Contents lists available at SciVerse ScienceDirect

CIRP Annals Manufacturing Technology

Journal homepage: www.elsevier.com/locate/cirp

Optimized joint motion planning for redundant industrial robots

Gábor Erdős[a,b], András Kovács[a], József Váncza (1)[a,b]

[a] Fraunhofer Project Center PMI, Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary
[b] Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Budapest, Hungary

The paper presents a model and solution method for optimized robot joint motion planning of redundant industrial robots that execute a set of tasks in a complex work environment, in face of various technological and geometric constraints. The approach aims at directly exploiting redundancy to optimize a given performance measure, e.g., cycle time. Alternative configurations along the path are captured in a graph model, whereas bi-directional transition between task and configuration spaces facilitates generating relevant, collision-free configurations only. Re-parametrization of the trajectory warrants compliance with the robot's kinematic constraints. Successful application of the method is demonstrated in remote laser welding.

Robot; Tool path; Optimization

1. Introduction

Industrial robots have inhabited production systems for a long time, still their effective and efficient *automatic off-line programming* (OLP) poses serious challenges [1]. Depending on their actual tasks (such as welding, cutting, grinding, assembly, measurement, etc.), all domain-specific technological constraints must be complied with. In parallel with meeting task level constraints on ordering and resource assignment, a *joint motion plan* has to be generated that is compatible with the robot's kinematic model and minimizes some global criterion like cycle time, jerk, energy demand or a combination thereof. Any motion plan must be collision-free, a requirement that can be verified in case of intricate geometrical relationships and potential interactions of objects (parts, tools, fixtures, etc.) only on the task level. *Redundancy*—when the robot has more degrees of freedom than needed for performing the tasks—provides room for optimization, because, at least in principle, infinitely many joint configurations may result in executing the same task.

The paper investigates the issue whether and how automatic OLP aimed at a global optimization criterion can be supported by a bi-directional *transition* between relatively high-dimensional task and configuration spaces. This approach is in contrast with the traditional hierarchical refinement methods where task sequencing and path planning [2] is followed by inverse kinematics resulting in some executable motion plan [3]. However, computational complexity implied by high dimensions, redundancy, cluttered work environment and task space constraints allow only for the use of custom-tailored solution methods [4]. Alternatively, direct kinematic model of the robot can be used in search for a series of feasible configurations, even in a bilevel optimization scheme [5].

For a working example, the emerging technology of robotic *remote laser welding* (RLW) will provide the background [6]. RLW joins metal sheets by a laser beam that is emitted from a scanner that is mounted as an end-effector on an industrial robot, see Fig. 1. This contactless welding comes together with technological constraints on the inclination angle and the focal length of the beam, as well as on the welding speed and power applied to individual welding stitches. Ideally, the scanner processes the stitches on the fly, without idle time as the robot moves along its optimized path. However, this can seriously be hampered by the geometrical distribution of welding stitches and the *fixture*—a dense structure of clamps, seats and structural elements—that is responsible for part-to-part gap control. In industrial practice, robot programming for RLW is typically performed by time consuming and inefficient on-line teach-in methods. The first automated OLP methods for RLW were proposed in [7] whose performance was later surpassed by new task modelling and integrated task sequencing and path planning methods [6][8]. Recently, all main aspects of introducing RLW into the automotive industry have been discussed in [9], while [10] took a systemic approach to workstation design and motion planning in multi-robot spot welding. The above works put much emphasis on cycle time optimized task sequencing and path planning, but were content with providing heuristic, *feasible* solutions to the corresponding robot motion planning problem. In fact, redundancy of robots has not been exploited, optimality of motion plans in the configuration space has not been explicitly tackled, hence, the final executed robot codes might easily run with fairly suboptimal performance.

Furthermore, the paper has also a broader motivation by recalling the chances of cross-fertilization of ideas in robot joint motion planning and trajectory generation for *CNC machine tools* [11]. In the latter domain as well, cycle time reducing trajectory planning schemes are sought that comply with both task level constraints (such as precisely machined freeform surfaces, or smooth corners with desired tolerances) and drive-specific constraints on speed, acceleration and jerk limits [12][13].

