TRAJECTORY OPTIMIZATION OF A SPOT-WELDING ROBOT IN THE JOINT AND CARTESIAN SPACES

Ehsan Sharafian Moghaddam*,** Maryam Ghassabzadeh Saryazdi*,** and Afshin Taghvaeipour*

Abstract

This paper targets the trajectory planning of a serial robot which is used in the spot-welding process of an automobile industry. The problem is first defined based on a single-objective optimization algorithm in which the minimization of travelling time is aimed. Next, a multi-objective optimization problem is defined where the minimization of both the travelling time and energy consumption is demanded. In the proposed methodology, trajectory planning is conducted by considering the position and orientation of the welding gun simultaneously. The corresponding optimization problems are approached by incorporating heuristic methods such as genetic algorithm and particle swarm optimization, and in both the joint and Cartesian spaces. In this regard, the trajectory planning of a real welding robot by considering the most effective parameters is conducted. First, the best space and solution algorithm are identified on the part of the trajectory, and then, the obtained conclusions are used for the remained parts. Results reveal that the obtained trajectories just for a case study robot in the production line can reduce the travelling time of the spot-welding operation up to 24% for the minimum time trajectory and up to 11% for the optimized multi-objective trajectory compared with the current performed trajectory, which is determined manually and non-computationally by human operators.

Key Words

Spot-welding robot; trajectory planning; travelling time; energy consumption; multi-objective optimization; automobile industry

1. Introduction

These days, robots play a critical role in manufacturing and production processes. However, excessive usage of energy and the corresponding environmental impacts lead researchers and engineers to search for optimum design and operation solutions. European Union commission of energy ordered industries to reduce energy consumption in 2020 by 20% comparing with what was consumed in 2007 [1]. In addition to the energy consumption, to elevate the efficiency of a production line, the minimization of travelling time is commonly demanded by engineers. In the best scenario, researchers have been looking for a trajectory, which minimizes both the travelling time and energy consumption simultaneously.

Generally, industrial robots are categorized into two different types, serial and parallel. Serial robots are made by concatenation of links and joints, while parallel counterparts are composed of two or more serial chains, which connect the end-effector (EE) to a fixed base platform. Repetitive tasks such as pick-and-place operations, palletizing, colour spraying, and welding normally call for serial robots. In such tasks, an EE should follow a spatial path with a certain velocity and acceleration, so called a trajectory. Besides respecting essential kinematic and dynamic constraints such as joints motion limit, obstacle avoidance, and actuators limit, it is also demanded to optimize one or more objective functions such as the travelling time, the energy consumption, and the maximum jerk. In recent years, many researchers have focused on robot trajectory planning by resorting to different objectives and optimization algorithms [2]. Many of them have considered energy consumption as a priority and tried to optimize the trajectory of industrial robots [1, 3, 4, 5, 6]. Boscariol and Richiedei [7] defined an optimization problem with spline interpolation methods to obtain energy-efficient trajectories for Cartesian robots. The authors used polynomial functions to produce a trajectory passing through a sequence of *via*-points; however, the proposed method is only suitable for the Cartesian robot with uncoupled dynamics. While energy consumption minimization has been widely targeted as the main reason for trajectory optimization, some researchers targeted the minimization of travelling time in a specific task [8, 9, 10, 11]. Ghasemi [12] *et al.* presented a method for computing time-optimal trajectories for a serial planner manipulator by considering actuator and jerk limitations, and the same goal in addition to collision avoidance constraint was investigated by Diao [13] for a grinding robot as a spatial manipulator. These

* Department of Mechanical Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran; E-mails: esharafianm@aut.ac.ir; mghsaryazdi@aut.ac.ir; atagh vaei@aut.ac.ir

** Vehicle Technology Research Institute, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran. Maryam Ghassabzadeh Saryazdi.
Corresponding author: Maryam Ghassabzadeh Saryazdi

Recommend by Chaomin Luo
(DOI: 10.2316/J.2023.206-0701)

two papers investigated the collision avoidance trajectory in a simple workspace which is not compatible with an actual workspace in industry. Abu-Dakka *et al.* [14] incorporated an evolutionary algorithm to obtain minimum time trajectories in the joint space for a three joints robot that can be generalized to six joints considering the orientation of tool as well. Instead of dealing with single-criterion optimization problems, the robot trajectory can be better planned by resorting to a multi-criterion or multiple objective functions. In this vein, Wu [15] considered time-energy-jerk as the objective function and obtained the candidate cases of the optimal trajectories for a serial robot. Liu *et al.* [16] and Huang *et al.* [17] coped with a trajectory optimization problem to minimize both the travelling time and maximum jerk. Saravanan *et al.* [18] focused on an optimization problem in which a trajectory was designed to minimize a multi-criterion objective function while the joint limits and payload constraints were respected. Implementing the constraints such as collision avoidance has received less attention in these researches. The objective function is in fact a weighted scalar function which is composed of the travelling time, the total energy, joint jerks, joint accelerations, and manipulability. The authors further enriched their study by coping with a multi-objective optimization problem in which the travelling time and energy consumption of an industrial robot were minimized while some obstacles were avoided [19]. Both studies have incorporated evolutionary algorithms to deal with optimization problems. In the foregoing studies, reaching some defined target points was only considered; however, the orientation of EE, while reaching a target point, is also important in some applications [20], [21].

The automotive industry is the one that benefits the serial robots for various tasks such as arc and spot-welding procedures. Currently, the welding robots are serial robots with a welding gun as an EE, which is traditionally planned manually to perform welding tasks. However, manual planning cannot be expected to meet the optimal goals. With the recent advancement of computational methods, welding path planning has been efficiently conducted by defining proper optimization problems [22, 21, 22, 23, 24, 25]. In a recent study, Wang *et al.* [26] targeted the path optimization of a welding robot to reach the minimum path length and energy consumption simultaneously, while the welding gun (spot-welding tool connected to the end of the robot) is avoided to hit obstacles. This research investigated the trajectory of the robot kinematically and geometrically. In other words, the dynamic equations and the related constraints were not considered in the model. Among heuristic optimization methodologies, particle swarm optimization (PSO) and genetic algorithm (GA) are often used by researchers for trajectory optimization problems. Beik [27] generally reviewed the welding optimization procedures and investigated the evolutionary algorithms like GA and PSO, which can be applied for robots in the automobile industry; however, the constraints, such as torque limitations and collision avoidances, were not included in the algorithm. Wang *et al.* [28] discussed the effectiveness of intelligent optimization strategies and targeted the travelling time minimization problem by means of an improved PSO algorithm; however, the dynamic parameters were not implemented in the equations.

In this study, the trajectory planning of a spot-welding robot is aimed while compared to the previous researches, methods to design both trajectories in joint space and Cartesian space will be proposed, and after assessing the results, the most suitable design space will be chosen. Furthermore, due to the significant size proportion of the gun's dimension to the robot's one, the corresponding orientation of the gun will be considered as an effective parameter in designing trajectories. In this regard, polynomial functions are used, and the optimal trajectory is obtained *via* single and multi-objective optimization problems while respecting real dynamic, kinematic, and obstacle avoidance constraints. In the end, the results of the optimal trajectories will be compared with the real values in the automobile company. It should be noted although the results of this investigation will be computed for a specified spot-welding robot in a production line, the proposed methods can be implemented readily for all types of serial robots in industries.

2. Trajectory Planning

To guide a robot, a proper path should be designed, which is, in fact, composed of a set of points between the initial and end positions. The EE can operate on a path with different velocities and accelerations. These points are the position of the EE in the Cartesian space, which can be mapped into the joint variables as well. In general, when the initial and end points and the constraints are specified, the path can be generated by resorting to some analytical functions.

Depending on a robot's task, the EE should be located in a specific position with a certain orientation to enter to and depart from the target points. For example, in a spot-welding robot, the positioning of the gun on the body of the vehicle is specified according to a welding process. Therefore, after the target points and the corresponding orientation of the EE are specified, an inverse kinematic analysis should be conducted to calculate the joint variables. In this regard, a matrix is needed to relate the EE position and orientation (with respect to the reference frame) to the joint variables, referred to as the *homogeneous target matrix* (HTM) [29, 30]. In a spot-welding process, each pair of adjacent target points can be considered initial and end positions for which a trajectory should be identified. The given target points are located on the body of a car, and the gun becomes fixed near the body. In a real welding process, the angle between the electrode and a body plate can vary up to 7° from the vertical axis; however, in this study, it is considered normal to the plate. Figure 1 represents the unit vectors along the gun and normal to the plate by vectors **a** and **r**, respectively.

The normal vector, **r**, can be obtained by the cross product of two vectors which are directed from the target point to two other points on the car's body. The selected points on the body should be close enough to the target point to approximately form a plate,

$$^{0}\mathbf{r} = {}^{0}\mathbf{r}_1 \times {}^{0}\mathbf{r}_2 \tag{1}$$

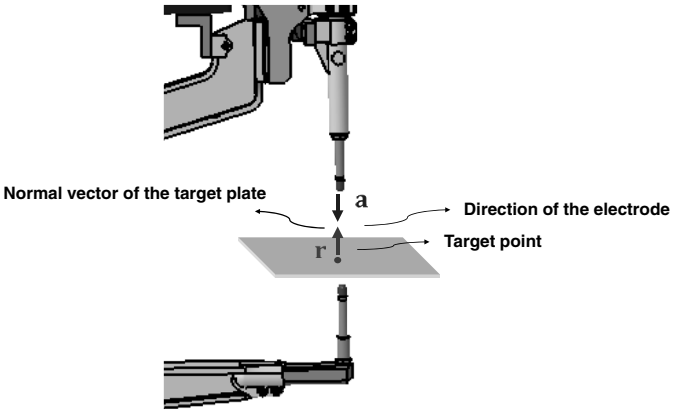

Figure 1. The orientation of the EE with respect to the plate of care body.

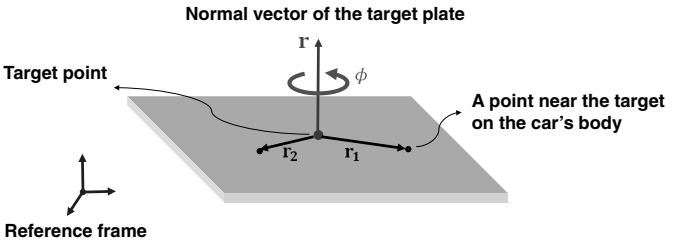

Figure 2. The normal vector of the target plate.

where ${}^0\mathbf{r}_1$and ${}^0\mathbf{r}_2$ are depicted in Fig. 2. Generally, the car's body comprises a curved geometry, and hence, a normal vector can be obtained by taking the average cross product vector of more than a pair of vectors. As it is depicted in Fig. 1, the electrode unit vector $\mathbf{a}$ is in the opposite direction of the obtained normal vector on a target point. It is apparent from Fig. 3 that the gun does not approach a welding point with a unique orientation as it can rotate around the axis of the electrode. This rotation angle can be denoted by ϕ, and the corresponding rotation matrix for an arbitrary vector $\mathbf{r} = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T$ is defined as [29],

$$\mathbf{R}_{\mathbf{r},\phi} = \begin{bmatrix} r_x^2 V\phi + C\phi & \pm r_x r_y V\phi - r_z S\phi & \pm r_x r_z V\phi + r_y S\phi \\ \pm r_x r_y V\phi + r_z S\phi & r_y^2 V\phi + C\phi & \pm r_y r_z V\phi - r_x S\phi \\ \pm r_x r_z V\phi - r_y S\phi & \pm r_y r_z V\phi + r_x S\phi & r_z^2 V\phi + C\phi \end{bmatrix} \tag{2}$$

$$V\phi = verse\,(\phi) = 1 - \cos(\phi)\, C\phi = \cos(\phi), S\phi = \sin(\phi)$$

The choice of the positive or negative sign in (2) depends on the type of EE arrival to a target point, which means arrival at the front side or the backside of the plate. As the position vector of a target point with respect to the reference frame, denoted by ${}^0\mathbf{p}_t$, is already known, the corresponding HTM can be readily obtained as,

$$\mathbf{T}_t = \begin{bmatrix} \mathbf{R}_{\mathbf{r},\phi} & {}^0\mathbf{p}_t \\ \mathbf{0} & 1 \end{bmatrix}_{4\times 4} \tag{3}$$

where $\mathbf{0}$ is a 1×3 zero matrix.

