Manipulator Motion Planning in Redundant Robotic System for Fiber Placement Process

Jiuchun Gao, Anatol Pashkevich, Stéphane Caro

HAL Id: hal-01692533
https://hal.science/hal-01692533

Submitted on 4 May 2018

Manipulator Motion Planning in Redundant Robotic System for Fiber Placement Process

Jiuchun Gao, Anatol Pashkevich and Stéphane Caro

Abstract This paper proposes a new methodology for motion planning in redundant robotic system for the high-speed fiber placement technology. The considered system is composed of a 6-axis industrial robot and a one-axis actuated positioner. Compared to the previous works assuming constant rate, the particularity of this work lies in the time-optimal motion coordination based on the combinatorial optimization and the polynomial approximation techniques. Advantages of the developed technique are confirmed by an application example that deals with a planar robotic system for fiber placement.

Keywords Redundant robotic system · Motion planning · Dynamic programming · Fiber placement

1 Introduction

Composites have been increasingly used in aerospace and automotive industries because of their good strength-to-weight ratio and durability [1]. For fabricating complex composite parts, fiber placement is an important technique [2]. The fiber placement process can be implemented by using either specifically designed machines or robotic systems, which are redundant in this application. The machines have no limitations on the component size, but they usually require large work floor areas and are expensive [3]. Compared to the process-dedicated machines, the

J. Gao (✉) · A. Pashkevich
Ecole des Mines de Nantes, Nantes, France
e-mail: Jiuchun.Gao@irccyn.ec-nantes.fr

A. Pashkevich
e-mail: Anatol.Pashkevich@mines-nantes.fr

S. Caro
IRCCyN, CNRS, Paris, France
e-mail: Stephane.Caro@irccyn.ec-nantes.fr

robotic system (usually includes a 6-axis robot and a one-axis positioner) is relatively cheap and flexible allowing changing the product type easily.

In robotic fiber placement, planning of manipulator motions is an important issue. The main difficulty here arises due to the redundancy of the robotic system with respect to the manufacturing task. In literature, there are a number of works that deal with the redundancy resolution. Relevant techniques are usually based on the pseudo inverse of the kinematic Jacobian. However, they can be hardly applied to the considered problem because they do not allow generating optimal trajectories satisfying real-life industrial requirements [4]. There are a few researches dealing with this problem in particular technological process. For laser-cutting and arc-welding applications, relevant numerical techniques were proposed in [5], but the main assumption (constant speed of the tool) is not in agreement with the nature of the fiber placement. For fiber placement process, the motion planning problem was studied by Martinec et al. [6] and Mlynek and Martinec [7], who also assumed the end-effector velocity is constant. For this reason, the problem of optimal manipulator motion planning for the fiber placement has not been solved completely yet.

This paper proposes a new methodology that aims at obtaining the time-optimal trajectory taking into account real industrial constraints. The proposed approach is based on the discretization of the redundant variable and dynamic programming technique. Further, a smoothing technique is also applied.

2 Robotic Fiber Placement Problem

A typical robotic fiber placement system (a 6-axis robot and a one-axis positioner) is presented in Fig. 1. The workpiece is mounted on the positioner that is able to change its orientation in order to improve accessibility of certain desired zones by the robot end-effector. The latter is attached to the manipulator flange and ensures placement of the fiber tows in the desired locations.
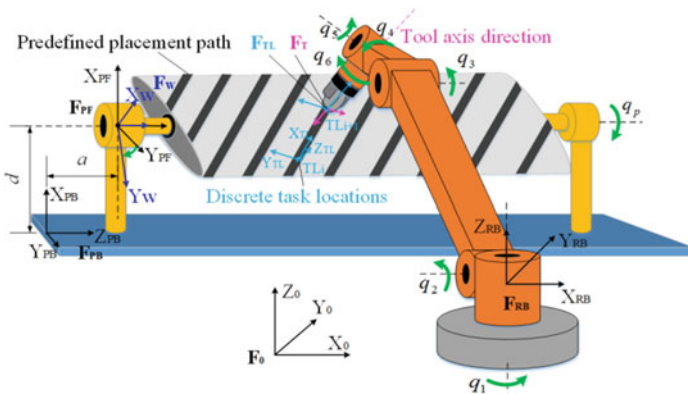