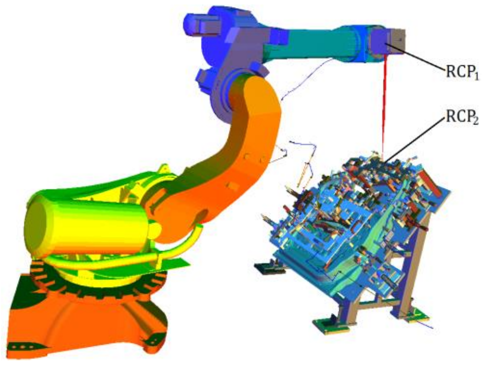

Fig. 1. RLW robot welding a car door in a fixture.

2. Problem statement

Given a *robot path* defined in the *task space*, together with the kinematic model of the robot and the geometric model of the robot and its environment, *joint motion planning* addresses finding a continuous-time *robot trajectory* in the *robot configuration space* that realizes the desired path and satisfies the relevant constraints. These constraints tackle all aspects of feasibility, such as technology (e.g., the execution of the effective tasks according to given motion laws), geometry (e.g., collision avoidance), as well as the kinematic constraints of the robot. Moreover, a *performance measure* such as cycle time or energy consumption must be optimized.

This paper looks at joint motion planning as a crucial step of the overall OLP *workflow* [6] where, departing from the workpiece model and the definition of the tasks to be accomplished by the robot, *task sequencing* and *path planning* are responsible for constructing the optimal robot path. However, these steps work in the task space, and therefore, are not suitable to address criteria that can only be interpreted in the robot's configuration space. This is exactly the duty of joint motion planning while converting the task space path into a configuration space trajectory. The detailed workstation design required for this, including the placement and calibration of the workpiece in the robot's reference frame, is specified by *setup planning*. The robot trajectory computed by joint motion planning, typically after verification by simulation, is then submitted for *commissioning*: the robot code is automatically generated and uploaded to the robot controller. In the sequel focus is set on the formal model and solution of the joint motion planning problem.

3. Modelling the optimized joint motion planning problem

In order to reflect the redundancy of the robot's kinematic chain, the robot path in the input is defined in the Cartesian task space coordinate system for multiple *robot controlled points* (RCPs) in discrete time instants. Each RCP path is given as a series of path base points $(\boldsymbol{r}^k_{t_1}, \boldsymbol{r}^k_{t_2}, \dots, \boldsymbol{r}^k_{t_T})$, where $\boldsymbol{r}^k_{t_i} \in \mathbb{R}^3$ and k=1, ..., K is the number of RCPs. The multiple RCP paths are synchronized, i.e., different RCP paths are defined over the same set of time instants, and therefore, the ensemble of all RCP positions at time t can be denoted by $\boldsymbol{r}^*_t = [\boldsymbol{r}^1_t, \dots, \boldsymbol{r}^K_t]^T$. Path sections are classified into two types: *effective task* and *idle movement* sections. Time instants t_i include (but are not limited to) the start and end times of all effective tasks. Set A includes the base point indices i such that the robot performs an effective task in time interval $[t_i, t_{i+1})$. For each effective task, the input contains a technologically feasible *motion law* $\boldsymbol{r}^k(t)$. Hence, the continuous-time $\boldsymbol{r}^k(t)$ function is defined only over time intervals $[t_i, t_{i+1})$ such that $i \in A$.

In case of RLW, the two RCPs (K=2) are the end point of the laser beam, focused at the welding stitches, and the scanner center point, see Fig. 1 and 2. The two synchronized RCP paths guarantee technological feasibility, including aspects of welding speed (by motion laws), incidence angle and laser focal length (by the relative position of the corresponding RCP base points), as well as collision avoidance for the laser beam. Collision avoidance for the complete robot will be achieved by joint motion planning.

Finally, the *robot's kinematic model* is also given, in terms of $\theta^j(t)$ joint values, joint ranges, velocity and acceleration limits $\dot{\theta}^j_{MAX}$ and $\ddot{\theta}^j_{MAX}$ for each j=1, ..., N joint. The input also contains the *CAD model* of the complete workstation, including the robot, the workpiece and the fixture, which will be used for collision avoidance.

