–654 (2010).
24. F. Sellmann, T. Haas, H. Nguyen, S. Weikert and K. Wegener, "Orientation smoothing for 5-axis machining using quasi-redundant degrees of freedom," *Int. J. Autom. Technol.* **10**(2), 263 (2016).
25. S. Nowotny, S. Thieme, D. Albert, F. Kubisch, R. Kager and C. Leyens, "Generative Manufacturing and Repair of Metal Parts through Direct Laser Deposition Using Wire Material," **In**: Digital Product and Process Development Systems. IFIP Advances in Information and Communication Technology (G. L. Kovács and D. Kochan, eds.) vol. 411, (Springer, Berlin, Heidelberg, 2013).
26. B. Donald, P. Xavier, J. Canny and J. Reif, "Kinodynamic motion planning," *J. ACM* **40**(5), 1048–1066 (1993).
27. J.-C. Latombe, *Robot Motion Planning* (Kluwer Academic Publishers, Boston, 1991).
28. L. Kavraki, P. Svestka, J. -C. Latombe and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.* **12**(4), 566–580 (1996).
29. S. LaValle and J. Kuffner, "Randomized kinodynamic planning" *Int. J. Robot. Res.* **20**(5), 378–400 (2001).
30. S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.* **30**(7), 846–894 (2011).
31. D. Webb and J. Van den Berg, "Kinodynamic RRT*: Optimal Motion Planning for Systems with Linear Differential Constraints," *Proceedings of the IEEE International Conference on Robotics and Automation ICRA*, Karlsruhe, Germany (May 6–10, 2013)
32. AC&E. CimStation (2016). [Online]. Available: http://www.acel.co.uk/cimstation-robotics/. [Last accessed 14 Jan. 2018].
33. Siemens. RobCad (2016). [Online]. Available: https://www.plm.automation.siemens.com/. [Last accessed 14 Jan. 2018].
34. ABB Robotics. RobotStudio (2016). [Online]. Available: http://new.abb.com/products/robotics/robotstudio. [Last accessed 14 Jan. 2018].
35. Visual Components. 3DAutomate (2016). [Online]. Available: http://www.visualcomponents.com/products/3dautomate. [Last accessed 14 Jan. 2018].
36. Dassault Systems. Delmia (2016). [Online]. Available: http://www.3ds.com/products-services/delmia. [Last accessed 14 Jan. 2018].
37. Chalmers. Industrial Path Planner (2016). [Online]. Available: http://www.fcc.chalmers.se/software/ips. [Last accessed 14 Jan. 2018].
38. GAMMA Group, University of North Carolina, Chapel Hill. FCL: A Flexible Collision Library (2016). [Online]. Available: http://gamma.cs.unc.edu/FCL. [Last accessed 14 Jan. 2018].
39. I. Sucan, URDF ROS Package (2016). [Online]. Available: http://wiki.ros.org/urdf. [Last accessed 14 Jan. 2018].
40. J. H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numer. Math.* (1), 84–90 (1960).
41. B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics: Modelling, Planning and Control* (Springer Science & Business Media, Springer-Verlag London, 2010).
42. R. H. Byrd, M. E. Hribar and J. Nocedal, "An interior point algorithm for large scale nonlinear programming," *SIAM J. Opt.* **9**(4), 877–900 (2000).
43. J. Um, M. Rauch, J.Y. Hascot *et al.*, "STEP-NC compliant process planning of additive manufacturing: remanufacturing," *Int. J. Adv. Manuf. Technol.* **88**, 1215–1230 (2017).
44. M. Zhao, X. Gao and Q. Zhang, "An efficient stochastic approach for robust time-optimal trajectory planning of robotic manipulators under limited actuation," *Robotica* **35**(12), 2400–2417 (2017).

Appendix

A portion of the original G-CODE is provided in this appendix. This G-CODE has been generated by SKM-DCAM software in the KUKA KRL language. This portion of the G-CODE is represented in Fig. 12.

;; Track 56
LIN {X 85.296, Y -98.6, Z 78.727, A -179.983, B -2.629, C -0.35617, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

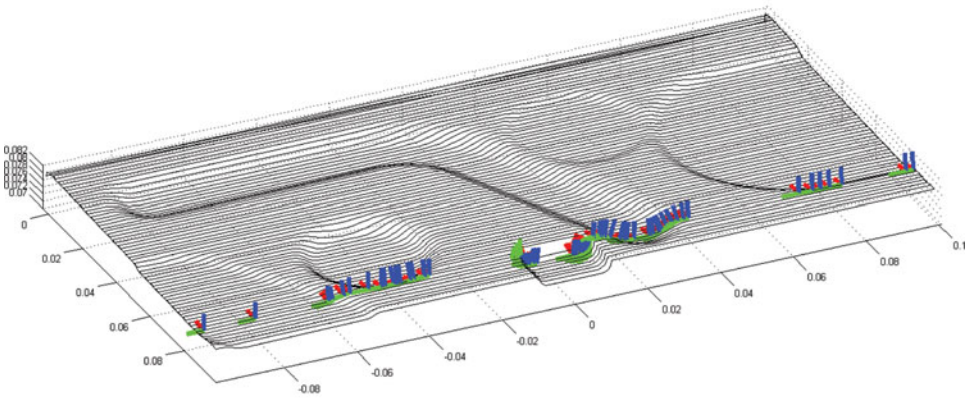


Fig. 12. Portion of path with G-CODE reported in the appendix.