Fig. 1 Robotic fiber placement system

To describe the fiber placement task, let us introduce several coordinate systems attached to the robotic manipulator, the positioner, the workpiece and the robot end-effector (see Fig. 1). It is assumed that the fiber placement task is defined with respect to the frame F_W attached to the workpiece. The task is presented as a set of n discrete task locations (position and orientation) sampled on the given placement path. For each of these locations, it is also defined a frame F_{TLi}, $i = 1, 2, \ldots n$, in such a way that the axis X_{TLi} is directed along the path direction and the axis Z_{TLi} is normal to the workpiece surface pointing outside of it. Using these notations, the fiber placement task can be presented as a sequence of homogenous 4×4 matrices describing the mapping from the frame F_W to F_{TLi}

$$^{W}\mathbf{T}_{TLi} = \begin{pmatrix} \mathbf{n}_i & \mathbf{s}_i & \mathbf{a}_i & \mathbf{p}_i \\ 0 & 0 & 0 & 1 \end{pmatrix}_{4 \times 4} = \begin{pmatrix} \mathbf{R}(\boldsymbol{\varphi}_i) & \mathbf{p}_i \\ 0\ 0\ 0 & 1 \end{pmatrix}; \ i = 1, 2, \ldots, n \qquad (1)$$

where $\mathbf{p}_i = (x_i, y_i, z_i)^T$ is the position vector of the frame F_{TL_i} origin, $\boldsymbol{\varphi}_i = (\alpha_i, \beta_i, \gamma_i)^T$ is the orientation vector in accordance with the *Z-Y-X* Euler angles definition, and $\mathbf{n}_i$, $\mathbf{s}_i$ and $\mathbf{a}_i$ are the unit vectors giving directions of the axes *X*, *Y*, and *Z*. To execute the given fiber placement task, the robot end-effector must visit the frames (1) sequentially and as fast as possible.

3   Robotic System Model

The robotic system started in this paper is composed of two components: an industrial robot and a positioner. The positioner model describes the mapping from the positioner base to the positioner flange that depends on the positioner joint coordinate q_p. This transformation can also be presented as a 4×4 homogenous transformation matrix $pos(q_p) = \mathbf{R}_X(q_p)$. Further, the matrix function $pos(q_p)$ will be referred to as the positioner direct kinematics that depends on the positioner parameters a and d (see Fig. 1). The robot model describes the mapping from the robot base to the robot flange. For serial manipulators, it can be obtained using the *D-H* technique and presented as the product of the 4×4 homogenous matrices

$$rob(\mathbf{q}_r) = {}^{RB}\mathbf{T}_1(q_1) \cdot {}^{1}\mathbf{T}_2(q_2) \cdot {}^{2}\mathbf{T}_3(q_3) \cdot {}^{3}\mathbf{T}_4(q_4) \cdot {}^{4}\mathbf{T}_5(q_5) \cdot {}^{5}\mathbf{T}_6(q_6) \qquad (2)$$

where the matrix function $rob(\mathbf{q}_r)$ is also called the robot direct kinematics; and $\mathbf{q}_r$ is the vector of the robot joint coordinates.

To derive the designed model of the robotic system, let us present the task frames in two ways, using the robot and positioner kinematics as follows

$$^{0}\mathbf{T}_{TLi}(\mathbf{q}_r) = {}^{0}\mathbf{T}_{RB} \cdot rob(\mathbf{q}_r) \cdot {}^{RF}\mathbf{T}_T \cdot {}^{T}\mathbf{T}_{TLi} \qquad (3)$$

$$^0\mathbf{T}_{TLi}(q_p) = {}^0\mathbf{T}_{PB} \cdot pos(q_p) \cdot {}^{PF}\mathbf{T}_W \cdot {}^W\mathbf{T}_{TLi} \tag{4}$$

where $\mathbf{T}$ is the matrix which defines the mapping from the frame referred to the left superscript to the frame referred to the right subscript.

After equating these two presentations, the robotic model can be expressed as

$$^0\mathbf{T}_{RB} \cdot rob(\mathbf{q}_r) \cdot {}^{RF}\mathbf{T}_T \cdot {}^T\mathbf{T}_{TLi} = {}^0\mathbf{T}_{PB} \cdot pos(q_p) \cdot {}^{PF}\mathbf{T}_W \cdot {}^W\mathbf{T}_{TLi} \tag{5}$$

It should be stressed that this model describes the geometry of a closed kine-matic loop whose configuration depends on the joint coordinates $\mathbf{q}_r$ and q_p of the robotic manipulator and the positioner, respectively. Besides, it should be men-tioned that this model is redundant with respect to the joint variables since it gives six independent scalar equations for seven unknowns. The latter does not allow the users to obtain the unique solution for the robot and positioner motions corre-sponding to the given fiber placement path. On the other hand, it provides some rooms for the optimization of robot /positioner motions when tracking a specified curve.

