robotics

MDPI

Article

Evaluation Criteria for Trajectories of Robotic Arms

Michal Dobiš [1,*], Martin Dekan [1], Peter Beňo [2], František Duchoň [1] and Andrej Babinec [1]

[1] Institute of Robotics and Cybernetics, Slovak University of Technology in Bratislava, 812 19 Bratislava, Slovakia; martin.dekan@stuba.sk (M.D.); frantisek.duchon@stuba.sk (F.D.); andrej.babinec@stuba.sk (A.B.)

[2] Department of Robot Applications, Photoneo s.r.o. Company, 821 09 Bratislava, Slovakia; beno@photoneo.com

* Correspondence: michal.dobis@stuba.sk

Abstract: This paper presents a complex trajectory evaluation framework with a high potential for use in many industrial applications. The framework focuses on the evaluation of robotic arm trajectories containing only robot states defined in joint space without any time parametrization (velocities or accelerations). The solution presented in this article consists of multiple criteria, mainly based on well-known trajectory metrics. These were slightly modified to allow their application to this type of trajectory. Our framework provides the methodology on how to accurately compare paths generated by randomized-based path planners, with respect to the numerous industrial optimization criteria. Therefore, the selection of the optimal path planner or its configuration for specific applications is much easier. The designed criteria were thoroughly experimentally evaluated using a real industrial robot. The results of these experiments confirmed the correlation between the predicted robot behavior and the behavior of the robot during the trajectory execution.

Keywords: trajectory optimization criteria; robotic arms; trajectory evaluation; path planning

check for updates

Citation: Dobiš, M.; Dekan, M.; Beňo, P.; Duchoň, F.; Babinec, A. Evaluation Criteria for Trajectories of Robotic Arms. *Robotics* **2022**, *11*, 29. https://doi.org/10.3390/robotics11010029

Academic Editors: António Paulo Moreira, Félix Vilariño and Pedro Neto

Received: 17 January 2022
Accepted: 11 February 2022
Published: 15 February 2022

1. Introduction

In many modern applications of robotic arms, there is a need to compute collision-free trajectories with variable start and goal positions. An example of such application is the problem of bin-picking, where the main goal is to pick randomly placed objects from within the bin. This problem is described in the literature as a problem of dynamic path planning. Several solutions to this problem have been proposed and applied over the past decade.

Many of these popular methods are based on randomized sampling, such as probabilistic roadmap methods (PRM) [1] or rapidly random trees (RRT) [2]. Another approach to path planning is using optimization-based methods. Algorithms such as covariant Hamiltonian optimization for motion planning (CHOMP) [3] or stochastic trajectory optimization for path planning (STOMP) [4] belong to this group. Most of the methods utilize random exploration of configuration space to speed up the computation. The disadvantage of such algorithms is their non-deterministic behavior. The repeated path planning computation from the same start state to the same goal state could yield different results. Furthermore, attributes of the calculated paths depend dramatically on the chosen path planning algorithm and its configuration. The trajectories can differ in the distance traveled by the endpoint of the robotic arm, energy consumption, or maximal joint accelerations achieved during the motion execution. Several metrics were proposed for the measurement of difference between two robot states.

Many path planner comparisons and path planner articles for specific applications evaluate mainly planning time and success rate. In their paper [5], Rodriguez and Suarez (2015) provide a comparison of sampling-based algorithms, which is based on planning time, success rate, and number of samples. Likewise, Paulin et al. (2015) [6] compare path planners for grape vine pruning application, but the trajectory quality is not considered. Magyar et al. (2019) proposed a modified version of the STOMP algorithm called

GSTOMP [7] and use the term "smoothness", which represents a cumulative function of the end effector's linear and angular accelerations. Similarly, in their study [8] De Maeyer and Demeester (2021) are focused on benchmarking path planning for robotic arc welding, where path length is computed as the cumulative sum of joint differences. On the other hand, Larsen et al. (2017) [9] used the Euclidean distance of a tool center point for the path length calculation. In all of these algorithms factors/characteristics such as tool orientation, acceleration or jerks are neglected. In our previous research [10], we attempted to compare the quality of paths computed by various path planners and their configurations. However, we were not able to find any common framework for evaluation of the quality of the path.

It is hard to define what is the best universal trajectory, but it is possible to determine which trajectory is better with respect to a selected criterion or their combinations for specific setup and production type. It makes little sense to talk of the best universal trajectory, however it is possible to determine which trajectory is best suited depending on a selected criterion (or a combination of several criteria) for a specific setup and production type. Optimization typically aims to minimize the production cycle time, based on the robot's velocity and the total trajectory length. Other optimization criteria could be maximizing the durability of the robot gearing or decreasing energy consumption. [11]

In this article we formulate a complex trajectory evaluation framework consisting of multiple criteria. The objective of this article is not to provide a comparison of path planning algorithms, but to provide a methodology to compare the resulting paths. For the purpose of this article, the result of a path planner is a path defined as a set of robot states in the robot joint space.

In order to ensure a smooth motion execution, the robot control system needs to post-process the given path and execute its approximation. Therefore, it will not traverse all the positions precisely. Some of our criteria try to predict how the robot will behave during the execution. Therefore, in experimental evaluation of our framework, we tried to find the correlation between the planned and performed trajectory for each criterion.

The evaluation criteria are fully described in Section 2 "Criteria" and they are validated in Section 3 "Validation of criteria" with data measured using a real robot. These measurements are also used for evaluation of the aforementioned hypotheses about prediction of path execution, where the measured and predicted data are compared. Section 4 "Usage of criteria" provides criteria examples and an introduction on how to use them. The impact of environment complexity and the position of the bin were investigated and analyzed as well. Finally, Section 5, Results, summarizes and recapitulates the findings reported in previous sections.

2. Criteria

The first trivial criterion often used by manufacturers is the Cartesian distance of tool center point (TCP) between each waypoint of the path. This means that the sum of the Euclidean distance between each consecutive TCP position in R^3 should be minimal. In many cases programmers try to optimize the robotic program in order to reduce the production cycle time [12] in specific applications like spot welding [13], where the TCP path must be minimal. In the joint space, we could define the joint distance criterion, which is very closely related to the Cartesian distance criterion. The joint distance uses an accumulated summary of differences for each joint value. An interesting application is collaborative robotics, because robots must indicate appropriate proxemic behavior, and unnecessarily long and nonsensical trajectories could evoke fear [14].

$$x = \sum_{i=2}^{n} \sum_{j=1}^{m} ||q_{i,j} - q_{i-1,j}|| \tag{1}$$

$$x = \sum_{i=2}^{n} ||p_i - p_{i-1}|| \tag{2}$$

The score of the joint distance criterion is calculated using (1), where $q_{i,j}$ is joint value, i is waypoint ID of n waypoint, and j is the joint ID of m robot joints. Equation (2) is used for the calculation of the Cartesian distance score, where instead of the accumulated sum of joint values, the tool point position p_i is used.