LIN {X 86.896, Y -98.6, Z 78.656, A -179.989, B -2.554, C -0.38026, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -96.6, Z 78.669, A -179.982, B -2.551, C -0.38026, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL $OUT[1]=TRUE
LIN {X 86.896, Y -82.444, Z 78.766, A -179.976, B -2.461, C -0.54253, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -62.283, Z 78.948, A -179.816, B -3.680, C -2.86269, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -61.124, Z 79.066, A -177.825, B -9.658, C -12.7531, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -58.186, Z 79.831, A -179.048, B -6.351, C -8.54243, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -56.458, Z 79.993, A -179.791, B -3.13, C -3.79399, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -51.363, Z 80.034, A -179.990, B 1.101, C 0.4745, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -47.605, Z 79.932, A -179.631, B 8.251, C 2.56905, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -46.801, Z 79.882, A -179.207, B 15.976, C 2.87743, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -44.156, Z 79.823, A 178.153, B 18.21, C -5.8861, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -43.426, Z 79.793, A -179.826, B 16.365, C 0.61504, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -42.398, Z 79.802, A 179.641, B 16.127, C -1.29019, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -39.666, Z 79.913, A 178.975, B 15.810, C -3.75664, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -38.663, Z 80.004, A 179.026, B 12.646, C -4.4384, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -35.834, Z 80.103, A 179.79, B 4.654, C -2.53534, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -34.495, Z 80.166, A 179.978, B 1.082, C -1.15307, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -9.516, Z 80.166, A 152.629, B -0.698, C 88.65151, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -9.451, Z 79.431, A 154.078, B -6.402, C 77.07685, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -9.261, Z 78.862, A 157.570, B -10.907, C 65.37327, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

LIN {X 86.896, Y -8.942, Z 78.328, A 162.594, B -13.585, C 53.15602, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -8.51, Z 77.863, A 168.156, B -13.975, C 40.96718, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -8.091, Z 77.572, A 166.989, B -19.650, C 34.49329, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -7.32, Z 77.226, A 177.572, B -10.231, C 13.42202, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -5.833, Z 77.052, A 179.174, B -10.223, C 4.63822, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y -5.0, Z 77.02, A 179.701, B -10.7912, C 1.59081, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 5.0, Z 77.02, A -179.900, B -10.798, C -0.53018, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 5.833, Z 77.052, A -179.174, B -10.22, C -4.63822, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 7.32, Z 77.226, A -175.677, B -13.241, C -18.26309, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 8.091, Z 77.572, A -166.989, B -19.650, C -34.49329, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 8.511, Z 77.863, A -168.156, B -13.975, C -40.96718, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 8.942, Z 78.328, A -160.797, B -12.932, C -57.27604, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 9.261, Z 78.862, A -157.570, B -10.907, C -65.37327, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 9.451, Z 79.431, A -154.078, B -6.402, C -77.07734, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 9.51, Z 80.201, A -152.833, B 0.20512, C -90.40016, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 10.572, Z 80.418, A -179.982, B 0.460, C 2.12049, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 12.117, Z 80.341, A -179.76, B 2.246, C 5.91709, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 12.716, Z 80.265, A -179.525, B 3.297, C 8.19813, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 13.638, Z 80.096, A -178.878, B 5.125, C 12.3554, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 15.109, Z 79.674, A -176.253, B 8.989, C 22.73739, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 17.698, Z 78.549, A -176.562, B 8.797, C 21.44144, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 18.951, Z 78.174, A -178.316, B 6.396, C 14.77345, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 19.797, Z 78.016, A -179.091, B 6.344, C 8.16356, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 21.784, Z 77.871, A -179.811, B 3.511, C 3.07369, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 26.01, Z 77.877, A 179.948, B 1.452, C -2.01501, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 27.184, Z 77.999, A 179.545, B 4.206, C -6.17408, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 28.784, Z 78.194, A 178.070, B 6.778, C -15.9297, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 29.556, Z 78.474, A 177.176, B 7.497, C -20.70535, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press

LIN {X 86.896, Y 31.166, Z 79.098, A 177.395, B 5.981, C -23.58121, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 32.473, Z 79.693, A 178.210, B 4.142, C -23.38435, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 33.824, Z 80.128, A 179.31, B 2.543, C -15.11852, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 35.236, Z 80.396, A 179.950, B 0.487, C -5.76816, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 36.658, Z 80.479, A -179.977, B -0.56, C -2.26216, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 67.083, Z 80.448, A -179.998, B 0.136, C 0.60735, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 70.821, Z 80.371, A -179.848, B 4.930, C 1.76465, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 72.944, Z 80.241, A -179.731, B 7.177, C 2.14816, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 75.482, Z 80.185, A -179.917, B 6.246, C 0.75625, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 78.985, Z 80.151, A -179.967, B 6.223, C 0.30316, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL
LIN {X 86.896, Y 96.6, Z 80.08, A -179.974, B 6.3505, C 0.2352, E1 0.0, E2 0.0, E3 0, E4 0, E5 0, E6 0} C_VEL $OUT[1]=FALSE

https://doi.org/10.1017/S0263574718000048 Published online by Cambridge University Press