The output is a continuous-time *motion plan* $\boldsymbol{c}(t)$ defined in the robot configuration space that implements the prescribed RCP paths, respects the motion laws and the robot's kinematic limits, is free of any collisions, and is (close-to-)optimal according to the defined objective function. It is noted that the motion plan is re-parameterized in time compared to the RCP paths, since path planning in task space cannot deliver feasible and optimal time parameterization in configuration space. Hence, motion planning can be briefly formulated as follows:

$$\boldsymbol{r}^*_t \;\forall t \;\; \rightarrow \;\; \boldsymbol{c}_t \;\forall t \;\; \rightarrow \;\; \boldsymbol{c}(t) = \begin{pmatrix} \theta^1(t) \\ \dots \\ \theta^N(t) \end{pmatrix}$$

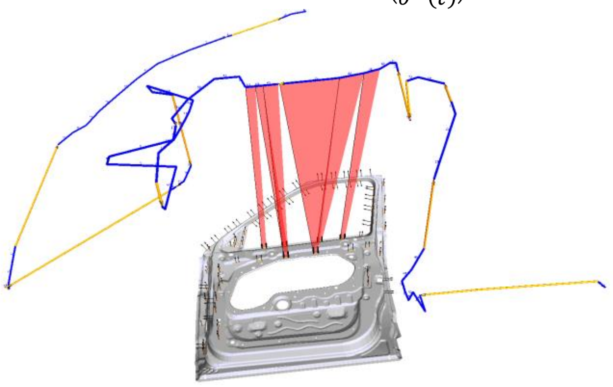

Fig. 2. RCP paths in RLW: scanner path (blue: effective tasks, yellow: idle movement) synchronized with the movement of the laser over stitches.

4. Solution method

The proposed solution method aims to effectively exploiting redundancy so as to optimize the given objective function. For this purpose, it first (1) generates a set of candidate robot configurations for each path base point by solving the inverse kinematic problem, and (2) filters this set for collision-free configurations. Then, it (3) selects the optimal combination of configurations by constructing the shortest path in a graph representation, and finally (4) re-parameterizes the trajectory in time to respect all kinematic constraints of the robot. Each of these steps is presented in detail below.

4.1. Computing inverse kinematics

The direct kinematics (DK) problem $\boldsymbol{r}^*_t = f(\boldsymbol{c}_t)$ transforms the points of the configuration space $(\boldsymbol{c}_t)$ to the robot task space $(\boldsymbol{r}^*_t)$ with a smooth nonlinear vector function f. The inverse kinematics (IK) problem $\boldsymbol{c}_t = f^{-1}(\boldsymbol{r}^*_t)$ maps task space points back to the configuration space, as shown in Fig. 3.

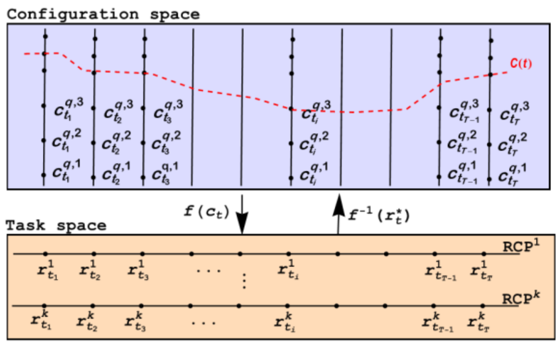

Fig. 3. Illustration of the direct and inverse kinematic transformations.

In case of redundant robots, there are infinitely many IK solutions for a given task space base point. As shown in [14], the infinity of IK solutions can be grouped into a finite and bounded set of disjoint continuous manifolds, called *self-motion* manifolds. Hence, the result of the IK problem can be considered as ordered pairs of a task space base point and a set of joint configurations $\boldsymbol{r}^*_t \rightarrow \{\boldsymbol{c}^{q,s}_t\}, s \in S_q$, where S_q denotes the q-th self-motion manifold. Each self-motion manifold reduces into a single point in case of non-redundant IK problems, and they are referred to as branches of IK solutions (for an RLW example, see Fig. 4).