The Cartesian distance does not include orientation, although in specific types of production the tool point orientation is fundamental e.g., in grinding, cutting, milling, or painting [15,16]. In the cases of object handling with pneumatic or magnetic grippers, picked objects could be lost during a move. Another interesting and practical example is object handling in the human environment [17,18] or liquid handling [19]. It follows then that the orientation change is the next crucial criterion. It is defined by (3), where q_{xi}, q_{yi}, q_{zi}, q_{wi} are values of tool point quaternion on i waypoints.

$$x = \sum_{i=2}^{n} \cos^{-1}\left(q_{xi} * \; q_{xi-1} + q_{yi} * \; q_{yi-1} + q_{zi} * \; q_{zi-1} + q_{wi} * \; q_{wi-1}\right) \tag{3}$$

The joint distance criterion could be modified by weights, which multiply each joint value, thus achieving optimization of selected joints [20]. In this article, we call this criterion control pseudo-cost.

$$x = \sum_{i=2}^{n} \sum_{j=1}^{m} w_j \left(||q_{i,j} - q_{i-1,j}|| \right) \tag{4}$$

The definition (4) of this feature is similar to joint distance, with the added multiplication by w_j which represents the weight in the range between 0 and 1. This could simplify energy consumption in the application, where the actual energy consumption often remains unknown before the trajectory is executed [21,22]. The computation of the energy for each robot link is dependent on weight and inertia. A great example is an arm on a mobile platform where the power is limited by a battery [23]. There are many approaches to trajectory planning for energy efficiency [24,25], where this criterion can prove useful for comparison and evaluation of planned paths.

In addition to control pseudo-cost, in [20] the robot displacement metric is mentioned that could be used as a relevant criterion. It could also be applicable in collaborative robotics, similarly as the joint distance and the Cartesian distance criteria. Definition of criterion by [20]: for any two configurations q_1 and q_2, a robot displacement metric is defined as (5),

$$x = \sum_{i=2}^{n} \max_{a \epsilon A} \{ ||a(q_i) - a(q_{i-1})|| \} \tag{5}$$

In which $a(q_i)$ is the position of the point a in the world when the robot A is at configuration q_i.

The aforementioned criteria are solely focused on robot states and positions, but do not include any robot dynamics. Industrial programmers often teach robots smooth trajectories without any high jerks, which have the potential to damage mechanical units or stop movement. Less jerk reduces the steadfastness and increases the robot's durability, which can economize production costs. Jerk is defined as the third time-derivation of position, which is applied in research for optimal time-jerk trajectory planning, as shown in Huang et al. (2017) [26]. However our current research does not include any time-parametrization of trajectories, therefore we propose four criteria hypotheses that focus on jerk and are calculated solely using robot positions. One of these hypotheses is the joint jerk, defined as a summary of jerks on each joint (6):

$$x = \sum_{i=2}^{n} \sum_{j=1}^{m} ||q_{i,j} - q_{i-1,j}|| di^3 \tag{6}$$

Similar to the above, the next criterion: the Cartesian jerk; is also defined as the third derivation of the tool point position, and its score equals the sum of all jerks between each waypoint (7):

$$x = \sum_{i=2}^{n} ||p_i - p_{i-1}||di^3 \tag{7}$$

The criterion is important in applications requiring gripper load, where higher jerk on the load is undesirable. Consider situations such as using soft robotic grippers [27] or a robot handling fragile materials or food [28].

Alternatively, another approach could be employed by finding the maximal jerk between waypoints, which could be applicable as a safety threshold. Therefore, this is defined as joint max jerk in the joint space, as opposed to Cartesian max jerk in the Cartesian space of the tool [29,30]. Joint max jerk is defined by (8) and Cartesian max jerk by (9) below.

$$x = \max_{q_i \epsilon Q} \left\{ \sum_{j=1}^{m} ||q_{i,j} - q_{i-1,j}||di^3 \right\} \tag{8}$$

In which q_i is a joint state on i waypoint, and Q represents all joint states of a path.

$$x = \max_{p_i \epsilon P} \left\{ \sum_{j=1}^{m} ||p_i - p_{i-1}||di^3 \right\} \tag{9}$$

In which p_i is tool position on i waypoint, and P represents all tool positions of a path.

3. Validation of Criteria

It is essential to verify the validity of the stated criteria as well as that of the hypothesis that criteria could predicate trajectory execution. The verification was carried out by planning and executing the trajectories using a real robot. Firstly, the paths are planned by the STOMP algorithm using FCL (Flexible Collision Library) [31]. The selection of the path planner is not relevant in this section and, therefore, the specific parameters of the path planner are not presented. Subsequently, these planned paths are executed on a real KUKA KR120 robot with gripper (see in Figure 1), controlled by KRC4.As the robot performs the motion measurements on positions and energy consumption are sampled in time and collected.

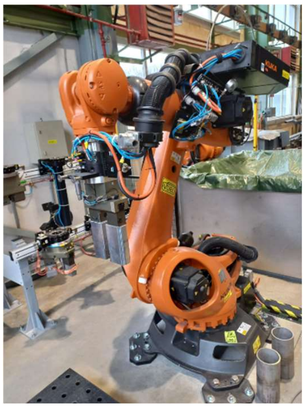


Figure 1. Kuka KR120 with gripper, used to executed planned paths in a series of criteria-validating experiments.

For the purpose of criteria validation an experiment was designed, where the robot moves from the start state [0.455, −1.506, 1.885, 0.0, 1.17, 0] to the goal state [−0.498, −1.543, 1.924, 0.0, 1.17, −1.061] with an obstacle added to its path. The path is illustrated in Figure 2. The dataset consists of 30 different paths between the start and the goal states, generated by the STOMP algorithm using FCL collision checking [31].

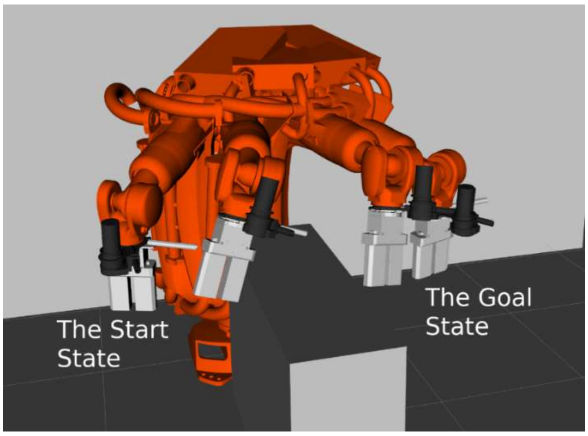


Figure 2. The path between the start state and the goal state.

The first four criteria joint distance, Cartesian distance, orientation change and robot displacement are directly related to the position of the robot in the joint space and of the Cartesian space of its tool point. As for the next set of criteria—joint jerk, joint max jerk, Cartesian jerk, Cartesian max jerk, and control pseudo-cost—these are evaluated separately, as they require the use of specific measurement procedures. The equations for criteria were applied to the entire dataset of planned paths. To provide a clear comparison, they were applied to the joint and Cartesian positions measured during planned path executions as well. Figure 3 shows the differences between planned and executed positions for all paths in the form of a histogram.