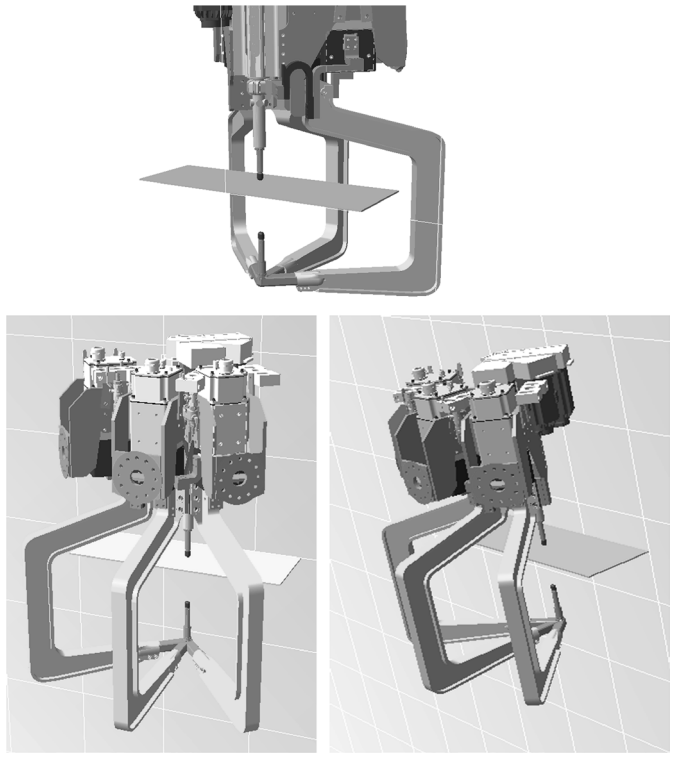

Figure 3. Rotation of the gun around the electrode axis.

2.1 Trajectory Planning in the Joint Space

To design a trajectory as a function of joint variables, first, the HTM of the EE for the initial and end target points should be specified. With a known HTM, it is possible to obtain up to eight solution path branches, and thus, it is necessary to choose a proper branch depending on criteria, which suit the robot's condition at the target points. For instance, the EE of a robot normally dwells when it departs from an initial point (I) and reaches to a final target point (F). Moreover, joints' jerk, which corresponds to $\dddot{\theta}$, is also required to be eliminated to prevent the impact at joints. These conditions can be written as below,

$$\theta_i^I(0) = \theta_i^I, \dot{\theta}_i^I(0) = 0, \quad \ddot{\theta}_i^I(0) = 0, \quad \dddot{\theta}_i^I(0) = 0 \tag{4a}$$

$$\theta_i^F(T_i) = \theta_i^F, \dot{\theta}_i^F(T_i) = 0, \quad \ddot{\theta}_i^F(T_i) = 0, \quad \dddot{\theta}_i^F(T_i) = 0 \tag{4b}$$

in which $i = 1, 2, \ldots, n$, and n is the number of joints. Here, it is assumed that, while the EE travels from the initial to target points, the travelling time of each joint can be different, denoted by T_i. Therefore, with these eight constraints on a joint variable and its derivatives, a 7^{th}-degree polynomial can be a good candidate, namely [31],

$$s_i(\xi_i) = a\xi_i^7 + b\xi_i^6 + c\xi_i^5 + d\xi_i^4 + e\xi_i^3 + f\xi_i^2 + g\xi_i + h \tag{5}$$

$$\xi_i = \frac{t}{T_i},\ 0 \le s_i \le 1,\ 0 \le \xi_i \le 1$$

in which t $(0 \le t \le \max\{T_1, T_2, \ldots, T_n\})$ denotes the travelling time, and ξ_i is the dimensionless time variable whose value is zero at the initial and one at the end target points. If a joint reaches its final position earlier than the

EE gets to the final target point, then the corresponding non-dimensionalized variable ξ is manually set to one.

Each joint variable, $\theta_i(t)$, can now be represented as,

$$\theta_i(t) = \theta_i^I + \left(\theta_i^F - \theta_i^I\right) s_i\left(\xi_{\mathrm{i}}\right) \quad , \quad i = 1,\ 2, \ldots, 6 \tag{6}$$

And for all joint variables, (6) can be cast in the following matrix format:

$$\theta(t) = \theta_I + (\theta_F - \theta_I)\, \mathbf{s}^T \tag{7}$$

where $\theta(t) = [\theta_1(t),\ \theta_2(t), \ldots, \theta_n(t)]^T$ is the vector of joint variables and $\mathbf{s} = [s_1(\xi_1), s_2(\xi_2), \ldots, s_n(\xi_n)]^T$.

The boundary conditions in (5) are also rewritten in terms of the non-dimensionalized time variable as it follows:

$$\mathbf{s}(\mathbf{0}) = \mathbf{0}, \mathbf{s}^{'}(\mathbf{0}) = \mathbf{0}, \mathbf{s}^{''}(\mathbf{0}) = \mathbf{0}, \quad \mathbf{s}^{'''}(\mathbf{0}) = \mathbf{0} \tag{8a}$$

$$\mathbf{s}(\mathbf{1}) = \mathbf{1}, \mathbf{s}^{'}(\mathbf{1}) = \mathbf{0}, \quad \mathbf{s}^{''}(\mathbf{1}) = \mathbf{0}, \mathbf{s}^{'''}(\mathbf{1}) = \mathbf{0} \tag{8b}$$

Therefore, a system of linear equations can be solved to obtain the unknown coefficients, and thus the 4-5-6-7 polynomial function can be obtained,

$$s_i(\xi_i) = -20\xi_i^7 + 70\xi_i^6 - 84\xi_i^5 + 35\xi_i^4 \tag{9}$$

The polynomial function and its derivatives are shown in Fig. 4. As it is apparent, the jerk is zero at the initial and end target points.

By resorting to the calculated 4-5-6-7 polynomial function in (9), the travelling time of the i^{th} joint can be calculated by identifying the maximum value of joint rate as below,

$$T_i = \frac{35\left(\theta_i^F - \theta_i^I\right)}{16\dot{\theta}_{i_{max}}} \tag{10}$$

The travelling time of the EE, T, is such that all joints reach their final states, which equals to the maximum value of the joints travelling time,

$$T = \max\{T_1, T_2, \ldots, T_n\} \tag{11}$$

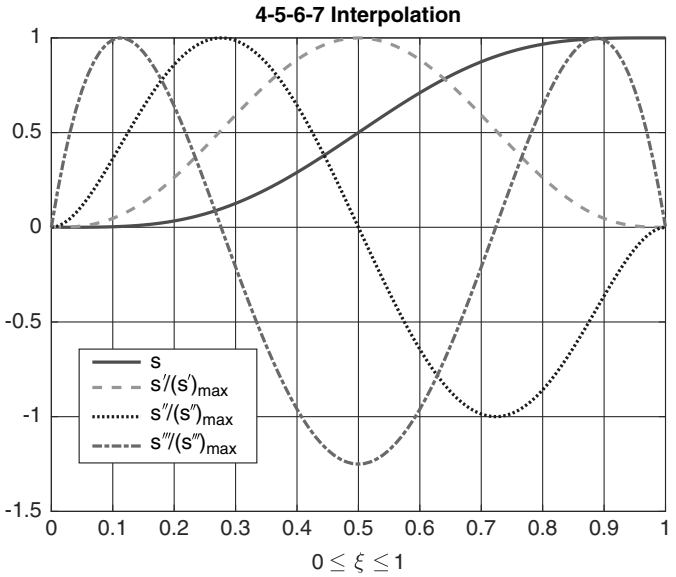

Figure 4. 4-5-6-7 polynomial function.

2.2 Trajectory Planning in the Cartesian Space

Traditionally, in automotive industries, a robot's trajectory is manually designed by incorporating a human operator who once locates the EE on the desired target points in the Cartesian space. The obtained trajectory is then recorded by the control system embedded in the robot. Besides a more precise understanding of the robot's motion, the aforementioned procedure is easier to guide the robot's EE to reach the target points. On the other hand, to computationally plan a trajectory in the Cartesian space, similar to what has been conducted in the joint space, a 7^{th}-degree polynomial function with constraint (8a) should be incorporated. In fact, in the foregoing methodology, instead of joint variables, the position vector of EE, $\mathbf{p}$, and the vector of Euler angles ($\mathbf{E} = \begin{bmatrix}\psi & \varphi & \theta\end{bmatrix}^T$ which represents the orientation of EE) are described by a 7^{th}-degree polynomial. In this regard, a six-dimensional array, denoted by χ, is introduced which comprises both position and orientation vectors, namely

$$\chi = \begin{bmatrix}\mathbf{p} \\ \mathbf{E}\end{bmatrix}_{6\times 1} = \begin{bmatrix}p_x & p_y & p_z & \psi & \varphi & \theta\end{bmatrix}^T \tag{12}$$

Trajectory planning in the Cartesian space is more computationally expensive because, at every posture, the EE's position and orientation vectors should be mapped into the joint space by means of the robot Jacobian matrix. In addition, at singular points, where the Jacobian matrix becomes rank deficient, it is not possible to derive the joint rates from a known posture of the EE. In such cases, during the procedure of trajectory planning, if the EE moves towards one of the singular points, the speed of one or more joints may theoretically attain infinite values. This condition often causes the arm to deviate from the desired path.

3. Constraints

In a trajectory planning procedure, generally, there are constraints between the initial and end target points, which should be respected by the robot. Important factors involved in a trajectory planning problem and types of the constraints are briefly shown in Fig. 5 [29].

For example, due to the geometrical and operational conditions of a robot, which can be normally obtained from the manufacturer catalogue, joint variables and its rates should be bounded between certain limits, namely

$$(\theta_i)_{Min} \leq\ \theta_i(t) \leq (\theta_i)_{Max} \tag{13a}$$

$$|\dot{\theta}_i(t)| \leq \left(\dot{\theta}_i\right)_{Max} \tag{13b}$$

$$|\ddot{\theta}_i(t)| \leq \left(\ddot{\theta}_i\right)_{Max}, \quad |\dddot{\theta}_i(t)| \leq (\dddot{\theta}_i)_{Max} \tag{13c}$$

where θ_i , $\dot{\theta}_i$, $\ddot{\theta}_i$, $\dddot{\theta}_i$ are the angle of joint, angular velocity, angular acceleration, and jerk, respectively. Also, the robot's servo motors cannot provide more than a certain amount of input torque (τ_i),

$$|\tau_i| \leq (\tau_i)_{Max} \tag{13d}$$

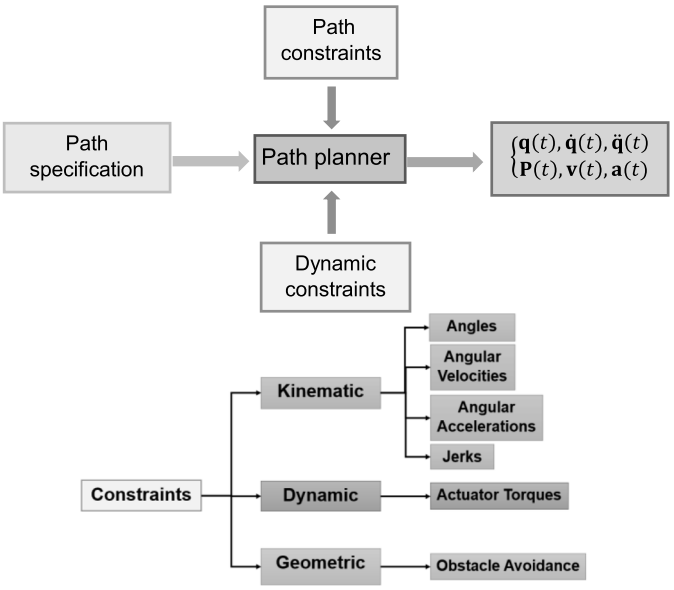

Figure 5 . The outlook of the trajectory planning problem and type of the constraints.