The goal of IK is to calculate the discretized set $\{\boldsymbol{c}_t^{q,s}\}$ of the self-motion manifolds to prepare the ground for subsequent optimization. An approach is proposed that solves the IK problem in two steps. First, an initial solution $\boldsymbol{c}_t^{q,0}$ is generated for each solution branch by reducing the redundant IK problem into a non-redundant one. Then, starting from this initial solution, the set of discrete solutions is computed for the self-motion manifold associated with the given branch.

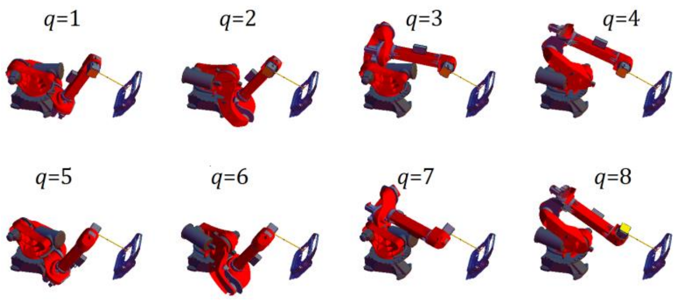

Fig. 4. Eight branches of IK solutions. Branches q=4 and q=8 are feasible.

The RLW robot of the case study has 5 revolute degrees of freedom (DOF) in the robot arm carrying the scanner head, 1 revolute DOF by the actuated mirror in the scanner, and 1 prismatic DOF corresponding to the focal length. By the construction of the robot, only solution branches q=4 and q=8 can satisfy the joint range limits (Fig. 4). The self-motion manifolds corresponding to each solution branch are one dimensional submanifolds that can be parametrized with the θ^6 joint variable.

Initial IK solutions $\boldsymbol{c}_t^{q,0}$, $q \in \{4,8\}$ are determined using an IK algorithm introduced in [6], which combines closed-form and iterative elements, and reduces the redundant IK problem into a non-redundant one by fixing the last revolute DOF, the mirror in the scanner, at midrange ($\theta^6 = 0$). Then, the self-motion manifolds $\{\boldsymbol{c}_t^{q,i}\}, q \in \{4,8\}$ are caluculted by finding solutions with different additional constraints $\theta^6 = p$ while p scans the feasible range of joint θ^6. For each value of p, the corresponding solution is determined by solving the DK equations using an iterative root finding algorithm and an initial solution value received by substituting θ^6=p into $\boldsymbol{c}_t^{q,0}$. The reduced DK function g received by substituting out θ^6 can be formulated as follows:

$$\begin{gathered}
\boldsymbol{r}_t^* = f(\theta_t^1, \theta_t^2, \theta_t^3, \theta_t^4 \theta_t^5, \theta_t^6, F_t)\,, \qquad f\colon \mathbb{R}^7 \to \mathbb{R}^6 \\
g(\theta_t^1, \theta_t^2, \theta_t^3, \theta_t^4 \theta_t^5, F_t) = f(\theta_t^1, \theta_t^2, \theta_t^3, \theta_t^4 \theta_t^5, p, F_t) \\
\boldsymbol{r}_t^* = g(\mathrm{x})\,, \qquad g\colon \mathbb{R}^6 \to \mathbb{R}^6 \\
\boldsymbol{x} = [\theta^1, \theta^2, \theta^3, \theta^4, \theta^5, F\,]^T
\end{gathered}$$

Table 1 below shows an excerpt of the self-motion manifold corresponding to solution branch q=4 for a given task space base point, received by varying θ^6 over the range $[-7°, 9.5°]$. It is noted that the value of F, the laser focal length, is determined by the input RCP paths.

Table 1. Alternative configurations for one path base point, received by sampling for $\boldsymbol{\theta^6}$ in the range $[\boldsymbol{-7°, 9.5°}]$, using solution branch q=4.

θ^1[°]	θ^2[°]	θ^3[°]	θ^4[°]	θ^5[°]	θ^6[°]	F[mm]
-16.757	-32.957	-136.531	-44.741	21.644	-7.0	1052.14
-17.145	-33.117	-136.995	-47.526	21.471	-6.0	1052.14
-17.518	-33.285	-137.484	-50.339	21.254	-5.0	1052.14
…	…	…	…	…	…	…
-19.102	-34.267	-140.316	-65.04	19.478	0.0	1052.14
…	…	…	…	…	…	…
-19.999	-36.921	-147.93	-100.342	11.251	9.5	1052.14

4.2. Filtering for collision-free configurations

To verify the feasibility of each computed configuration $\boldsymbol{c}_t^{q,s}$, the configurations are transformed back to the task space by DK, and collision detection is performed among robot links and all components of the workstation CAD model, including the workpiece and the fixture. If no feasible configuration exists for an RCP base point, then the joint motion planning problem itself is infeasible, and the input RCP paths must be modified.