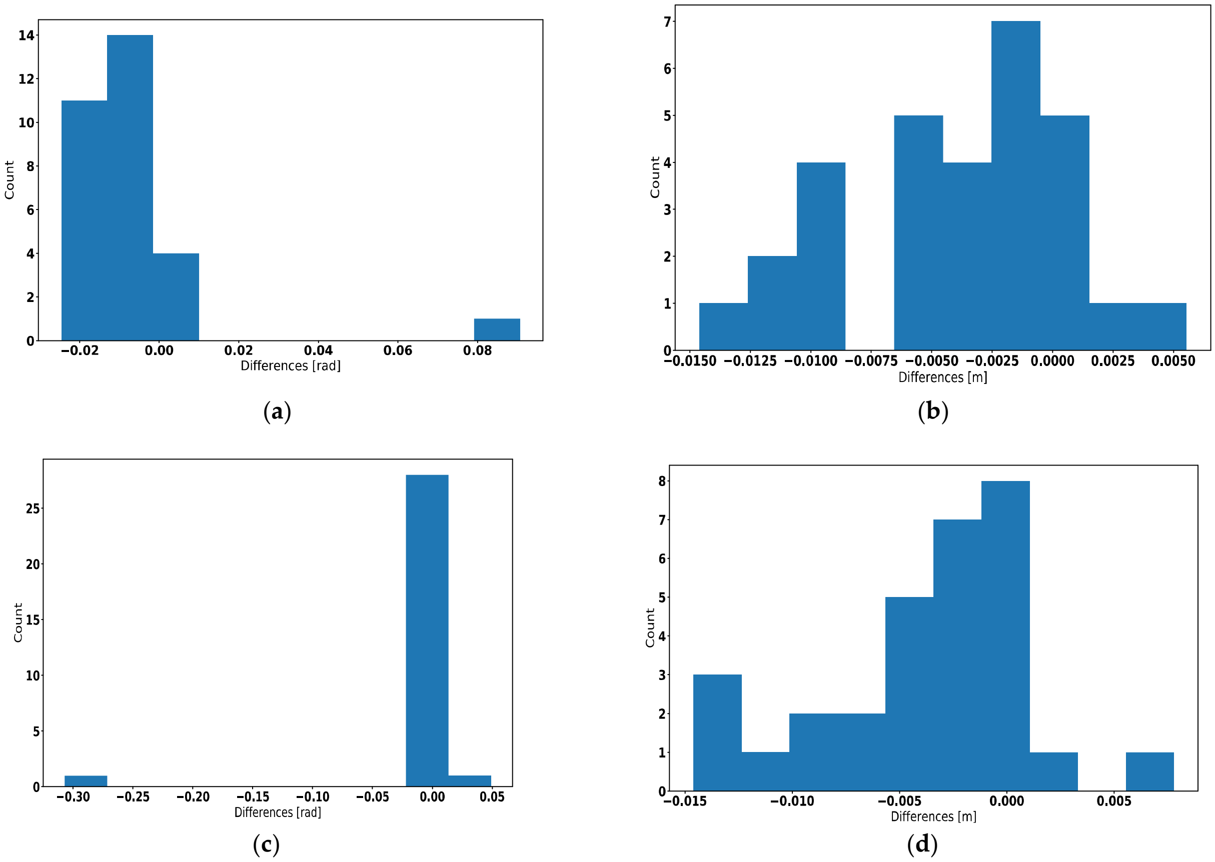


Figure 3. Histograms of differences between planned paths and executed trajectories, compared for the following criteria: (**a**) joint distance; (**b**) Cartesian distance; (**c**) orientation change; (**d**) robot displacement.

For further illustration, the averages and variances in Table 1 show that discrepancies between planned and executed trajectories are correlated.

Table 1. The average and the variance differences between planned path and executed trajectory.

Criterions	Average	Variance
Joint Distance [rad]	-5.75×10^{-3}	3.76×10^{-4}
Cartesian Distance [m]	-4.09×10^{-3}	2.22×10^{-5}
Orientation Change [rad]	-1.24×10^{-2}	3.09×10^{-3}
Robot Displacement [m]	-3.75×10^{-3}	2.52×10^{-5}

3.1. Control Pseudo-Cost

Energy consumption is an excellent example of where the control pseudo-cost criterion can play a part. It could simplify dynamic robot parameters for each arm joint, or these parameters could be replaced by criterion weights as seen in (4). However, the problem is that the weights are unknown, as they need to be calculated from the dynamic robot model. Unfortunately, not all robot vendors provide parameters such as weights, center of gravity, or an inertial matrix for each robot link. A further difficulty arises from the fact that the parameters in the whole robot workspace are not constant. They need be variable, depending on the type of movement, load, etc.

It is essential to ensure that the measuring process is not compromised by systematic error. Therefore, the robot is allowed to "warm up" as its energy consumption tends to be higher immediately after start-up. (The decreased energy consumption is illustrated in Figure 4). Variable velocity and acceleration could also affect the measurement adversely, therefore these and other robot parameters are kept constant.

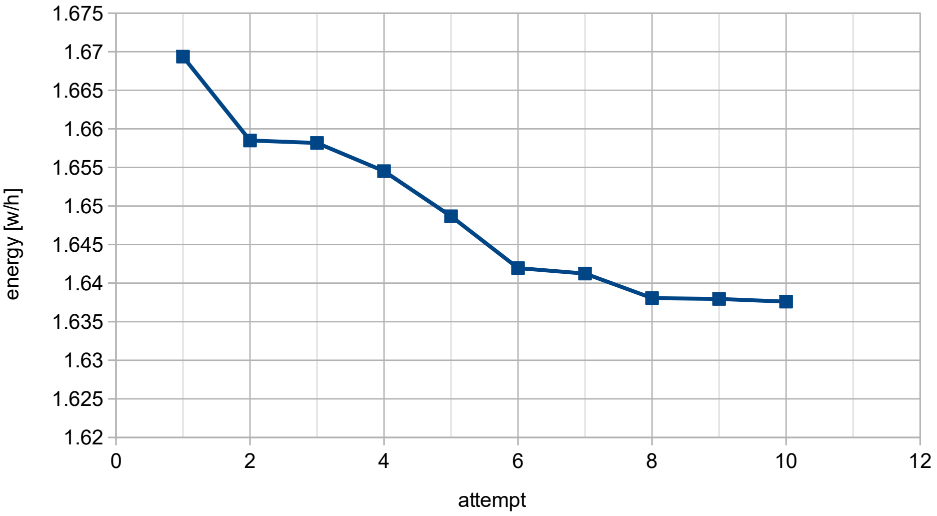


Figure 4. Energy consumption after a cold start of the robot.

The KUKA KRC4 controller allows for the measurement of consumed energy during a single trajectory execution. These measured values are used as a target for our optimization. As we do not possess all the robot parameters for the calculation of the criterion constants, we instead implement some simplifications. We employ the brute force method to determine the constants calculated on a specific set of paths, meaning that the paths will be planned between the same start and goal state. In our case, they are the positions defined in the introduction of the section "Validation of Criteria" (Figure 2). The robot will perform a "horizontal" movement (movement of the TCP of the robot will approximately correspond to the y-axis), and in this section, the experimental setup is denoted as no. 1.