In addition to the above-mentioned constraints, in a trajectory planning procedure, there are also collision avoidance constraints, which are more challenging. In fact, a robot should not collide with other obstacles such as car body and production line equipment while operating. To cope with these constraints computationally, some points should be first distributed over the surface of the robot, referred to as a *cloud of points*. Next, the obstacles that should be avoided, such as the car body, links, or production line equipment, are replaced with parallelepiped geometries, which form the *red zones* or *forbidden zones*. To be more specific, the positions of the forbidden zone boundaries are obtained in the reference frame of the robot ($X_0Y_0X_0$ coordinate in Fig. 10). The procedure of obstacle avoidance constraint is proposed in more detail and is added to the text. There are some critical points on the car's surface, by connecting which the forbidden regions can be obtained. Not only do these regions indicate that the surface or boundaries, but they also provide the volumes that none of the robot's components can be there. To be more specific, considerable points on the robot's surface (including all of the links and the gun) cannot anytime pass through the boundaries of the forbidden zone. With this in mind, a chosen trajectory has a specific time travelling, which can be discretized in tiny step times. In each step, positions of all "clouds of points" are calculated with the aid of inverse and forward kinematic. Thus, in all the step times, the positions of the robot's components (which are variable during the steps) and the forbidden regions (which are constant all the time) are available. Consequently, it can be seen whether a point of the cloud is located in the forbidden regions or not; if it is true, it means that the colliding happened and the trajectory will be rejected. If it is not true, it means that the trajectory could pass their components to their destination without any collision, and it satisfied the obstacle avoidance constraint and will be proposed to the optimization algorithms to evaluate the cost functions.

It can be said that significant chosen points can represent the volumes of the robot and the car body as well. It is important to note that the collision constraints' accuracy will also enhance if the selected points increase. Additionally, decreasing the length of step times or, in other words, increasing the total step times can lead to having more confidence that the robot did not collide with the car's body. It can be claimed that the whole selected numbers of the robot's/car's volume and step times can guarantee whether the collision for any desired trajectory will happen or not. The length of step time for section 6.4 and the cases a to d (in the text) have been considered 0.001 s, due to

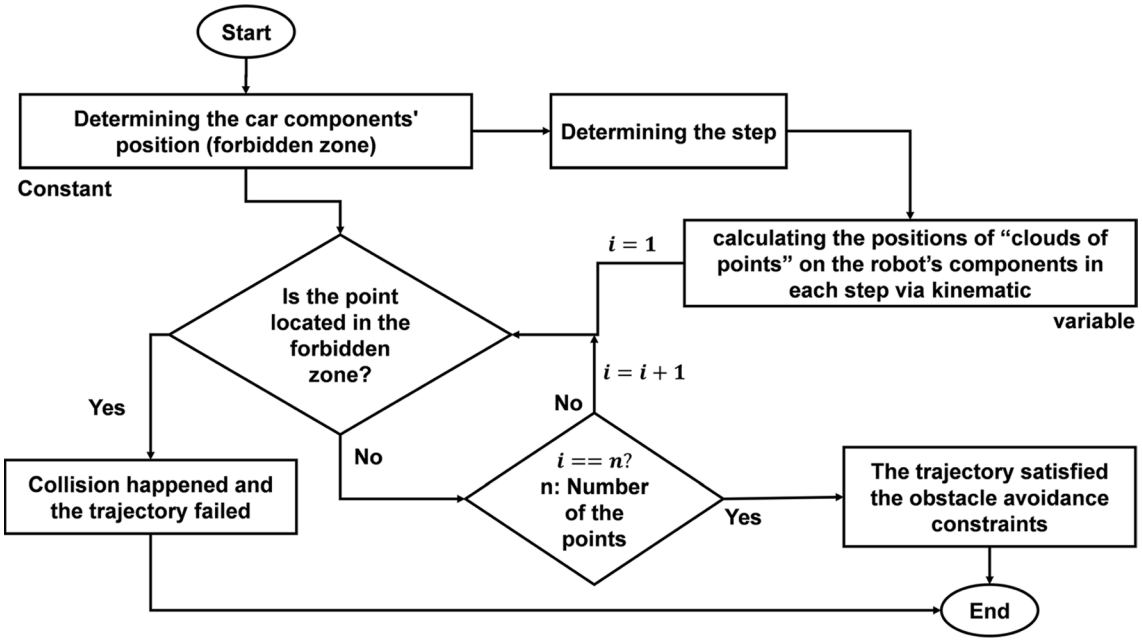

Figure 6 . The procedure of obstacle avoidance constraint in each step of time.

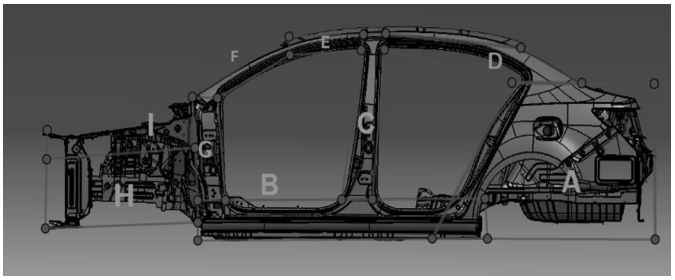

Figure 7. The red zone on the car's body.

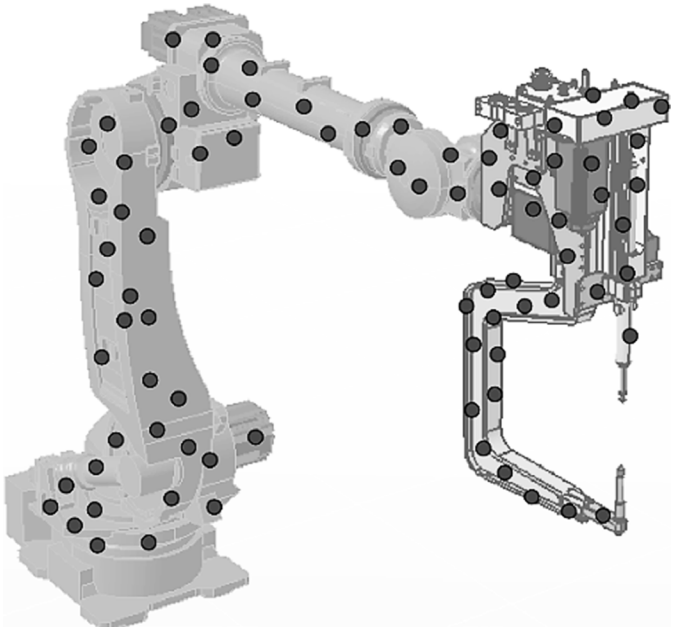

Figure 8. The cloud of points.

their short distance. However, for sections 6.5 and 6.6, the step has been chosen 0.01. On the other hand, the total number of clouds of points is 139, which can be claimed that they represent the robot's shape. To better delineate the procedure, the flowchart of the obstacle avoidance constraint for each step time is shown in Fig. 6. Figures 7 and 8 depict the red zones and the cloud of points of a spot-welding robot in a production line of the automotive industry, respectively, which is also the case study of the current research. As it is depicted in Fig. 8, in this study, the cloud contains 86 points on the gun and 53 points on the other links.

4. The Trajectory Optimization Problem

The ultimate goal in a trajectory optimization problem is to obtain a trajectory with which a robot can perform a task with a minimum travelling time and energy consumption while constraints are respected. However, the aforementioned optimization problem is very challenging. In this study, first, two distinct single-objective minimization problems are targeted, one with the travelling time and the other with energy consumption as objective functions. Then, a multi-objective optimization problem is defined

Table 1
The Vector of Design Variables for the Cartesian and Joint Spaces.

Design space	**Variables for optimization**
Joint space	$\left[\boldsymbol{\phi_{\mathbf{F}}}, \mathbf{b_F}, \dot{\theta}_{\mathbf{1_{max}}}, \ldots, \dot{\theta}_{\mathbf{n_{max}}}\right]$
Cartesian space	$[\boldsymbol{\phi_{\mathbf{F}}}, \mathbf{b_F}, \dot{\chi}_{\mathbf{1_{max}}}, \ldots, \dot{\chi}_{\mathbf{n_{max}}}]$

in which the minimization of both the travelling time and energy consumption is demanded.

To design a trajectory, it is needed to determine the initial and final posture of the robot. These postures can be readily obtained by resorting to given initial (p_i) and final (p_F) target points, an initial orientation of the gun (ϕ_I), and a path branch of the initial state (b_I). After that, the trajectory designing accomplished by choosing final orientation of the gun (ϕ_F), a path branch of the final state (b_F), and the maximum rate of the joints or EE velocities. The vector of design variables in the Cartesian and joint spaces is summarized in Table 1.

Here, the time cost function (C_t) is defined as the maximum time which takes for all variables, in the joint or the Cartesian space, θ or χ, respectively, to attain their final values, namely

$$C_t = \max\{T_i\}_{i=1,\ 2,3,\ \ldots,n} \tag{14}$$

in which T_i is obtained by resorting to (10) considering that the maximum rate for the i^{th} coordinate is known. Also, the energy cost function (C_e) is defined as the amount of energy which is consumed by the servo actuators from the initial to the end target points, namely

$$C_e = E = \sum_{i=1}^{6} \int_{t_I}^{t_F} |\tau(t)^T \dot{\theta}(t)| dt \tag{15}$$

where $\tau(t)$ is the vector of the actuator torques which is obtained from the dynamic equations of the robot [32]. The procedure of calculation $\tau(t)$ is explained in appendix A and verified in appendix B. In the next step, constraints should be introduced to complete the optimization problem. Among the constraints that were defined in the previous section, avoiding obstacles plays a crucial role in the minimization of the time cost function, and thus, it affects the final solution significantly. Finally, an optimization algorithm should be chosen based on the number of design variables and the nature of cost functions. Generally, in optimization problems with a large number of design variables and complex cost functions without closed-form expressions, heuristic algorithms are incorporated. However, these types of algorithms are time-consuming and prone to being trapped in local minima. In this regard, and to increase the reliability of solutions, two different heuristic algorithms, GA and PSO method, are used [33] and the obtained results will be compared. On the other side, satisfying the various constraints of the considered optimization problem plays a pivotal role in the performance of the algorithms. The computational cost of evaluating the constraints is significant; therefore, an algorithm should

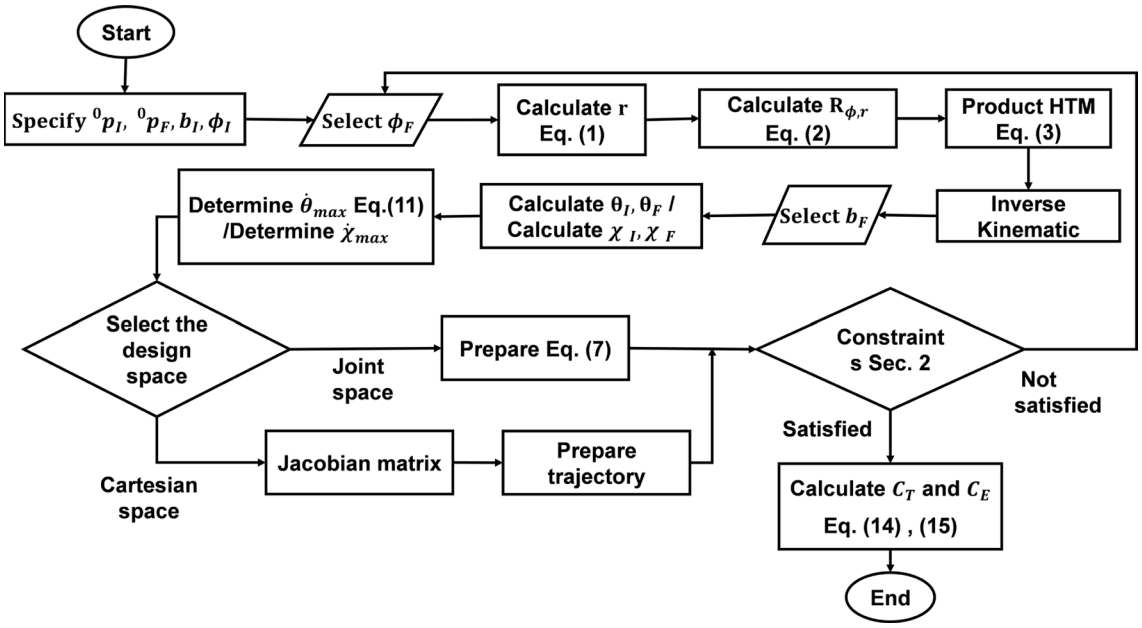

Figure 9. The algorithm of trajectory planning.