4 Optimization Problem

To utilize the redundancy in the best way, it is reasonable to partition the required movement between the robot and the positioner ensuring that the end-effector executes the desired task with smooth motion in the shortest time.

4.1 Motion Generation Problem

To present the above mentioned problem in a more formal way, let us introduce the functions $\mathbf{q}_r(t)$ and $q_p(t)$ that describe the robot and positioner motions on the time interval $t \in [0, T]$. In addition, let us define the time instances $\{t_1, t_2, \dots t_n\}$ corresponding to the cases where the robot end-effector visits the task nodes $\{F_{TLi} \,|\, i = 1, 2, \dots n\}$ defined by Eq. (1), where $t_1 = 0$, $t_n = T$. Using this notation, the problem can be presented as minimization of the total travelling time

$$T \rightarrow \min_{\mathbf{q}_r(t),\; q_p(t)} \tag{6}$$

over $\mathbf{q}_r(t)$ and $q_p(t)$ subjected to the equality/inequality constraints, which are imposed by the prescribed task

$$^0\mathbf{T}_{RB} \cdot rob(\mathbf{q}_r(t_i)) \cdot {}^{RF}\mathbf{T}_T \cdot {}^T\mathbf{T}_{TLi} = {}^0\mathbf{T}_{PB} \cdot pos(q_p(t_i)) \cdot {}^{PF}\mathbf{T}_W \cdot {}^W\mathbf{T}_{TLi}; \quad i = 1, \, 2, \, \ldots n$$

(7)

and which describe the capacities of the robot/positioner actuators

$$q_j^{\min} \leq q_j(t_i) \leq q_j^{\max} \tag{8a}$$

$$\dot{q}_j^{\min} \leq \dot{q}_j(t_i) \leq \dot{q}_j^{\max} \tag{8b}$$

$$\ddot{q}_j^{\min} \leq \ddot{q}_j(t_i) \leq \ddot{q}_j^{\max} \tag{8c}$$

where $j = 0, \, 1, \, \ldots 6$ is the common index for the joint variables, and $j = 0$ corresponds to q_p. Besides, the collision constraints are defined as follows:

$$cols \, (q_p(t), \, \mathbf{q}_r(t)) = 0; \quad \forall t \in [0, \, T] \tag{9}$$

must be taken into account, where the binary function $cols \, (.)$ verifies the intersections between the system components (robot and positioner links, fixture, etc.).

For this optimization problem, which aims at finding continuous functions $\mathbf{q}_r(t)$ and $q_p(t)$, there is no standard technique that can be straightforwardly applied to. The main difficulty here is related to the equality constraints that are written for the unknown time instances $t_1, \, t_2, \, \ldots t_n$. Besides, this problem is highly nonlinear and includes redundant variables (for the non-redundant case without the collision constraints the problem was solved by Bobrow et al. [8]). For these reasons, this paper proposes a discrete optimization based approach that is able to take into account both the system redundancy and the collision constraints.

4.2   Search Space Discretization

For the considered robotic system, there is a single redundant variable with respect to the task. It is convenient here to consider q_p as the redundant one. This allows us, after solving the positioner direct kinematics for any given q_p, straightforwardly apply the manipulator inverse kinematics and find corresponding configurations $\mathbf{q}_r$ of the robot. This approach permits us to take into account explicitly the equality constraint in Eq. (7).

To present the problem in a discrete way, let us sample the allowable domain of the redundant variable $q_p \in [q_p^{\min}, \, q_p^{\max}]$ with the step Δq_p

$$q_p^{(k)} = q_p^{\min} + \Delta q_p \cdot k; \quad k = 0, \, 1, \, \ldots m \tag{10}$$

where $m = (q_p^{\max} - q_p^{\min})/\Delta q_p$. Then, applying sequentially the positioner direct kinematics and the manipulator inverse kinematics in accordance with Eq. (7), one can get a set of possible configuration states for the robotic system. For the unique mapping from the task space to joint space, let us define the configuration index vector $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)^T$ that corresponds to the manipulator posture, where μ_1, μ_2 and μ_3 specify the shoulder, elbow and wrist indices respectively. Then, the manipulator configuration corresponding to the given $q_p^{(k)}$ can be obtained as follows:

$$\mathbf{q}_r^{(k)}(t_i) = g_r^{-1}(g_p(q_p^{(k)}(t_i)), \boldsymbol{\mu}); \ k = 0, 1, \ldots m; \ i = 1, 2, \ldots n \qquad (11)$$

where t_i specifies the unknown time instant corresponding to the task point TL_i, the functions $g_p(.)$ and $g_r^{-1}(.)$ denote the positioner direct kinematics and the manipulator inverse kinematics, respectively. Therefore, for each task location we can generate a number of configuration states, i.e. $TL_i \rightarrow \mathbf{L}_c^{(k,i)}; \ \forall k, i$, where $\mathbf{L}_c^{(k,i)} = (\mathbf{q}_r^{(k)}(t_i), q_p^{(k)}(t_i))$ will be further referred to as the location cell.

While considering the task locations are ordered in time, the original sequence of TL_i described by Eq. (1) may be converted into a directed graph presented in Fig. 2. It should be noted that some of the configurations generated by Eq. (11) should be excluded from the graph because of violation of the collision constraints (9) or the actuator joint limits (8a). Besides, from an engineering point of view, it is prudent to avoid some configurations that are close to the manipulator singular postures.
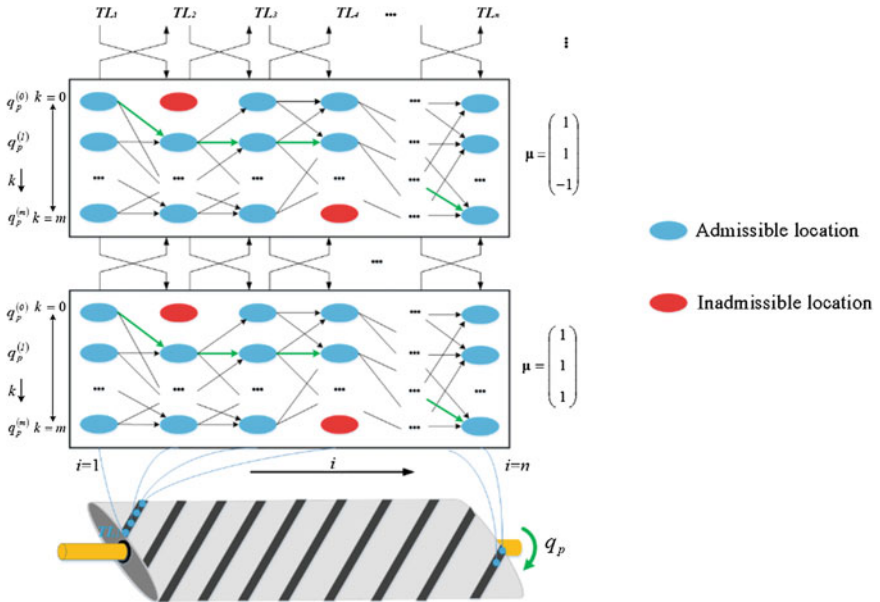

Fig. 2 Graph-based presentation of the discrete search space

These cases correspond to the "inadmissible" in Fig. 2, which are not connected to any neighbor. It is clear that due to time-irreversibility, the allowable connections between the graph nodes are limited to the subsequent configuration states $\mathbf{L}_c^{(k_i, i)} \rightarrow \mathbf{L}_c^{(k_{i+1}, i+1)}$, and the edge weights correspond to the minimum travelling time that are restricted by the actuator capacities expressed by Eq. (8a–8c).

Using the discrete search space, the considered problem can be transformed to the searching of the shortest path on the above presented graph. In accordance with the physical sense, the initial and final nodes must belong to the sets $\{\mathbf{L}_c^{(k_1, 1)}, \forall k_1\}$ and $\{\mathbf{L}_c^{(k_n, n)}, \forall k_n\}$ respectively. In the frame of this notation, the desired solution can be represented as the sequence $\{\mathbf{L}_c^{(k_1, 1)}\} \rightarrow \{\mathbf{L}_c^{(k_2, 2)}\} \rightarrow \dots \{\mathbf{L}_c^{(k_n, n)}\}$.