Our working hypothesis is that the calculated constants will be suitable for use in similar robot motion. To prove this hypothesis, four experiments were prepared, where the first three are identical trajectories no.2 (Figure 5a "Longer" trajectories), no.3 (Figure 5b "Shorter" trajectories), and no.4 (Figure 5c TCP is further on the *x*-axis). The last trajectory, no.5. (Figure 5d "vertical" trajectory), is of a different type. A deviation between the actual energy consumption and criterion validity is increased, because the robot executes a different set of trajectories ("vertical movement" the TCP of the robot is moving along the *z*-axis).

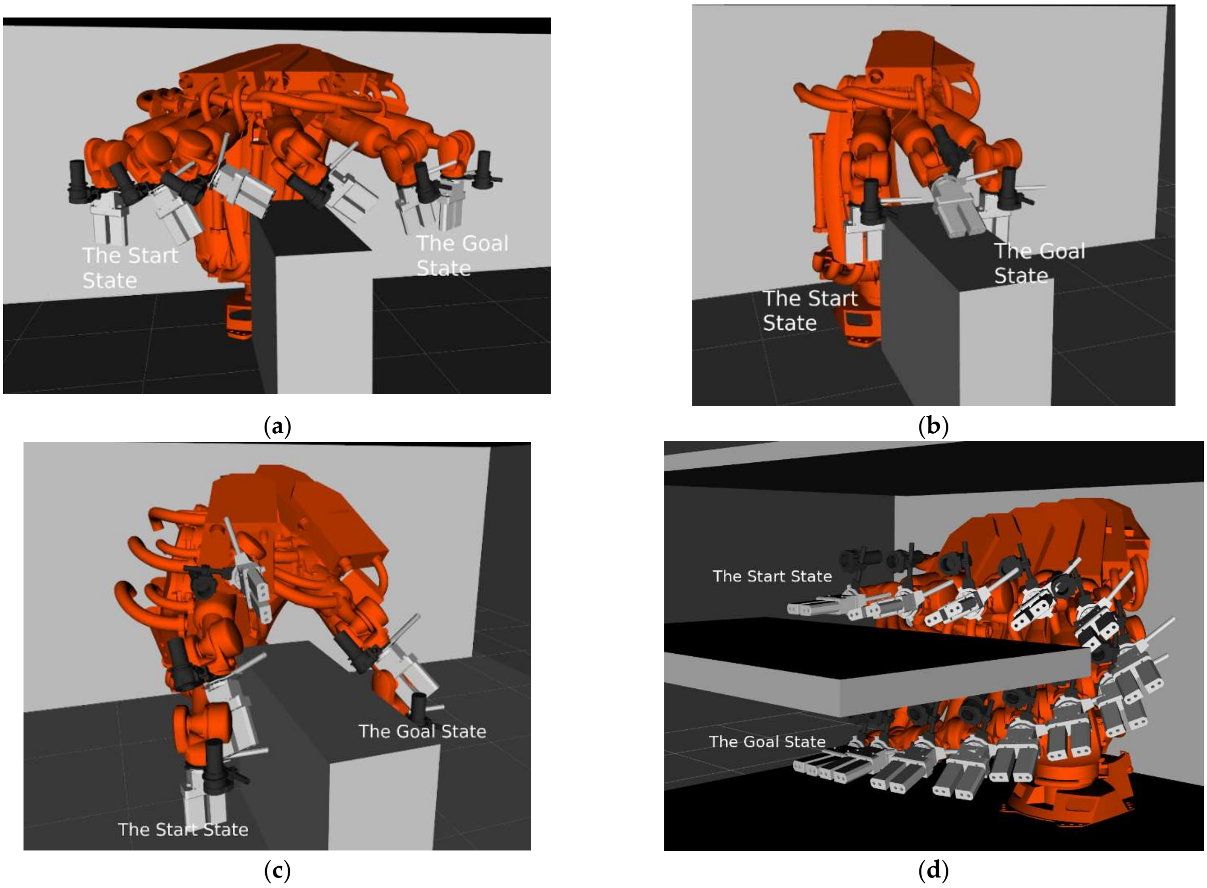


Figure 5. Illustration of one trajectory, which is executed in experiments 2–5: (**a**) experiment no. 2 the "longer" trajectory (**b**) experiment no. 3. the "shorter" trajectory; (**c**) experiment no. 4 the TCP is further on the *x*-axis trajectory; (**d**) experiment no 5. the "vertical" trajectory.

The brute force method for finding of constants requires the following steps:

1. Warm-up of the robot;
2. Generate paths between the same start and the same goal using a path planner;
3. Execute the generated trajectories on a real robot and measure energy consumption;
4. Make cycles through the whole constant space, in which constants are gradually changed:
    a. Compute the criterion score using current constants for each path;
    b. Normalize the measured energy with the criterion score;
    c. If the summary of differences between measured energy and criterion score on each trajectory is lower than the minimum (from previous cycles), store this value as a new minimum.

The output from the brute force method are the constants (0.94736842, 0.21052632, 0.42105263, 0.15789474, 0.05263158, 0.21052632) and Figure 6 shows the histogram of differences between the measured energy consumption and the criterion validity.

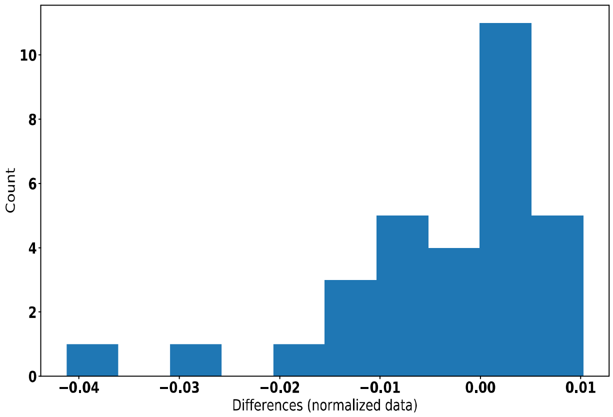


Figure 6. The histogram of differences between predicted and measured energy consumption in experiment 1, which has been used to find control cost constants.

In the next step these constants are validated in experiments 2–5. The results from experiments 2, 3, and 4 are very similar. However, the output from experiment 5 shows an increased deviation, as we anticipated. The results are reported in Figure 7. Figure 8 shows a comparison of results on the boxplot, and finally, the variances and averages for each experiment are compared in Table 2.