Figure 10. Hyundai's HS220 in the production line.

have flexible options for optimization problems to implement the constraints. In this vein, the GA and PSO are more developed than other heuristic algorithms. Specifically, first, the GA and then PSO provide more practical options than other heuristic algorithms such as ant colony. A flowchart of the explained optimization algorithm is depicted in Fig. 8.

5. Case Study

In this study, the foregoing optimization problems are conducted on a Hyundai's HS220 serial robot (like ABB arm robot, and PUMA [34]) with six revolute joints (Fig. 10), which is used in a production line of the Saipa Corporation. Also, the gun is attached to the end of the robot to operate the spot-welding procedure.

To express the position of the robot using forward kinematics, the coordinate systems of each link are represented by the Denavit–Hartenberg (D-H) notation [29]. The parameters of the robot's geometry, the coordinate systems, are shown in Fig. 11. The D-H parameters are specified in Table 2. The dimensions of the links, according to Fig. 11, are listed in Table 3, and other characteristics

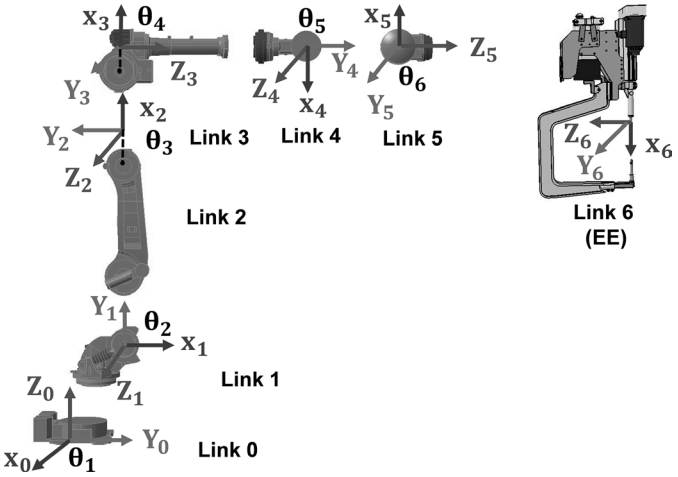

Figure 11. Links and coordinates of the robot.

such as mass, mass centre vectors, as well as inertial tensors with respect to the body coordinate systems are specified in Table 4, according to Fig. 11, which are extracted from the CAD files such as [35].

The joint constraints described in the previous sections including limitations on joints angle, maximum angular velocity, and maximum possible input torque are specified in Table 5. These values are chosen from the Hyundai Catalog associated with the HS220 robot. However, for the first three joints in the series, no restriction is mentioned in the manufacturer's catalogue. Also, in the corporation due to a conservative decision, and to avoid any unexpected damage to the robot, the maximum angular velocity of joints is usually considered from 60% to 70% of the maximum value.

In the production line, the spot-welding procedure is performed *via* four robot stations. In this study, the optimization problems are defined on one of these stations, as it is noted in Fig. 13, while the other three robots

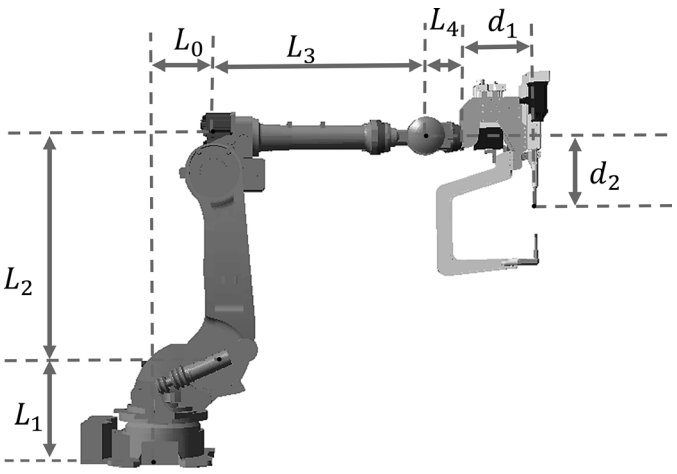

Figure 12. Geometric parameters of HS220 robot.

Table 2
D-H Parameters

Joint i	$\boldsymbol{\theta_i}$	$\mathbf{d_i}$	$\mathbf{a_i}$	$\boldsymbol{\alpha_i}$
1	$\pi/2$	L_1	L_0	$\pi/2$
2	$\pi/2$	0	L_2	0
3	0	0	0	$\pi/2$
4	π	L_3	0	$\pi/2$
5	π	0	0	$\pi/2$
6	π	$L_4 + d_1$	d_2	π

Table 3
Length of the Links (mm)

$\mathbf{L_0}$	$\mathbf{L_1}$	$\mathbf{L_2}$	$\mathbf{L_3}$	$\mathbf{L_4}$	$\mathbf{d_1}$	$\mathbf{d_2}$
312	608	1,330	1,250	220	425	396

perform the spot welding on other parts of a car's body. Figure 13 also depicts the reference coordinate frames which are attached to the robot base (X_0, Y_0) and the car body coordinate (X_C, Y_C). It is noteworthy to mention that the given target points are expressed in the coordinate frame which is attached to the car body. These target points are shown in Figs. 14 and 15.

When the gun is placed on a target point, there is a specific time for the spot-welding process itself. In fact, a spot-welding process includes six steps, closing, pressing, welding, holding, releasing, and opening. As it is reported by the company, depending on the location of target points, these steps take 1.1 or 1.3 s. This time does not play a role in the foregoing optimization problem and it is considered as a fixed time which will be added to the obtained travelling time.

6. Results

In the first step, it is important to search for an optimum combination between objective functions, the solution spaces, and optimization methodologies. In this regard, different strategies are tested on the first part of the trajectory, which is between the robot home posture (HP) and the first target point, and the concluding remarks will be extended to all remained parts of the trajectory. Therefore, in the sequel, optimization problems on the first part of trajectory will be conducted when (a) the travelling time is the objective function, and the solution is searched in the joint space *via* both the GA and PSO methods, (b) the energy consumption is the objective function, and the solution is searched in the joint space *via* both the GA and PSO methods, (c) the travelling time is the objective function, and the solution is searched in the Cartesian space *via* the GA method, and (d) the energy consumption is the objective function, and the solution is searched in the Cartesian space *via* the GA method.

Note that for the first part of the trajectory, the initial path branch index (b_I) and initial rotation angle of the gun (ϕ_I) should be arbitrarily selected; however, for the remained parts the path branch indices will be chosen automatically within the optimization algorithm.

6.1 Case A: The Travelling Time Minimization, in the Joint Space via the GA and PSO Methods

To minimize the travelling time from the robot home posture to the first target point (point 1 in Fig. 14), the population size of GA algorithm at each generation is chosen to be 200, and the number of particles in PSO method is chosen to be 80. As it is reported in Table 6, although the GA is computationally more expensive (approximately six times) than the PSO method, it ended up with better optimum results. Figure 16 depicts the trajectory of EE from the home posture to the first target point. As it is apparent in Fig. 17, the 4^{th} and 6^{th} joints are the last ones which attain their final states, and thus, they play a critical role in identifying the operation end time.

Results of the algorithm's calculation are listed below. This is related to the travelling time optimizations when the departure point is the home posture, and the arrival point is the first target point of the car's body.

6.2 Case B: The Energy Consumption Minimization, in the Joint Space via the GA and PSO Methods

In this case, there is a single-objective function, which is the energy consumption, and the design variables, constraints, and optimization algorithms are similar to the previous case. Figure 18 depicts the trajectory of EE from the home posture to the first target point. Also, Fig. 19 shows the input torques over the optimum trajectory provided by the motors, whose negative sign means it is applied in the opposite direction of the axis joint. The non-zero values at the initial and the end states provide

Table 4
Values of the Dynamic Parameters

Link i	$\mathbf{m_i(Kg)}$	${}^{\mathbf{i}}\mathbf{r_{G_i}(mm)}$	${}^{\mathbf{i}}\mathbf{I_{G_i}(Kg.m^2)}$
1	78.57	$[-215.6, +6.8, -171.2]^T$	$\begin{bmatrix} \mathbf{+9.894} & \mathbf{-0.725} & \mathbf{-1.129} \\ \mathbf{+0.725} & \mathbf{+10.231} & \mathbf{-0.393} \\ \mathbf{+1.129} & \mathbf{+0.393} & \mathbf{+10.622} \end{bmatrix}$
2	56.12	$[-849.3, -34.9, +181.1\ ]^T$	$\begin{bmatrix} \mathbf{+76.926} & \mathbf{+3.045} & \mathbf{-13.911} \\ \mathbf{-3.045} & \mathbf{+36.511} & \mathbf{+0.086} \\ \mathbf{+13.911} & \mathbf{-0.086} & \mathbf{+36.672} \end{bmatrix}$
3	41.68	$[-79.6, -3.6, +264.1\ ]^T$	$\begin{bmatrix} \mathbf{+9.658} & \mathbf{-0.001} & \mathbf{+1.539} \\ \mathbf{+0.001} & \mathbf{+13.034} & \mathbf{+2.223} \\ \mathbf{-1.539} & \mathbf{-2.223} & \mathbf{+7.181} \end{bmatrix}$
4	14.59	$[-0.1, +26.3, +36.6\ ]^T$	$\begin{bmatrix} \mathbf{+0.212} & \mathbf{+0.000} & \mathbf{-0.000} \\ \mathbf{-0.000} & \mathbf{+0.180} & \mathbf{+0.006} \\ \mathbf{+0.000} & \mathbf{-0.006} & \mathbf{+0.149} \end{bmatrix}$
5	3.36	$[-0.0, +0.0, +168.5\ ]^T$	$\begin{bmatrix} \mathbf{+0.108} & \mathbf{-0.000} & \mathbf{-0.000} \\ \mathbf{+0.000} & \mathbf{+0.108} & \mathbf{+0.006} \\ \mathbf{+0.000} & \mathbf{-0.006} & \mathbf{+0.109} \end{bmatrix}$
6	135.00	$[+575.7, +13.3, +215.7\ ]^T$	$\begin{bmatrix} \mathbf{+31.437} & \mathbf{+2.297} & \mathbf{-0.432} \\ \mathbf{-2.297} & \mathbf{+64.819} & \mathbf{+2.612} \\ \mathbf{+0.432} & \mathbf{-2.612} & \mathbf{+62.049} \end{bmatrix}$

Table 5
Constraints Values of the Robot

Joint i	**Angle limitation** (deg)	**Max. angular Vel.*** (deg/s)	**Max. torque** (N.m)
1	$[-178, +178]$	120	–
2	$[+10, +155]$	105	–
3	$[-10, +280]$	115	–
4	$[-360, +360]$	145	$\mathbf{3,422}$
5	$[-128, +128]$	145	$\mathbf{3,422}$
6	$[-360, +360]$	225	**770**

*The values inserted 60% of the catalogue.