4.3. Optimizing the configuration space path

Planning the configuration space path requires selecting for each path base point $\boldsymbol{r}_t^*$ one robot configuration $\boldsymbol{c}_t^{q,s}$ from the candidates computed above. In order to find a path that minimizes the cost, a graph representation, the so-called *configuration space graph* (CSG) is constructed as follows.

The CSG is a multi-partite graph with each partition (column of vertices in Fig. 5.) standing for one base point $\boldsymbol{r}_t^*$. The partition contains the feasible robot configurations $\boldsymbol{c}_t^{q,s}$ for the given base point as vertices. There are two additional dummy nodes, S and T as shown in the figure. Every vertex in partition *i* is connected to all vertices in partition *i+1* by a directed edge, characterized by edge weight w_e set according to the actual objective function. Then, the optimal configuration space path corresponds to a minimum cost $S \to T$ path in the CSG, which can be computed efficiently, in $O(|E|+|V|)$ time, where E and V are the number of edges and vertices, respectively.

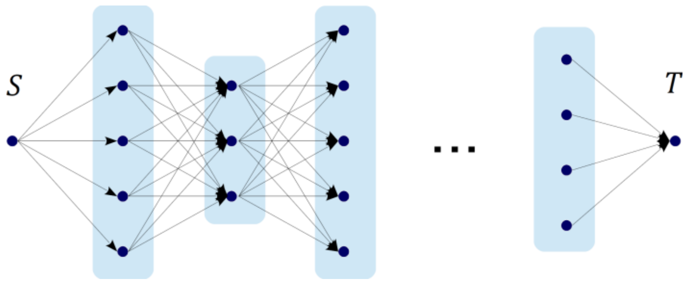

Fig. 5. Small-size illustration of a configuration space graph.

In the RLW case study, where the objective is minimizing the cycle time and motion laws prescribe non-zero velocities at base points, a particular complication is that weights w_e are not well defined: for idle movement edges, the time required for travelling the edge depends on the selection of the neighboring edges as well. To overcome this difficulty, when solving the shortest path problem, a surrogate cost function w'_e is applied that contains the time required for travelling the edge with a trapezoid velocity profile, with zero velocities at the start and end points. The efficiency of this approximation is confirmed by the computational experiments.

4.4. Computing the configuration space trajectory

The time parameterization of the robot path must respect the upper bounds on joint velocity and acceleration as defined in the kinematic model of the robot. Recall that the path consists of a series of effective task and idle movement sections. The time parameterization of effective task sections is determined by the given technology (e.g., applicable welding speed), and fully defined in the input motion laws. At the same time, the parameterization of the idle movement sections can only be approximated in the task space RCP paths. The time function $\boldsymbol{c}(t)$, $t \in [t_i, t_{i+1})$ of an idle movement section must be therefore re-parameterized in such a way that the end time of the section, t'_{i+1}, is minimized subject to the joint velocity and acceleration limits:

$$\dot{\boldsymbol{c}}(t) < \dot{\boldsymbol{\theta}}^{MAX}, \; \ddot{\boldsymbol{c}}(t) < \ddot{\boldsymbol{\theta}}^{MAX}, \qquad \forall\, t \in [t_i, t'_{i+1}).$$