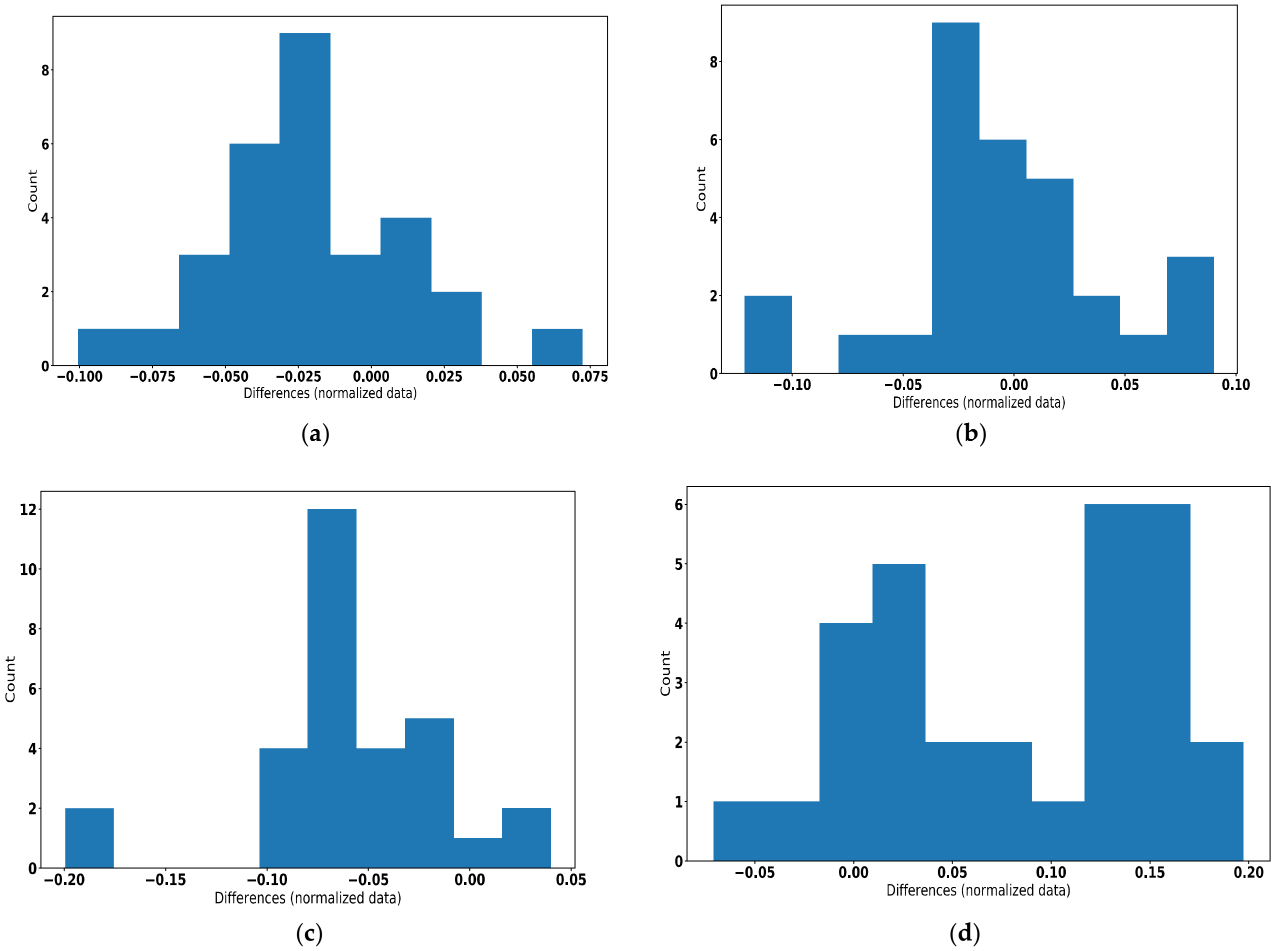


Figure 7. Histograms of the differences between predicted and measured energy consumption: (**a**) experiment no. 2; (**b**) experiment no. 3; (**c**) experiment no. 4; (**d**) experiment no. 5.

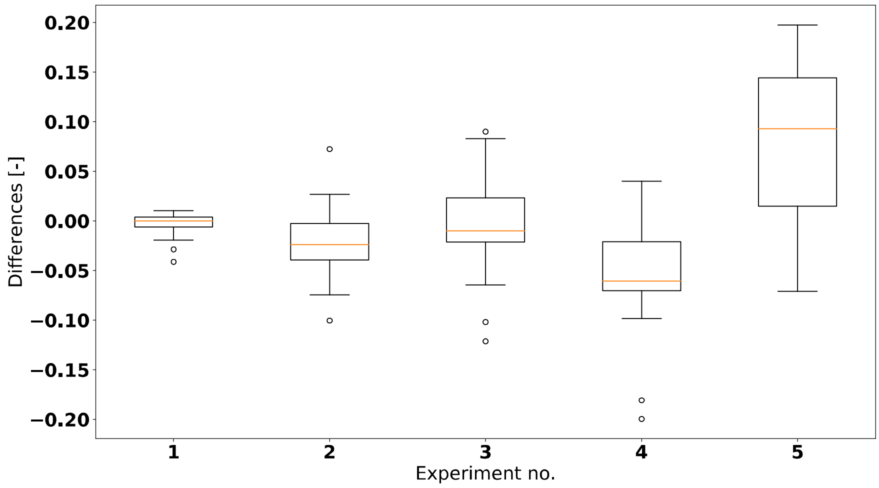


Figure 8. Boxplot comparing the results from Experiments 1–5.

Table 2. Averages and variances for each experiment.

Experiment No.	Average	Variance
1	-3.38×10^{-3}	1.25×10^{-4}
2	-2.20×10^{-2}	1.14×10^{-3}
3	-3.58×10^{-3}	2.28×10^{-3}
4	-5.67×10^{-2}	2.42×10^{-3}
5	8.08×10^{-2}	5.25×10^{-3}

3.2. Jerk-Based Criteria

The path is planned in the joint space of the robot where the path planner generates a set of waypoints. The motion between these waypoints is dependent on the robot controller. We expect that the waypoints are correctly distributed and that the path is always collision-free. Since the path planner only generates positions the jerk is unknown. It can be calculated as the third derivation of the position. This calculation returns an estimated jerk between two waypoints (Figure 9). However, it is not precisely a jerk because the planned path is not sampled in time, and therefore the unit cannot be rad/sˆ3 but just rad. Let us denote this jerk as a pseudo jerk. For the purpose of criterion validation, let us use the same calculation on the trajectories performed by the real robot. However, there is a drawback with this approach, namely that the measured positions are sampled in time (Figure 10). Due to the difference in sampling method and the density of waypoints the pseudo jerk calculated from the planned path and the jerk measured from the robot trajectory differ.

The real jerk is firmly dependent on the robot controller and the control parameters. In this experimental setup, we are using the KUKA KR120 robot and the KRC4 controllers. The motion command PTP_SPLINE is used; its explanation is available in the KUKA reference manual [32]. The velocity was set to 100% of the maximum speed for each joint, while acceleration was limited to 40% of maximum acceleration. The aim is to try to find a correlation between the completed movement and the expected jerk derived from the planned path. Our dataset consists of 30 different paths with different pseudo jerks.

Figure 11 shows pseudo jerk between each waypoint, calculated as joint jerk criterion by Formula (6). Between each waypoint the jerk is the Euclidean distance of the third derivation of joint positions. Therefore, the jerks are non-negative.

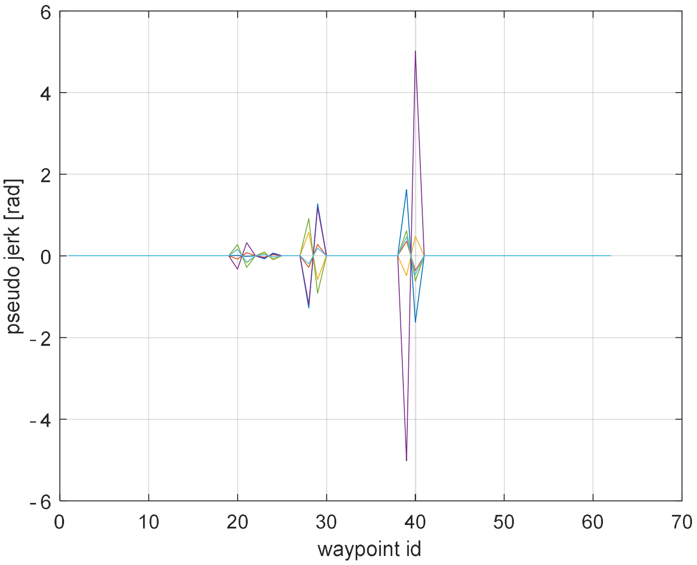


Figure 9. Pseudo jerk calculated from the planned path.