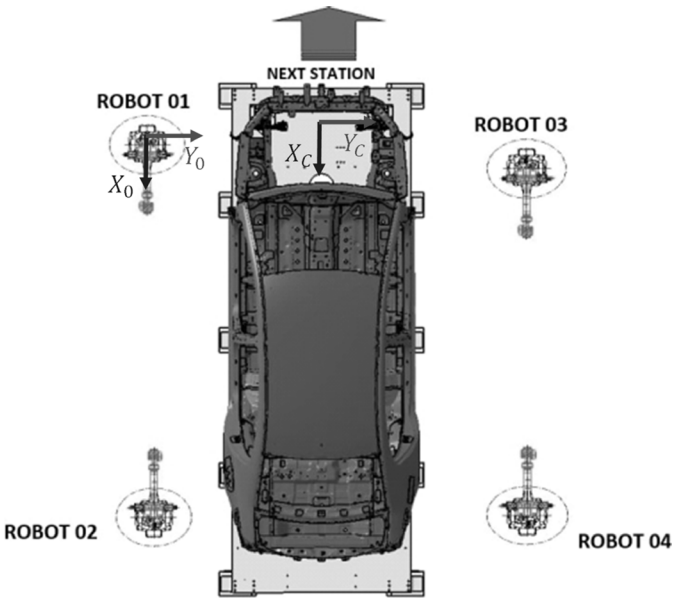

Figure 13. Station of the welding process.

the torques required for the static equilibrium of the robot at the home posture or during the welding process.

The results of the optimization problem are reported in Table 8. Same as what was concluded in the case a, the GA generates better results than the PSO, while its computational time is higher. By comparing the obtained results, it is concluded that solving the trajectory optimization problems with the GA results in a better trajectory. As all the optimization problems are conducted off-line and the CPU time is not an important issue, hereafter, the GA is chosen as the main optimization algorithm, and the PSO method is only considered when a result verification is needed.

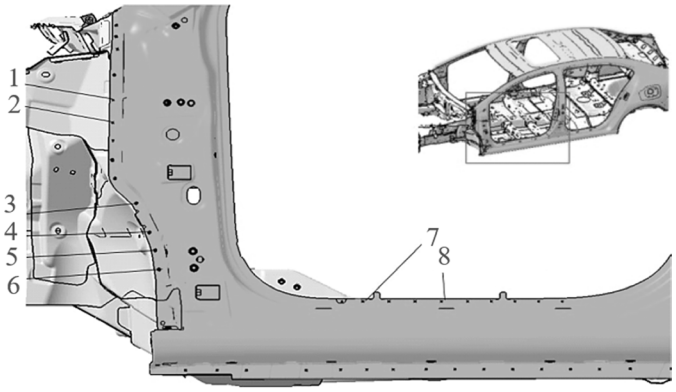

Figure 14. Target points on the sides of the car's body (a).

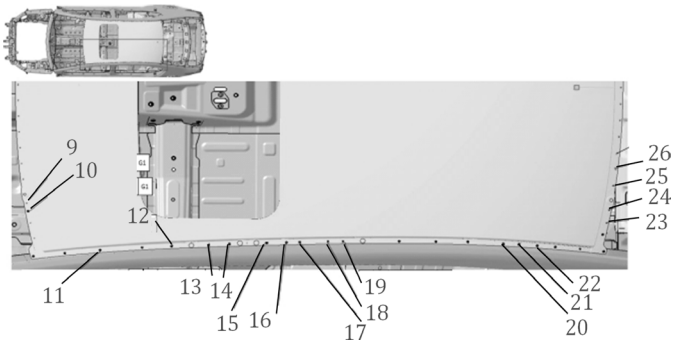

Figure 15. Target points on the roof of the car's body (b).

6.3 Case C: The Travelling Time Minimization in the Cartesian Space via the GA Method

In this problem, the minimum travelling time is targeted while the solution is searched in the Cartesian space by means of the GA method. Table 9 presents the characteristics of the obtained trajectory, and the corresponding travelling time, energy consumption, and the CPU time, which is taken by the optimization algorithm. As it is depicted in Fig. 20, optimization in the Cartesian space results in an optimum trajectory that is straighter than the one obtained in the joint space. Also, due to additional calculations of the Jacobin matrix in the Cartesian space, CPU time is elevated. The angular velocity of the joints related to this Cartesian trajectory are depicted in Fig. 21.

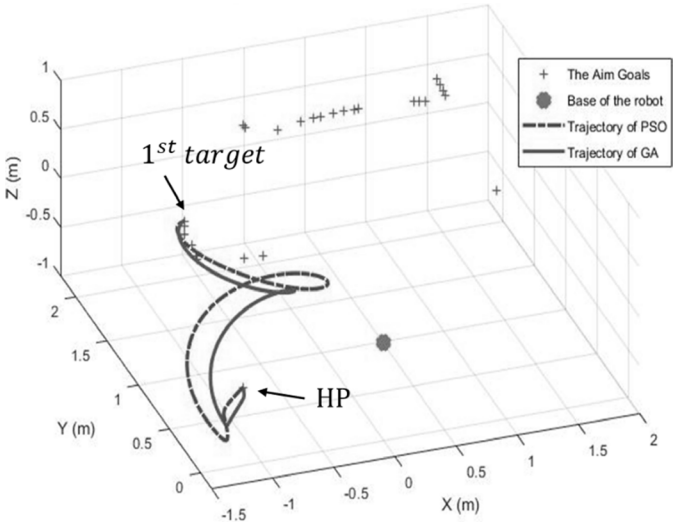

Figure 16. The trajectory of EE for the minimized travelling time *via* GA and PSO method.

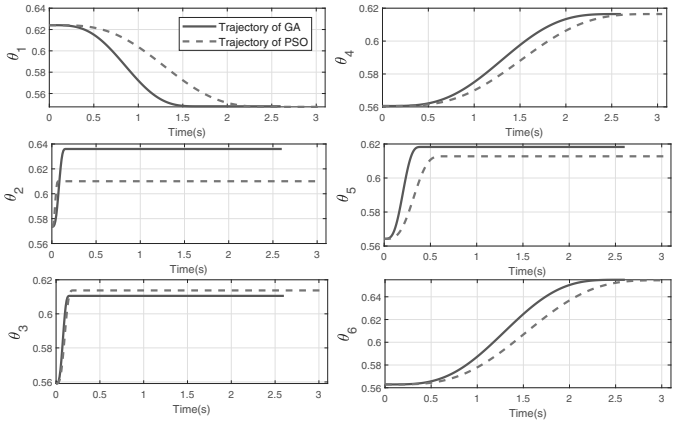

Figure 17. The angle of the joints (rad) for the trajectory of the minimized travelling time *via* GA and PSO method.

Table 6
Results of the Case (A)

Algorithm	Optimum variables		
	$\dot{\vec{\theta}}_{max}(\frac{rad}{s})$	$\phi_F(rad)$	b_F
GA	$[2.08\ 0.74\ 1.54\ 2.01\ 2.50\ 3.34]^T$	-0.60	3
PSO	$[1.39\ 0.91\ 1.31\ 1.73\ 1.42\ 2.80]^T$	-0.48	3
	Results of the minimization		
	Travelling time (s)	Consumed energy (j)	CPU time (s)
GA	**2.61**	8,015	6,215
PSO	**3.05**	7,168	1,226

*Initial branch index of the home posture assumed same for all calculations.

Table 7
Results of the Algorithm's Calculation for Case (A)

	GA	PSO	
Population	200	80	Number of particles
Generations	52	147	Generations
Funcount	541,000	24,000	Funcount

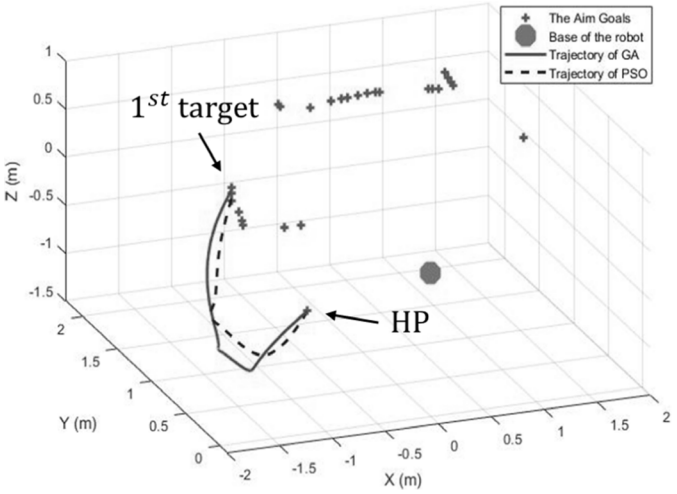

Figure 18. The trajectory of EE for the minimized energy consumption *via* GA and PSO method.

6.4 Case D: The Energy Consumption Minimization in the Cartesian Space via the GA Method

This case targets the minimization of energy consumption of the robot in the Cartesian space, which is solved by means of the GA. Table 10 presents the characteristics of the optimum trajectory, the corresponding travelling time, energy consumption, and CPU time. Figure 22 shows the input torques over the optimum trajectory provided by the motors.

By comparing the results of Tables 7 and 9, it can be concluded that lower speeds lead to lower energy consumption, which has led to higher travelling times. Thus, the obtained trajectories are not appropriate to be used in the production line. Moreover, the optimization problems which have been conducted on the first part of the

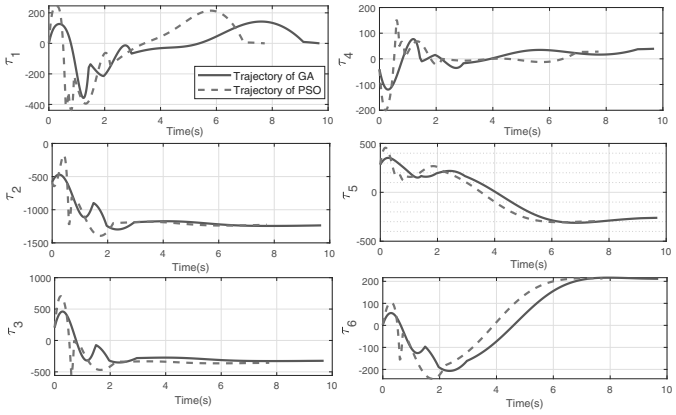

Figure 19. The input torques (N.m) for the trajectory of the minimized energy consumption *via* GA and PSO method.

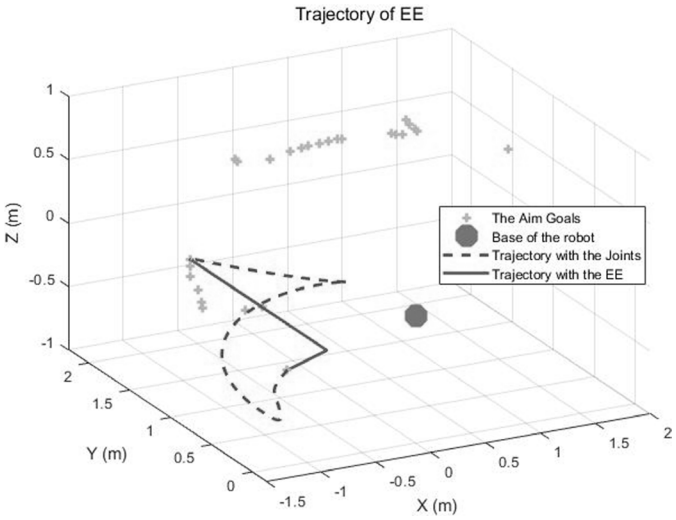

Figure 20. The trajectory of EE for the minimized travelling time in Cartesian and joint space *via* GA method.