In accordance with the actuator constraints (8b), the distance between subsequent nodes can be evaluated as the displacement time for the slowest joint

$$dist\left(\mathbf{L}_c^{(k_i, i)}, \mathbf{L}_c^{(k_{i+1}, i+1)}\right) = \max_{j=0,..6} \left(\left|q_{j,i}^{(k_i)} - q_{j,i+1}^{(k_{i+1})}\right| / \dot{q}_j^{\max}\right) \tag{12}$$

The latter allows us to present the objective function (travelling time) as follows

$$T = \sum_{i=1}^{n-1} dist\left(\mathbf{L}_c^{(k_i, i)}, \mathbf{L}_c^{(k_{i+1}, i+1)}\right) \tag{13}$$

that depends on the indices $k_1, k_2, \dots k_n$. It should be noted that the applied method of edge weights computing automatically takes into account the velocity constraints (8b), but the acceleration constraints (8c) must be examined as follows

$$2\left|\Delta t_i(q_{j,i+1}^{(k_{i+1})} - q_{j,i}^{(k_i)}) - \Delta t_{i+1}(q_{j,i}^{(k_i)} - q_{j,i-1}^{(k_{i-1})})\right| / \Delta t_{i+1}\Delta t_i(\Delta t_{i+1} + \Delta t_i) \leq \ddot{q}_j^{\max} \tag{14}$$

which is based on the second order approximation of the corresponding functions $\mathbf{q}_r(t)$ and $q_p(t)$. Here, $\Delta t_{i+1} = dist\left(L_c^{(k_i, i)}, L_c^{(k_{i+1}, i+1)}\right)$ and $\Delta t_i = dist\left(L_c^{(k_i, i)}, L_c^{(k_{i-1}, i-1)}\right)$.

5 Path Planning Algorithm

After discretization, the original optimization problem (6)–(9) is converted to the combinatorial one, which can generally be transformed to the classical shortest path search on the graph (by simply adding virtual nodes at the beginning and the end). However, this straightforward approach is extremely time-consuming. For instance, it takes more than 20 h to find desired solution for 100 task locations and the positioner discretization step 1° (Intel® i5 2.67 GHz). Besides, standard techniques are not able to respect acceleration constraints (8c) that are very important here. For these reasons, a more efficient optimization algorithm was proposed that takes

into account particularities of the graph describing the search space, which can be also treated as a multi-layer network (see Fig. 2).

The developed algorithm is based on the dynamic programming principle, which breaks down the full-size problem into a set of sub-problems [9], aiming at finding the shortest path from the initial node set $\{\mathbf{L}_c^{(k_1, 1)}, \forall k_1\}$ to the current $\{\mathbf{L}_c^{(k_i, i)}, \forall k_i\}$. To present the basic idea, let us denote $d_{k,i}$ as the length of the shortest path connecting one of the initial nodes $\{\mathbf{L}_c^{(k_1, 1)}, \forall k_1\}$ to the current node $\{\mathbf{L}_c^{(k', i)}\}$. Then, taking into account the additivity of the objective (13), the shortest path for the nodes belonging to the next layer $\{\mathbf{L}_c^{(k, i+1)}, \forall k\}$ can be found by combining the optimal solutions for the previous layer $\{\mathbf{L}_c^{(k', i)}, \forall k'\}$ and the distances between the nodes with the indices i and $i+1$. The latter corresponds to the formula

$$d_{k,i+1} = \min_{k'}\left\{d_{k',i} + dist\left(\mathbf{L}_c^{(k,i+1)}, \mathbf{L}_c^{(k',i)}\right)\right\}; \qquad (15)$$

that is applied sequentially starting from the second layer, i.e. $i = 1, 2, \ldots n - 1$. Finally, after selection of the minimum length $d_{k,i+1}$ corresponding to the final layer and applying the backtracking, one can get the desired optimal path. Therefore, the desired solution for n sets $\{\mathbf{L}_c^{(k_1, 1)}\} \rightarrow \{\mathbf{L}_c^{(k_2, 2)}\} \rightarrow \ldots \{\mathbf{L}_c^{(k_n, n)}\}$ is obtained. The desired path is described by the recorded indices $\{k_1, k_2, \ldots k_n\}$.

The developed algorithm is rather time-efficient and perfectly suits industrial requirements. In particular, it takes about 30 s to find the optimal solution for the above mentioned problem (100 task locations, discretization step 1°).