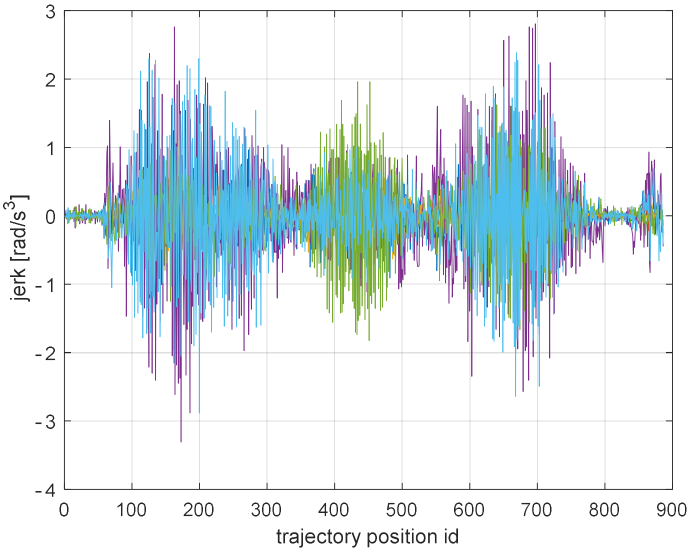


Figure 10. Jerk calculated from positions on the executed trajectory.

Interestingly, the jerk computed from measured robot positions does not correlate to the pseudo-jerk defined on the planned path. However, this highlights another aspect of the robot's motion. In each peak of a pseudo jerk (Figure 11) there is a notable change in the density of the measured robot positions around the planned waypoint. This is indicative of the evolution of the robot's velocity. Where the density of positions is higher, the robot's movement slows down and conversely, sparser density of positions signifies that the robot was moving faster.

Let us denote $X(i)$ as a count of measured robot positions around the waypoint in the planned path. When the $X(i)$ is calculated for each waypoint, then the minimum is denoted as MIN_DENSITY, and we expect that the velocity is the highest in this segment. Let us define the function:

$$f(i) = \frac{X(i)}{MIN\_DENSITY} \tag{10}$$

Function (10) represents a relation between $X(i)$ and $MIN\_DENSITY$, as shown in Figure 12. This function defines the notion of robot "slowdown" and the higher the value of the function, the higher the rate of robot "slowdown".

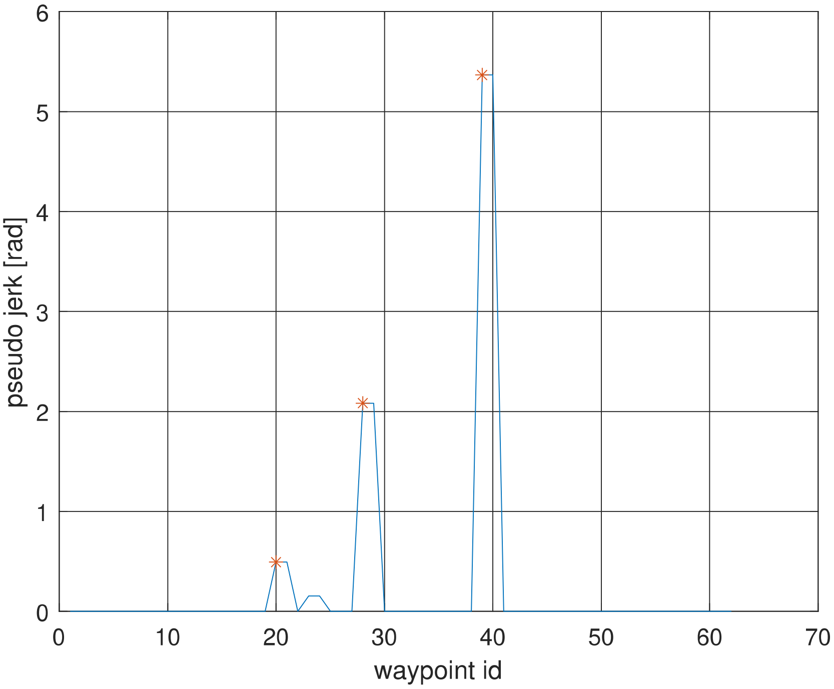


Figure 11. The calculated pseudo-jerk between each waypoint with marked local peaks.

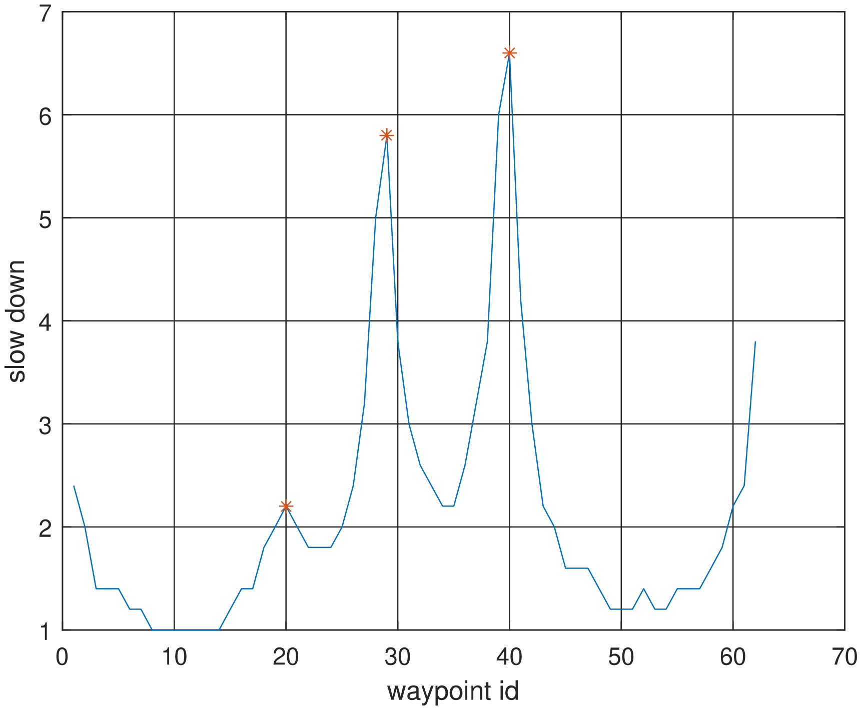


Figure 12. The "slowdown" of the robot with marked local peaks.