Table 8
Results of the Case (B)

Algorithm	Optimal variables		
	$(\frac{rad}{s})\ \dot{\overrightarrow{\theta}}_{max}$	$\phi_F\ (rad)$	b_F
GA	$[0.40 \quad 0.33 \quad 0.43 \quad 0.56 \quad 0.62 \quad 0.80]^T$	0.16	4
PSO	$[0.53 \quad 0.60 \quad 0.65 \quad 0.81 \quad 0.71 \quad 1.11]^T$	-0.12	4
	Results of the minimization		
	Travelling time (s)	**Consumed energy (j)**	CPU time (s)
GA	9.69	**2, 386**	12, 632
PSO	7.73	**2, 870**	3, 683

Table 9
Results of the Case (C)

Design space	Optimal variables		
	$\overrightarrow{\chi}_{max}([\frac{m}{s}, \frac{rad}{s}])$	$\phi(rad)$	b_F
Cartesian space	$[2.90 \quad 1.11 \quad 1.03 \quad 1.02 \quad 0.97 \quad 3.00]^T$	-0.45	4
	Results for the minimization		
	Travelling time (s)	Consumed energy (j)	CPU time (s)
Cartesian space	**2.85**	$10,683$	$7,342$

Table 10
Results of the Case (D).

Design space	Optimal variables		
	$\overrightarrow{\chi}_{max}([\frac{m}{s}, \frac{rad}{s}])$	$\phi(rad)$	b_F
Cartesian space	$[0.73 \; 0.74 \; 0.29 \; 0.28 \; 0.12 \; 1.20]^T$	-0.31	4
	Results for the minimization		
	Travelling time (s)	**Consumed energy (j)**	CPU time (s)
Cartesian space	10.36	$\mathbf{2,647}$	$14,574$

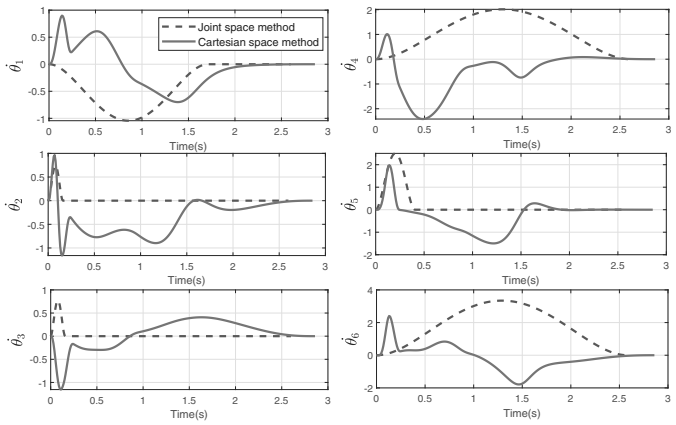

Figure 21. The angular velocity (rad/s) for the trajectory of the minimized travelling time in the Cartesian and joint spaces *via* GA method.

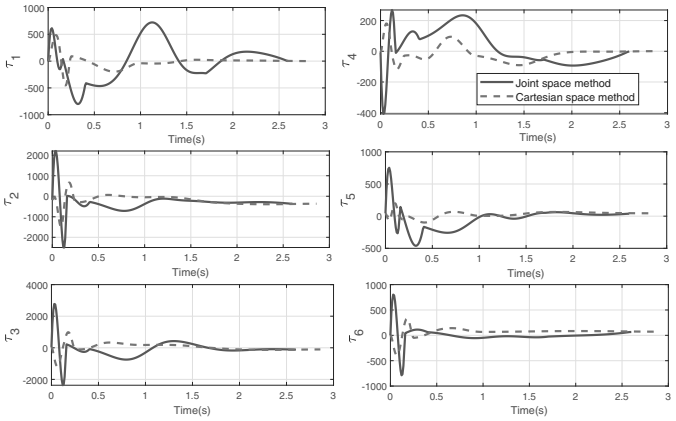

Figure 22. The input torques (N.m) for the trajectory of minimized energy consumption in Cartesian and joint space *via* GA.

trajectory in the joint space provided better results than what has been reached in the Cartesian space. Therefore, for the remained parts of the trajectory, the optimization problems will be conducted in the joint space and will be solved by the GA.

6.5 The Travelling Time Minimization of a Complete Spot-Welding Cycle in the Joint Space via the GA Method

Apparently, the energy consumption is directly related to the actuated joint rates and torques. In fact, for a constant external load, lower joint rates result in lower energy consumption. However, lower joint rates conversely lead to higher travelling time. As it is shown in Tables 8 and 9, the energy consumption minimization problems result in trajectories that are very time-consuming, which are not applicable in the industry. Therefore, for a complete spot-welding cycle, in the first try, a single-objective optimization problem is defined in the joint space in which the travelling time should be minimized by means of the GA. The obtained optimum trajectory and the angle of the joints are depicted in Figs. 23 and 24, respectively. Some noted positions are specified in Fig. 24 to generally illustrate each trajectory section belongs to which target points (according to Figs. 14 and 15). These positions include the HP, the target points that EE moves from the side of the car's body to the roof (8^{th} target point), and the ATT (auto tip dress) platform where EE reaches after welding

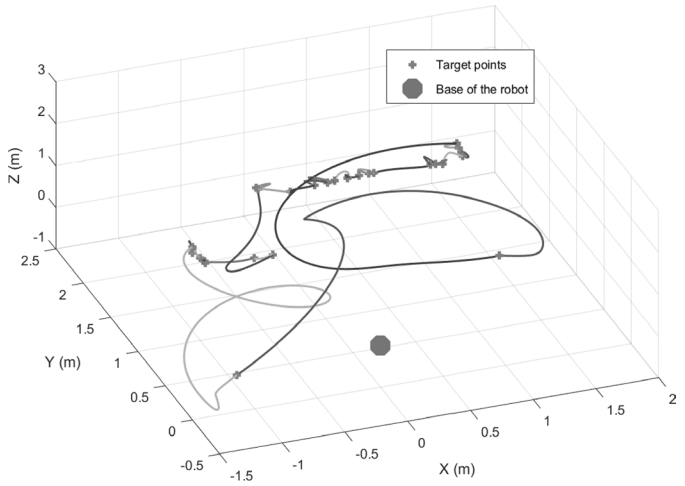

Figure 23. The trajectories of EE for the minimized travelling time for the complete spot-welding cycle.

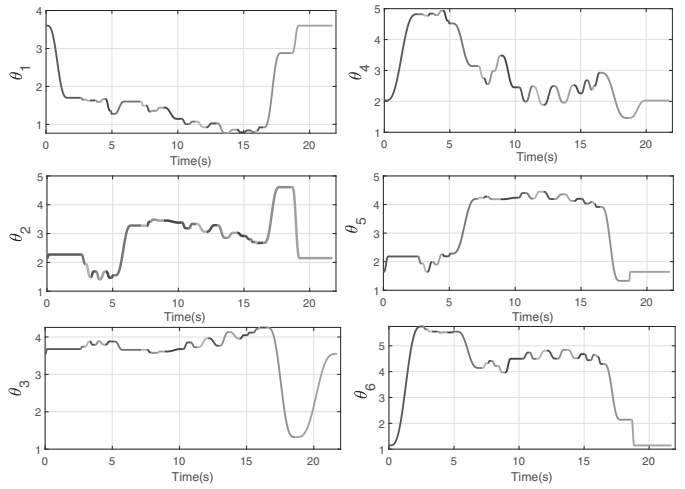

Figure 24. The angle of the joints (rad) for the trajectories of the minimized travelling time for the complete spot-welding cycle.

procedure and when it is needed to clean the electrode. At the end, the EE returns to the HP to complete a cycle.

The travelling time for the optimum trajectory is obtained as 21.73 s. In the Saipa production line, the travelling time for a similar spot-welding cycle, without the constant time of the welding process, is reported as 38 s. Therefore, the foregoing optimization problem results in an optimum trajectory that can reduce the travelling time by 42%; however, the energy consumption was not taken into account in the foregoing optimization problem.

6.6 Multi-Objective Trajectory Optimization of A Complete Spot-Welding Cycle in the Joint Space via the GA Method

As it was mentioned, the trajectory which was obtained in the energy minimization problem cannot be used in the industry as the corresponding travelling time is significantly large. On the other hand, the trajectories which are obtained in the time minimization problems are not efficient in terms of energy consumption. Hence, in such cases, a multi-objective optimization problem should be conducted with both travelling time and energy consumption as objective functions. A multi-objective optimization problem with conflicting criteria is normally ended up with a set of solutions, which is referred to as the *Pareto optimal set* [33]. In fact, the members of a Pareto set do not dominate each other in terms of the defined criteria of the optimization problem, and thus, graphically generate a frontier in the space of objectives of which the final solution can be selected.

In the case of a trajectory optimization for a spot-welding operation, each pair of target points comprise a Pareto set, which contains the optimum trajectories, and hence, another criterion is needed to select the final solution. In this regard, based on the results obtained for the first part of the trajectory, between the home posture and the first target point, a unique criterion can be established to be used for the remained parts.

The constraint moment vector of each joint can be calculated in addition to actuator torques by Eq. (A3). The magnitude of constraint moment at joints can be incorporated to define a criterion to select among the members of the final Pareto set. During an operation, constraint forces and the moments at joints vary; thus, the average of these constraints can be used to define a selection criterion. In this research, because the joints are revolute, the constraint moments are chosen. Therefore, the average of the magnitude of constraint moments is integrated from the initial and final travelling time for each joint. The summation for all joints is referred to as the *average constraint moments* (ACM), namely

$$\text{ACM} = \sum_{i=1}^{6} \frac{\int_{t_I}^{t_F} |\mathbf{n}_i| dt}{t_F - t_I} \tag{16}$$

in which $\mathbf{n}_i$ is the constraint moment of the ith joint, and t_I and t_F are the initial and final time, respectively. The selective choice in the Pareto set is the one with the *minimum* value of ACM. The general procedure of this selection among the members of Pareto set is depicted in Fig. 25.

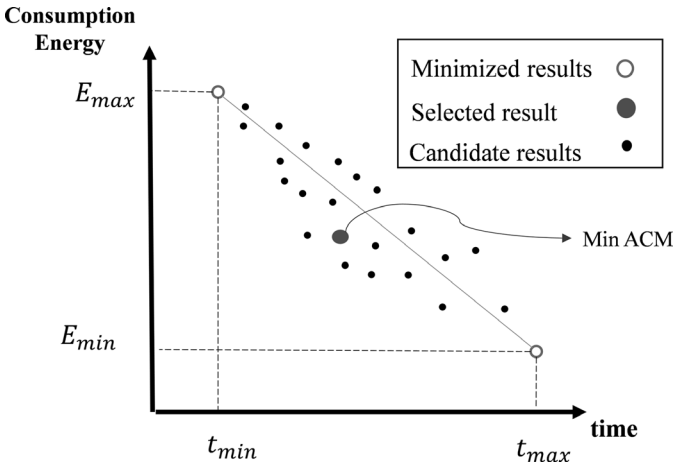

Figure 25. The criterion for Pareto selection.