Assume that d consecutive idle movement sections are preceded and followed by effective task sections in the path, where the effective tasks define the configuration velocity vectors $\dot{\boldsymbol{c}}_{t_i}$ and $\dot{\boldsymbol{c}}_{t_{i+d}}$. The time function $\boldsymbol{c}(t), t \in [t_i, t_{i+d})$ of the consecutive sections is then interpolated with a third-order spline function over the control points $\{\boldsymbol{c}_{t_i}, \dots, \boldsymbol{c}_{t_{i+d}}\}$ using the derivatives $\dot{\boldsymbol{c}}_{t_i}$ and $\dot{\boldsymbol{c}}_{t_{i+d}}$. The interpolated joint position function $\theta^j(t)$ can be calculated for each joint separately. Each of these scalar interpolation functions are re-parameterized by contracting or dilating the time parameter until the velocity and acceleration constraints are satisfied and the end time $t_{i+d}^{\theta^j}$ is minimized, as shown in Fig 6. Finally, t'_{i+d} is set to the maximum of the individual joints' calculated end time values:

$$t'_{i+d} = max\left\{t_{i+d}^{\theta^1}, t_{i+d}^{\theta^2}, \dots, t_{i+d}^{\theta^N}\right\}.$$

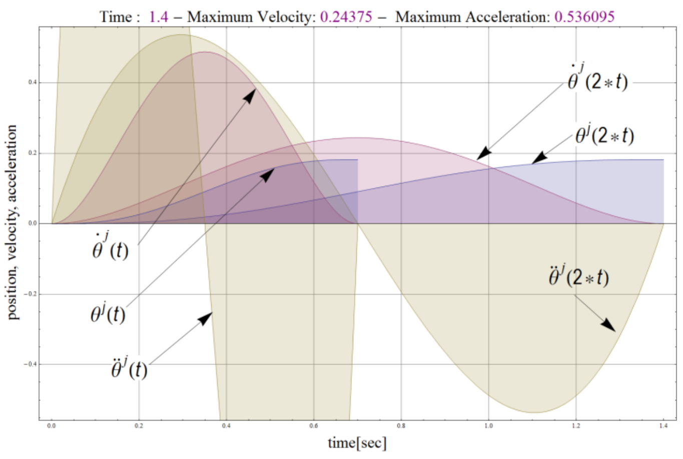

Fig. 6. Re-parameterizing the time interval $[\boldsymbol{t_i}, \boldsymbol{t_{i+d}})$ for one joint variable $\boldsymbol{\theta^j(t)}$, assuming $\dot{\boldsymbol{\theta}}^{\boldsymbol{j}}(\boldsymbol{t_i}) = \dot{\boldsymbol{\theta}}^{\boldsymbol{j}}(\boldsymbol{t_{i+d}}) = \boldsymbol{0}$.

5. Experiments in the RLW domain

Experiments addressed the validation of the approach on the assembly of a car door by RLW using a Comau C4G robot. The workpiece contained 72 welding stitches, distributed on all sides of the door, with total welding time of 38.3 s. The input RCP paths were computed and the output motion plan was converted into an OLP by using the techniques presented in [6][8]. Computing a motion plan using the algorithms proposed here, via generating 14 792 candidate robot configurations, took 3200 s on an Intel i7-4600 2.10GHz computer with 8GB RAM. The feasibility of the computed solution was verified both by simulation and by physically welding a small batch of doors in an experimental RLW workstation presented in [9].

Fig. 7 compares alternative time-parameterized motion plans for the same problem on Gantt charts. Colored stripes denote effective tasks (welding), while white stripes stand for idle movement. The alternative motion plans include:

(1) The input RCP path, with time parameterization determined in the task space using theoretical maximum RCP speeds. Hence, the 45.5 s cycle time of this infeasible plan can be considered as a lower bound on the optimal solution.

(2) A motion plan computed using the approach of [6], based on heuristically fixing the last revolute joint to eliminate the redundancy of the robot. This resulted in a cycle time of 82.6s.

(3) The optimized motion plan achieved by the proposed approach, with a cycle time of 62.6 s. This corresponds to a 24.3% improvement in cycle time, and notably 45.1% in idle time compared to the heuristic solution. This gain can be attributed to effectively exploiting the redundancy of the kinematic chain to optimize the given objective function.

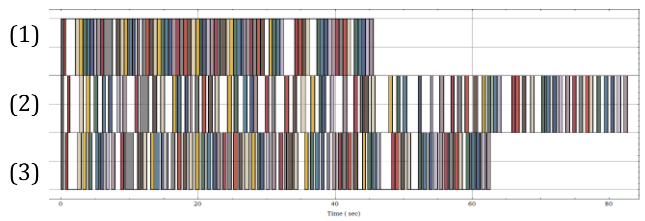

Fig. 7. Comparison of alternative RLW robot trajectories: lower bound estimation (1), heuristic solution (2), and optimized motion plan (3).