Our next goal is to find a relation between the calculated jerk and the robot "slowdown". The first step is to select peaks (local maximal jerks) on each path and assign a waypoint to each of these peaks. For the selection of peaks, a threshold of 0.4 rad is used, and the peaks are located where the derivation of the jerk is zero. The threshold has been found to be empirical on a series of datasets, but it can be modified depending on the indented application. When using a lower threshold the criterion is more sensitive to finding peaks. On the other hand, if the threshold is set to higher value the criterion picks out only relevant higher jerks. These peaks with their appointed waypoint id can then be assigned a value derived from Function (10) from Figure 11. Figure 13 below shows the selected peaks (blue dots) and the correlated robot "slowdown". This set of peaks is approximated by the function:

$$y = 3\, log_{10}\, x + 4 \tag{11}$$

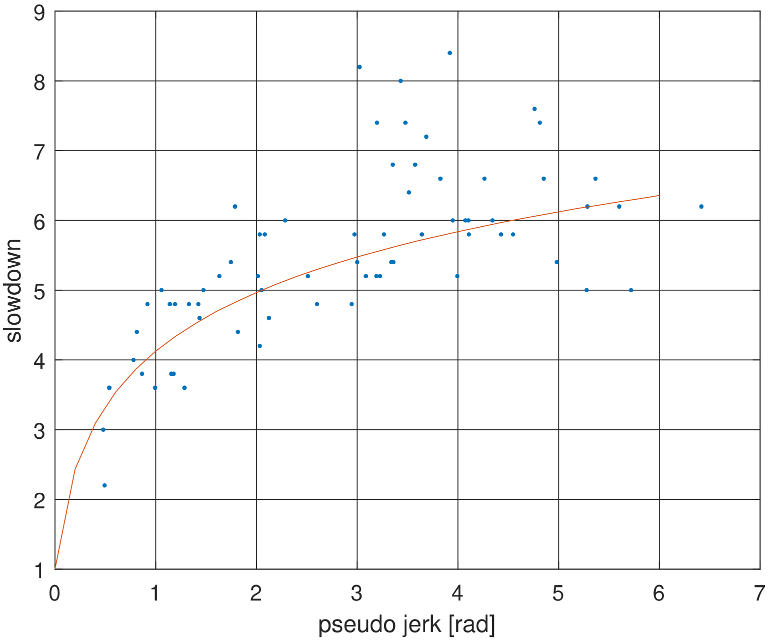


Figure 13. The dependency of the pseudo-jerk and real robot "slowdown".

The x stands for the pseudo-jerk peak on the planned path and y is the robot "slowdown" in the joint space. The function and the constants have been found by regression analysis. The function describes the correlation between pseudo-jerk and the robot "slowdown," which is rendered as a curved red line in Figure 13. Figure 14 shows the differences between the calculated and actual robot "slowdown."

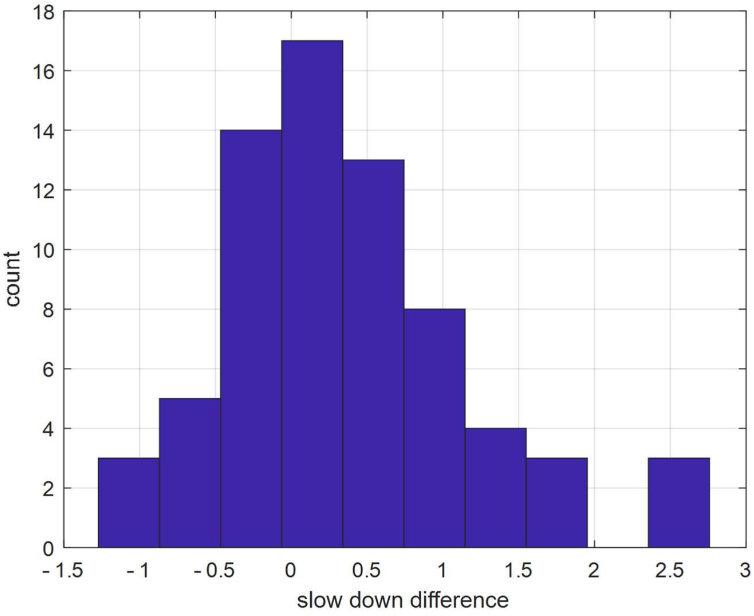


Figure 14. The histogram of differences between the predicated (function (11) and the curved red line in Figure 13) and the real measured (blue dots on Figure 13) robot "slowdown".

A similar approach is utilized for the Cartesian jerk criterion, where the Cartesian pseudo-jerk peaks were found using a threshold 0.002 m.

$$y = 1000\frac{\sqrt{2}}{2}\sqrt{x} + \sqrt[3]{4} \tag{12}$$

In Function (12) above the x stands for a Cartesian pseudo-jerk on the planned path and y is the "slowdown" of the tool point in Cartesian space. The correlation between the pseudo-jerk and the robot "slowdown" in Cartesian space rendered as a curved red line in Figure 15. Figure 16 shows differences between the predicated and real measured values.

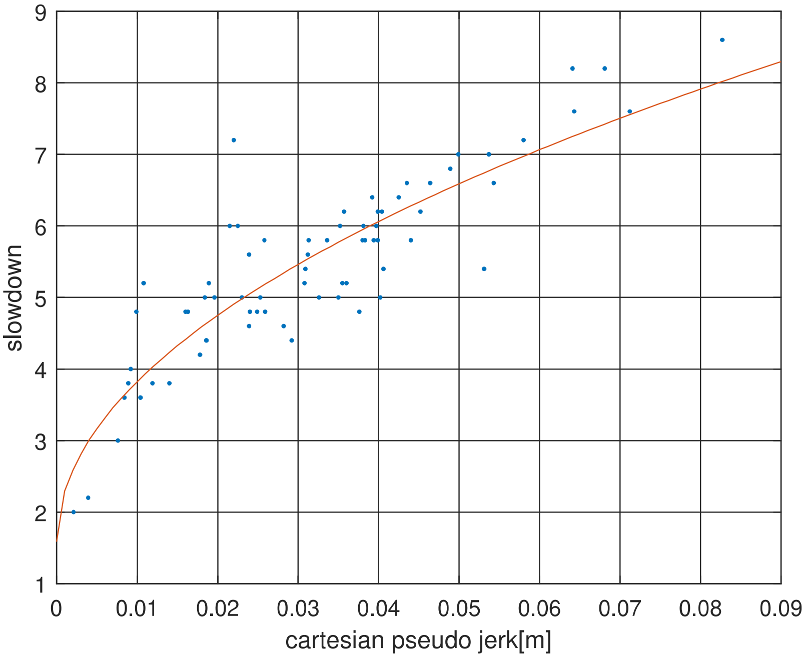


Figure 15. The dependency of the expected Cartesian pseudo-jerk and real robot "slowdown" in Cartesian space of the tool point.