Table 11
Comparing the Optimization Results

		Opt. time-energy		**Min. energy**		**Min. time**	
Trajectory	**Algo./Space**	**E (j)**	**T (s)**	**E (j)**	**T (s)**	E (j)	**T (s)**
From HP to the first target	GA/Joint	**3,736**	**4.42**	**2,386**	9.69	8, 015	**2.61**
	PSO/joint	-	-	**2, 870**	7.73	7,168	**3.05**
	GA/Cartesian	-	-	**2, 647**	10.36	10,683	**2.85**
Complete cycle	GA/Joint	**17,022**	**30.61**	-	-	31,208	**21.73**

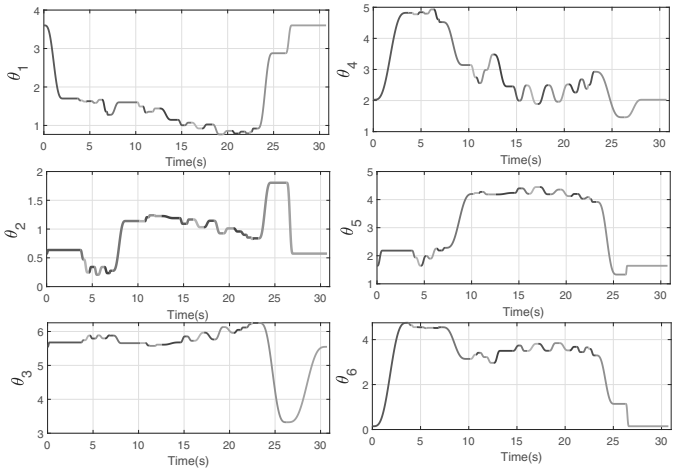

Figure 26. The angle of the joints (rad) for the trajectories of the optimized multi-objective for the complete spot-welding cycle.

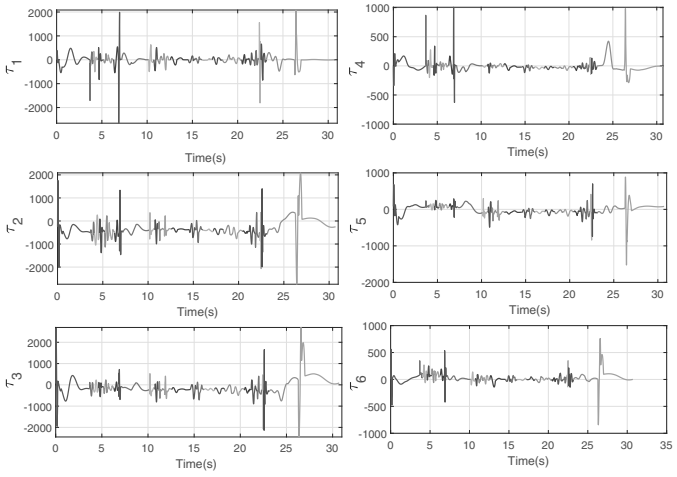

Figure 27. The input torques (N.m) for the trajectories of the optimized multi-objective for the complete spot-welding cycle.

The complete trajectory for a spot-welding procedure can be, then, formed by connecting the selected trajectories between all pairs of target points. In this study, for the obtained complete trajectory, the total travelling time and energy consumption are calculated as 30.61 (s) and 17,022 (j), respectively. Comparing the results of the multi-objective (time and energy) and the single-objective (time) minimization problems reveals that the total travelling time of the spot-welding operation is increased by 41% while the total energy consumption decreased by 45%. For a complete cycle, the plots of joints angle and the actuator torques are shown in Figs. 26 and 27.

Table 12
Comparing the Optimization Results with the Production Line

	Current production line	**Opt. time-energy trajectory**	**Min. time trajectory**
Time without considering stopping	38 s	30.61 s	21.73 s
Time with considering stopping	67 s	59.61 s	50.73 s
Capacity per working a day (8 h)	567	483	429

The energy consumption and travelling time of all conducted optimization problems are summarized in Table 11. For a better comparison, the constant stopping time of a spot-welding process is added to the travelling time, and the results are listed in Table 12. In the current production line, the nominal capacity of this workstation is 429 car's bodies in a working day (8 h). However, by resorting to the optimum trajectories, which are resulted from the foregoing time-energy optimization and time minimization, the capacity increases to 483 and 567, respectively.

7. Conclusion

This study aimed at a trajectory planning for a spot-welding procedure in the production line of an automotive industry. In this regard, the trajectory planning of a real welding robot by considering the most effective parameters is conducted. Different optimization problems were

first defined for the first part of the trajectory which is between the robot home posture and the first target point. Therefore, the different combination of objective functions, solution spaces, and the optimization methodologies was examined. Comparing the results in both design space stated that the almost straight lines between the points as the solution of the Cartesian space could not necessarily lead to obtaining generally optimal solutions, but the results of the joint space with carved lines of the EE obtained the most optimal solutions. Furthermore, the optimization by utilizing GA attained more precise results than PSO. Thus, these concluding remarks were extended to the remained pairs of target points to compute the corresponding trajectory in between by means of the GA.

The single-objective time minimization problem for a complete spot-welding cycle resulted in a trajectory with which the travelling time of a complete spot-welding cycle, including a fixed time for the spot-welding process, is reduced by 24% comparing with the current situation in the production line and this result just belongs to the specified robot in the station. Due to the conflicting behaviour of the travelling time and energy consumption, a multi-objective optimization problem was defined with both the travelling time and energy consumption as objective functions. The obtained trajectory reduced the travelling time by 11% while 45% less energy was consumed, comparing with the result of the time-minimization problem. Considering a normal operation, which now takes 8 h with 429 car's body in a day, the robot is now capable of performing the same task over 567 car's body for the minimized time trajectory or over 483 ones for the minimized time-energy trajectory. It must be noted that the significant reduction values in the optimization results in the production line are the result of the fact that determining the robots' trajectories is done manually by human operators, and that being optimal criteria do not play a crucial role in implementation of the trajectories.

Appendix A: The Dynamic Method

To obtain the energy consumption, first, the actuator torques of the joints must be calculated. In this line of thought, the dynamic methods were implemented. The Newton–Euler formulations were used to calculate the actuator torques and the constraint forces and moments. In fact, the constraint force/moments are considered as a criterion in the multi-objective optimization to select between Pareto sets. Noteworthy to mention that the calculated actuators were verified by the Lagrange–Euler formulation.

According to the Newton–Euler formulation [1] for each link, it can be written,

$$\begin{aligned} \mathbf{F_i} &= \mathbf{f_i} - \mathbf{f_{i+1}} \ , \\ \mathbf{N_i} &= \mathbf{n_i} - \mathbf{n_{i+1}} + \mathbf{d_{G_i}} \times \mathbf{f_i} - \mathbf{r_{G_i}} \times \mathbf{f_{i+1}} \end{aligned} \tag{A1}$$

where $\mathbf{f}_i$ and $\mathbf{n}_i$ are the exerted force and moment on the link ith, respectively. $\mathbf{d}_{G_i}$ and $\mathbf{r}_{G_i}$ are the position vectors from the mass centre to the body coordinates, and $\mathbf{F}_i = m_i \boldsymbol{a}_{G_i}$ and $\mathbf{N}_i = \dot{\boldsymbol{\omega}}_{G_i} + \boldsymbol{\omega}_{G_i} \times (\mathbf{I}_i \boldsymbol{\omega}_{G_i})$ which can be calculated by the kinematic equations. According to the number of links, the indices are considered as $i = 1, \ldots, 6$ which corresponds to the links numbers. The parameters in the equations are depicted in Fig. 28.

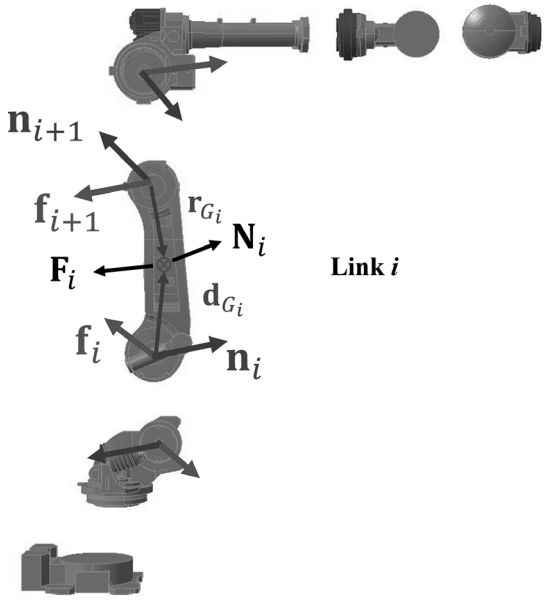

Figure 28. Forces and moments on the link **i**.

The recursive equation can be sorted as below from the last link to the first to calculate the $\mathbf{n}_i$ as the constraint moment vector of the joint,

$$\begin{aligned} \mathbf{f_i} &= \mathbf{F_i} + \mathbf{f_{i+1}} \ , \\ \mathbf{n_i} &= \mathbf{n_{i+1}} + \mathbf{N_i} - \mathbf{d_{G_i}} \times \mathbf{f_i} + \mathbf{r_{G_i}} \times \mathbf{f_{i+1}} \\ \mathbf{i} &= \mathbf{6}, \mathbf{5}, \ \ldots, \mathbf{1} \end{aligned} \tag{A2}$$

In the next step, the actuator torques are the projection of the moment vector in the direction of the joint's axis. Therefore, for the revolute joints, the τ_i can be obtained as below,

$$\tau_{\mathbf{i}} = \mathbf{n_i^T} \mathbf{e}_i \tag{A3}$$

in which $\mathbf{e}_i$ is the unit vector along the revolute joint axis. With calculated τ_i *via* dynamic equations, $\dot{\theta}_i$ *via* kinematic equations, and the time integral during the movement from one point to another point, the energy consumption can be obtained as below,

$$\mathbf{E} = \sum_{\mathbf{i=1}}^{\mathbf{6}} \int_{\mathbf{t_I}}^{\mathbf{t_F}} |\tau(\mathbf{t})^{\mathbf{T}} \dot{\theta}(\mathbf{t})| \mathbf{d}t$$

where $\boldsymbol{\tau}(t)$ is the vector of the actuator torques.

Appendix B: Verification

To verify the kinematic equations, Simscape (Simmechanic) as a multibody tool of Matlab is used. In an arbitrary trajectory, the position of the end-effector is calculated by the equations, and the software and the results are depicted in Fig. 29. The results show the verification of the forward and inverse kinematic equations.

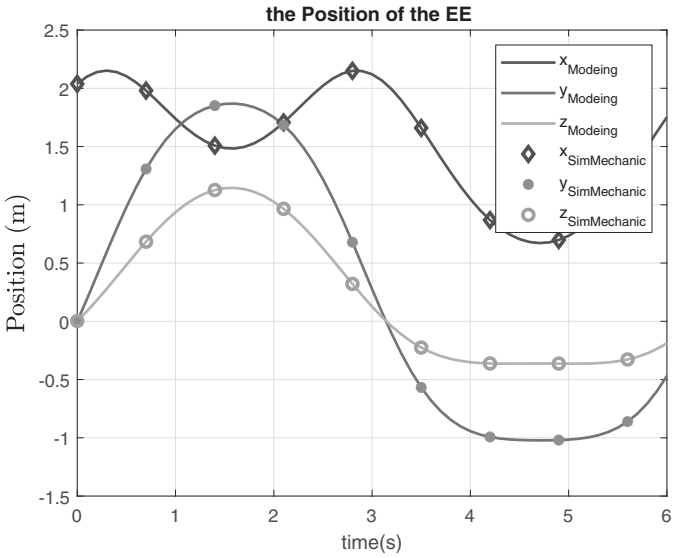

Figure 29. The position of EE to validate kinematic equations.