6. Conclusions

The paper suggested a novel approach to exploiting the kinematic redundancy of industrial robots to optimize their joint motion plans. The representation based on multiple synchronized RCP paths is generic enough to capture the results of advanced task sequencing and path planning algorithms. A graph representation of the optimization problem is built from all candidate robot configurations generated by inverse kinematics, whose geometric feasibility is checked by bi-directional transition between the task and configuration spaces to avoid any collisions. Final re-parametrization of the trajectory warrants compliance with joint velocity and acceleration limits. The successful use of the method has been demonstrated in RLW, where it significantly reduced idle times compared to a heuristic solution. Operations in high-dimensional constrained task spaces—like 5D grinding, robotic measurement, assembly, etc.—provide further application potential for the approach.

Acknowledgement

This research has been supported by the EU H2020 grant SYMBIO-TIC No. 637107 and the Hungarian Scientific Research Fund (OTKA), Grant No. 113038.

References

[1] Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) *Robotics: Modelling, Planning and Control*. Springer.

[2] Alatartsev S, Stellmacher S, Ortmeier F (2015) Robotic Task Sequencing Problem: A Survey. *Journal of Intelligent & Robotic Systems* 80(2):279-298.

[3] Gasparetto A, Zanotto V (2010) Optimal Trajectory Planning for Industrial Robots. *Advances in Engineering Software* 41(4):548-556.

[4] Colomé A, Torras C (2015) Closed-Loop Inverse Kinematics for Redundant Robots: Comparative Assessment and Two Enhancements. *IEEE/ASME Transactions on Mechatronics* 20(2):944-955.

[5] Menasri R, Nakib A, Daachi B, Oulhadj H, Siarry P (2015) A Trajectory Planning of Redundant Manipulators Based on Bilevel Optimization. *Applied Mathematics and Computation*, 250:934-947.

[6] Erdős G, Kardos Cs, Kemény Zs, Kovács A, Váncza J (2016) Process Planning and Offline Programming for Robotic Remote Laser Welding Systems. *Int. Journal of Computer Integrated Manufacturing*, in print.

[7] Reinhart G, Munzert U, Vogl W (2008) A Programming System for Robot-Based Remote-Laser-Welding with Conventional Optics. *CIRP Annals—Manufacturing Technology* 57(1):37-40.

[8] Kovács A (2016) Integrated Task Sequencing and Path Planning for Robotic Remote Laser Welding. *Int. Journal of Production Research*, in print.

[9] Ceglarek D, Colledani M, Váncza J, Kim D-Y, Marine C, Kogel-Hollacher M, Mistry A, Bolognese L (2015) Rapid Deployment of Remote Laser Welding Process in Automotive Assembly Systems. *CIRP Annals—Manufacturing Technology* 64(1): 389-394.

[10] Pellegrinelli S, Pedrocchi N, Tosatti LM, Fischer A, Tolio T (2014) Multi-robot Spot-Welding Cells: An Integrated Approach to Cell Design and Motion Planning. *CIRP Annals—Manufacturing Technology* 63(1):17-20.

[11] Altintas Y, Erkorkmaz K (2003) Feedrate Optimization for Spline Interpolation in High Speed Machine Tools. *CIRP Annals—Manufacturing Technology*, 52(1):297-302.

[12] Erkorkmaz K, Layegh SE, Lazoglu I, Erdim H (2013) Feedrate Optimization for Freeform Milling Considering Constraints from the Feed Drive System and Process Mechanics. *CIRP Annals—Manufacturing Technology* 62(1):395-398.

[13] Sencer B, Ishizaki K, Shamoto E (2015). High Speed Cornering Strategy with Confined Contour Error and Vibration Suppression for CNC Machine Tools. *CIRP Annals—Manufacturing Technology* 64(1):369-372.

[14] Burdick JW (1989). On the Inverse Kinematics of Redundant Manipulators: Characterization of the Self-Motion Manifolds. *IEEE International Conference on Robotics and Automation*, pp. 264-270.