6 Application Example

To show the efficiency of the proposed technique, let us apply it to a fiber placement system (a two-axis robot and a one-axis positioner). This system possesses a 1-dof redundancy with respect to the task. The geometrical parameters of the system and the actuator constraints are given in Fig. 3. The number of task locations is equal to $n = 100$ and the discretization step is equal to $\Delta q_p = 0.5°$.

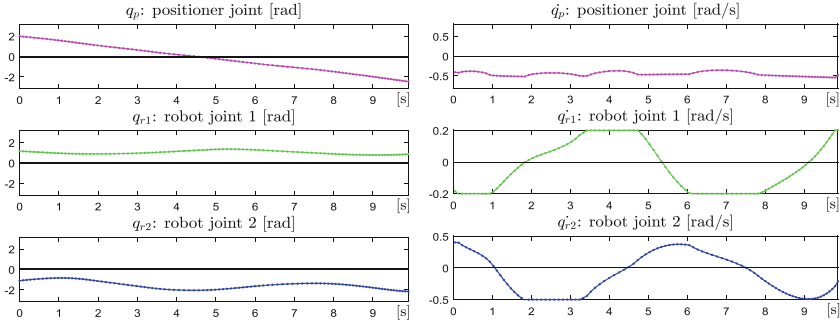**Fig. 3** Fiber placement system: two-axis robot and one-axis positioner
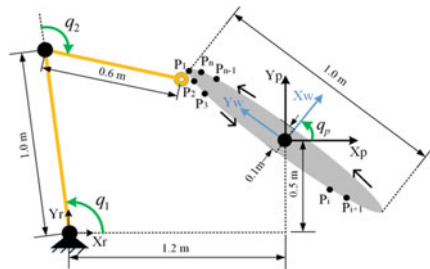
Fig. 4 Optimal trajectories for the robotic fiber placement system

A desired solution was obtained by implementing the developed algorithm described in Sect. 4. The result is presented in Fig. 4, which contains the displacement and velocity profiles for all three actuators. It should be noted that the trajectories obtained by direct application of the dynamic programming technique include oscillations in derivatives, which are undesirable from engineering point of view. For this reason, a specially developed smoothing technique was applied to improve locally the velocity and acceleration profiles. This technique is based on the polynomial approximation of the redundant variable profile and relevant correction of the displacement intervals, to ensure that the velocities and accelerations are within their limits. It is clear that this modification slightly increases the travelling time, but it yields important benefits to the actuator working conditions.

7 Conclusions

This paper contributes to a new methodology to optimize the manipulator motion in redundant robotic system for the fiber placement process. In contrast to the previous works, the proposed approach generates the time-optimal smooth trajectories and allows using the full capacities of the robotic system expressed as the velocity, acceleration and collision constraints. The developed algorithm is based on dynamic programming technique, and leads to significant computing time reduction compared to the standard shortest-path algorithms. Advantages of the developed technique are highlighted with an application example that deals with a planar robotic system for fiber placement. Future research works will concentrate on its integration into industrial robot control system software.

References

1. Kaw, A.K.: Mechanics of Composite Materials. CRC Press (2005)
2. Gay, D.: Composite Materials: Design and Applications. CRC Press (2014)
3. Gallet-Hamlyn, C.: Multiple-use robots for composite part manufacturing. JEC Compos. **62**, 28–30 (2011)
4. Kazerounian, K., Nedungadi, A.: Redundancy resolution of serial manipulators based on robot dynamics. Mech. Mach. Theory **23**(4), 295–303 (1988)
5. Dolgui, A., Pashkevich, A.: Manipulator motion planning for high-speed robotic laser cutting. Int. J. Prod. Res. **47**(20), 5691–5715 (2009)
6. Martinec, T., Mlýnek, J., Petrů, M.: Calculation of the robot trajectory for the optimum directional orientation of fibre placement in the manufacture of composite profile frames. Robot. Computer-Integrated Manuf. **35**, 42–54 (2015)
7. Mlynek, J., Martinec, T.: Mathematical model of composite manufacture and calculation of robot trajectory. In: 2014 16th International Conference on. IEEE Mechatronics-Mechatronika (ME), (2014)
8. Bobrow, J.E., Dubowsky, S., Gibson, J.: Time-optimal control of robotic manipulators along specified paths. Int. J. Robot. Res. **4**(3), 3–17 (1985)
9. Bertsekas, D.P., et al.: Dynamic Programming and Optimal Control, vol. 1, No. 2. Athena Scientific, Belmont, MA (1995)