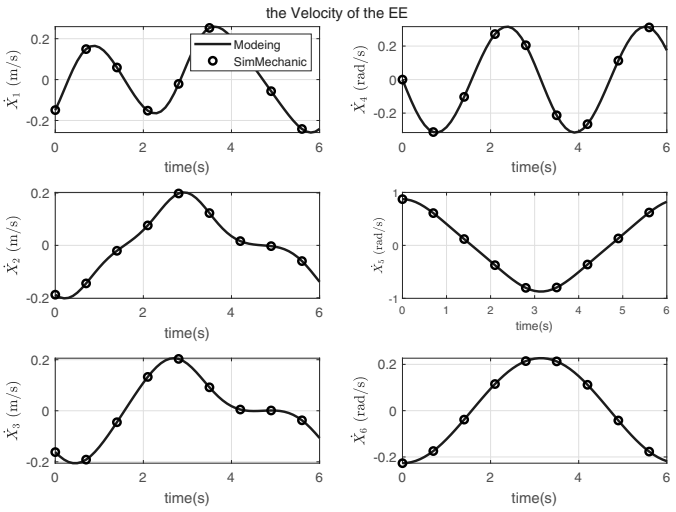

Figure 30. Velocity of EE to validate the Jacobian matrix.

Also, the angular velocity and the linear velocity of the end-effector are compared in Fig. 30 that can verify the Jacobian matrix. This matrix is the mapping between the joint rates and the Cartesian components of EE velocity.

Simscape (Simmechanic) also was used to verify the mathematical model of the dynamic equations. The Newton–Euler and Lagrange equations were used to implement the model, and the results on a trajectory are depicted in Fig. 31.

To clarify the difference of the calculated results, the errors can be obtained as below and are depicted in Fig. 32,

$$e_i = \left| \frac{\tau_{Moldel} - \tau_{Software}}{\tau_{Software}} \right| \times 100, \quad i = 1,\ 2,\ \ldots,\ 6 \quad \text{(B1)}$$

Consequently, by comparing the results the dynamic model is verified.

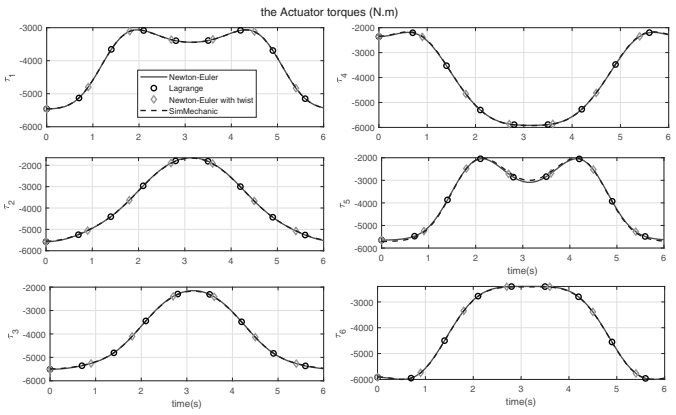

Figure 31. Comparison actuator torques in the model and the software.

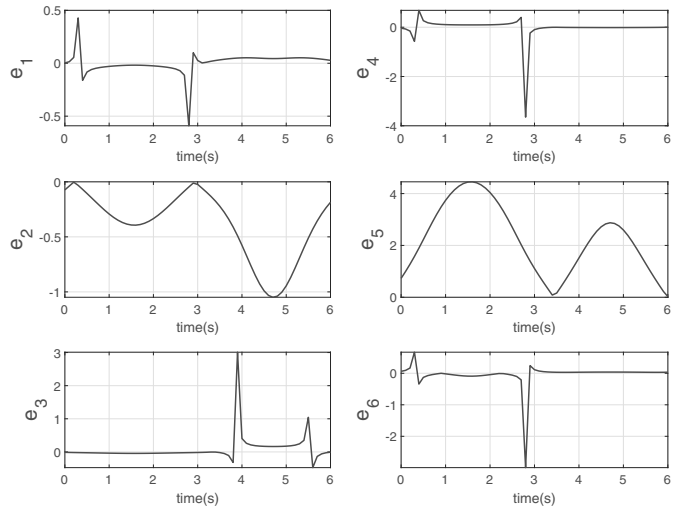

Figure 32. The error (in %) of the torques in the model and the software.

References

[1] K. Paes, *et al.*, Energy efficient trajectories for an industrial ABB robot, *Procedia Cirp*, *15*, 2014, 105–110.

[2] H.S. Lim, *et al.*, Particle swarm optimization algorithms with selective differential evolution for AUV path planning, *International Journal of Robotics and Automation*, *9*(2), 2020, 94–112.

[3] C. Hansen, *et al.*, Enhanced approach for energy-efficient trajectory generation of industrial robots, in *2012 IEEE International Conference on Automation Science and Engineering (CASE)* (IEEE, 2012).

[4] M. Pellicciari, *et al.*, A method for reducing the energy consumption of pick-and-place industrial robots, *Mechatronics*, *23*(3), 2013, 326–334.

[5] A. Mohammed, *et al.*, Minimizing energy consumption for robot arm movement, *Procedia Cirp*, *25*, 2014, 400–405.

[6] C. Hansen, J. Kotlarski, and T. Ortmaier, Optimal motion planning for energy efficient multi-axis applications, *International Journal of Mechatronics and Automation*, *4*(3), 2014, 147–160.

[7] P. Boscariol and D. Richiedei, Energy-efficient design of multipoint trajectories for Cartesian robots, *The International Journal of Advanced Manufacturing Technology*, *102*(5–8), 2019, 1853–1870.

[8] A. Fahim, M. Tetreault, and D. Necsulescu, Robot trajectory optimisation with dynamic constraints, *The International Journal of Advanced Manufacturing Technology*, *3*(1), 1988, 71–76.

[9] Q.-C. Pham, A general, fast, and robust implementation of the time-optimal path parameterization algorithm, *IEEE Transactions on Robotics*, *30*(6), 2014, 1533–1540.
[10] Q. Zhang and M.-Y. Zhao, Minimum time path planning of robotic manipulator in drilling/spot welding tasks, *Journal of Computational Design and Engineering*, *3*(2), 2016, 132–139.
[11] L. Cheng, *et al.*, An improved PSO algorithm for time-optimal trajectory planning of Delta robot in intelligent packaging, *The International Journal of Advanced Manufacturing Technology, 107*, 2019, 1091–1099.
[12] M. Ghasemi, N. Kashiri, and M. Dardel, Time-optimal trajectory planning of robot manipulators in point-to-point motion using an indirect method, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *226*(2), 2012, 473–484.
[13] S. Diao, *et al.*, Task-level time-optimal collision avoidance trajectory planning for grinding manipulators, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *233*(8), 2019, 2894–2908.
[14] F.J. Abu-Dakka, *et al.*, Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots, *The International Journal of Advanced Manufacturing Technology*, *89*(1–4), 2017, 389–406.
[15] G. Wu, W. Zhao, and X. Zhang, Optimum time-energy-jerk trajectory planning for serial robotic manipulators by reparameterized quintic NURBS curves, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2020, 0954406220969734.
[16] H. Liu, X. Lai, and W. Wu, Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints, *Robotics and Computer-Integrated Manufacturing*, *29*(2), 2013, 309–317.
[17] J. Huang, *et al.*, Optimal time-jerk trajectory planning for industrial robots, *Mechanism and Machine Theory*, *121*, 2018, 530–544.
[18] R. Saravanan, S. Ramabalan, and C. Balamurugan, Evolutionary optimal trajectory planning for industrial robot with payload constraints, *The International Journal of Advanced Manufacturing Technology*, *38*(11–12), 2008, 1213–1226.
[19] R. Saravanan, S. Ramabalan, and C. Balamurugan, Evolutionary multi-criteria trajectory modeling of industrial robots in the presence of obstacles, *Engineering Applications of Artificial Intelligence*, *22*(2), 2009, 329–342.
[20] H. Fang, S. Ong, and A. Nee, Orientation planning of robot end-effector using augmented reality, *The International Journal of Advanced Manufacturing Technology*, *67*(9–12), 2013, 2033–2049.
[21] M. Chalvin, *et al.*, Layer-by-layer generation of optimized joint trajectory for multi-axis robotized additive manufacturing of parts of revolution, *Robotics and Computer-Integrated Manufacturing*, *65*, 2020, 101960.
[22] M. Givehchi, A.H. Ng, and L. Wang, Spot-welding sequence planning and optimization using a hybrid rule-based approach and genetic algorithm, *Robotics and Computer-Integrated Manufacturing*, *27*(4), 2011, 714–722.
[23] X. Wang, *et al.*, Double global optimum genetic algorithm–particle swarm optimization-based welding robot path planning, *Engineering Optimization*, *48*(2), 2016, 299–316.
[24] A. Kovács, Integrated task sequencing and path planning for robotic remote laser welding, *International Journal of Production Research*, *54*(4), 2016, 1210–1224.
[25] H. Yang and H. Shao, Distortion-oriented welding path optimization based on elastic net method and genetic algorithm, *Journal of Materials Processing Technology*, *209*(9), 2009, 4407–4412.
[26] X. Wang, Y. Yan, and X. Gu, Spot welding robot path planning using intelligent algorithm, *Journal of Manufacturing Processes*, *42*, 2019, 1–10.
[27] V. Beik, H. Marzbani, and R. Jazar, Welding sequence optimisation in the automotive industry: A review, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *233*(17), 2019, 5945–5952.
[28] X. Wang, *et al.*, A survey of welding robot intelligent path optimization, *Journal of Manufacturing Processes*, *63*, 2020, 14–23.
[29] K.S. Fu, R. Gonzalez, and C.G. Lee, *Robotics: Control Sensing. Vis.* (Tata McGraw-Hill Education, 1987).
[30] J.J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd edition (Pearson Education India, 2009).
[31] J. Angeles, *Fundamentals of Robotic Mechanical Systems* (Springer, 2002).
[32] E. Sharafian, A. Taghvaeipour, and M. Ghassabzadeh, Revisiting screw theory-based approaches in the constraint wrench analysis of robotic systems, *Robotica*, *40*(5), 2022, 1406–1430.
[33] S.S. Rao, *Engineering Optimization: Theory and Practice* (John Wiley & Sons, 2019).
[34] A. Azarfar, Self-tuning fuzzy task space controller for puma 560 robot, *International Journal of Robotics and Automation (IJRA)*, *7*(4), 2018, 273–282.
[35] M. Mihola, Z. Zeman, and D. Fojtik, Automation of the design of the cross-section of the manipulator arms profile, *International Journal of Robotics and Automation (IRJA)*, *10*(3), 2021, 170.

Biographies

Ehsan Sharafian Moghaddam received his M.S. degree in dynamic and control of mechanical systems at Amirkabir University of Technology. Currently, he is focusing on parallel manipulators, especially those used in rehabilitation. In addition, he is working to implement heuristic algorithms to optimize the performance of the manipulators.

Maryam Ghassabzadeh Saryazdi studied Mechanical Engineering at the Sharif University of Technology, Iran, from 1994 to 1998. She received her M.Sc. from Amirkabir University of Technology, Iran in 2001 and her Ph.D. from the Sharif University of Technology in 2009. In 2016, she was appointed to Chair at the Amirkabir University of Technology. She is currently an assistant professor at the Technology Institute of Mechanical Engineering (TIME), Amirkabir University of Technology. Her current research interest includes Dynamic and Vibration analysis of mechanical systems.

Afshin Taghvaiepour is an assistant professor at Amirkabir University of Technology (AUT), Iran. He received his B.Sc. from AUT in 2006, his M.Sc. from Sharif University of Technology (SUT) in 2008, and his Ph.D. form McGill University in 2012. His research interests include Structural and Dynamic Analysis of Mechanical and Robotic Systems and Finite Element Analysis.