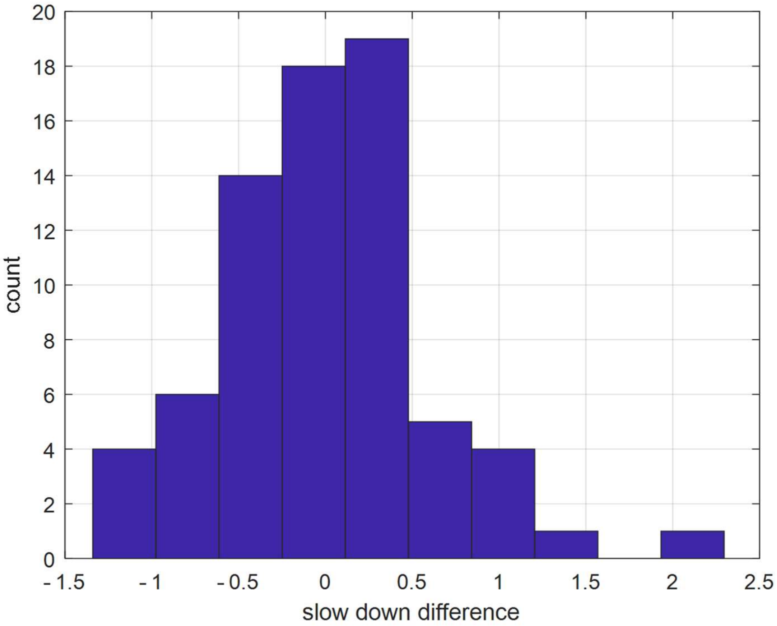


Figure 16. The histogram of differences between the predicated (function (12) and the curved red line in Figure 15) and the real measured (blue dots on Figure 15) robot "slowdown".

Our results show that the joint jerk, joint max jerk, Cartesian jerk, and Cartesian max jerk criteria cannot be compared with real jerk. For this reason, the above criteria were modified and will from now on be considered under the umbrella term joint jerk peaks. This new criterion still computes pseudo jerks, however it looks only for local pseudo jerk peaks. The score of this criterion equals the sum of all peaks, once Equations (11) and (12) have been run. The joint jerk peaks is defined as:

$$x = \sum_{i=1}^{n} 3 \ \log_{10} P_i + 4 \tag{13}$$

and the Cartesian jerk peaks is defined as:

$$x = \sum_{i=1}^{n} 1000 \frac{\sqrt{2}}{2} \sqrt{P_i} + \sqrt[3]{4} + 4 \tag{14}$$

Both in (13) and (14) i represents the peak id of n local pseudo jerk peaks and P_i is the peak value. Additionally, these modified criteria allowed us to compare whether the duration of the executed trajectory correlates with the jerk predicated in the planned path.

4. Criteria Usage

Now that the criteria are validated, our next step is a demonstration of the criteria being utilized in evaluating paths in different environmental setups. This step will be tested on a real robot. We will assess the impact an environment change will have on the path planner. Paths are generated by the STOMP path planner with the same configuration as in the section "Validation of criteria" [4]. In this section, however, we use an additional criterion "Duration", which represents measurement of computation time. The criterion "Duration" does not evaluate the quality of the path, nevertheless an assessment of the path planner's computation time is key when evaluating the impact caused by a change in the environment. Our hypothesis is that a complicated environment requires longer computation time.

4.1. The Impact of the Environment's Complexity on the Path Planner

The first setup simulates a common process in the industry—a pick and place process. The path will be computed between two robot states (−0.527, 0.498, 0.244, −0.103, 0.983, −0.462) (Figure 17a) and (0.465, 0.451, 0.312, 0.097, 0.965, 0.405) (Figure 17b), in both of which the robot tool is located inside a bin and the path planner must return a collision-free path.

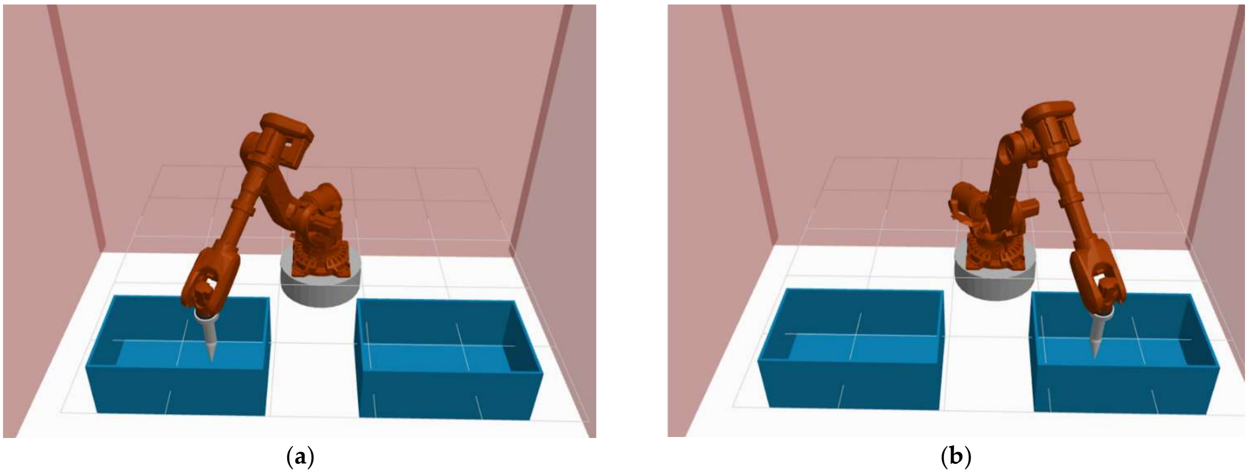
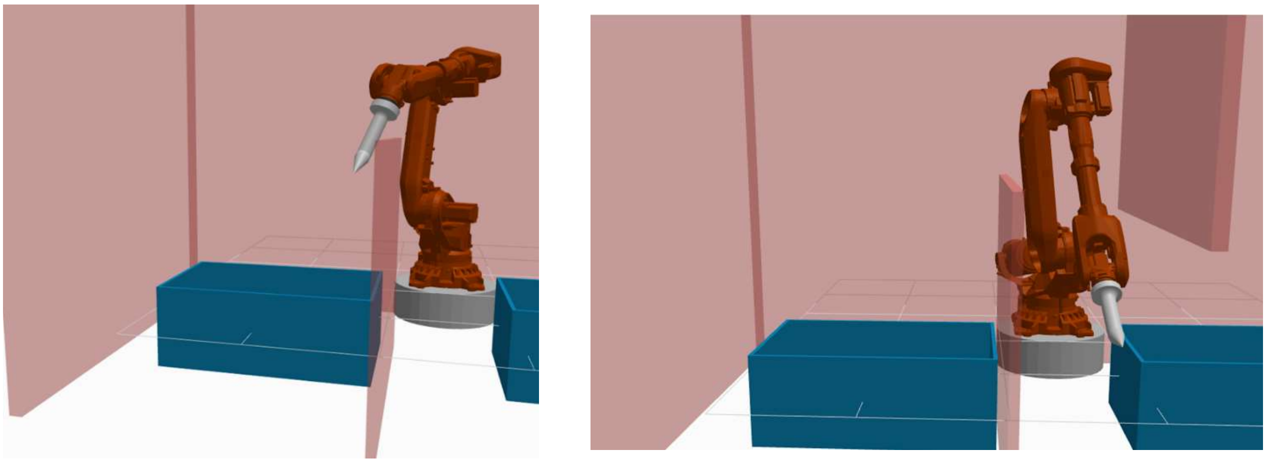


Figure 17. Illustration of the first experimental setup and robot states: (**a**) the start state; (**b**) the goal state.

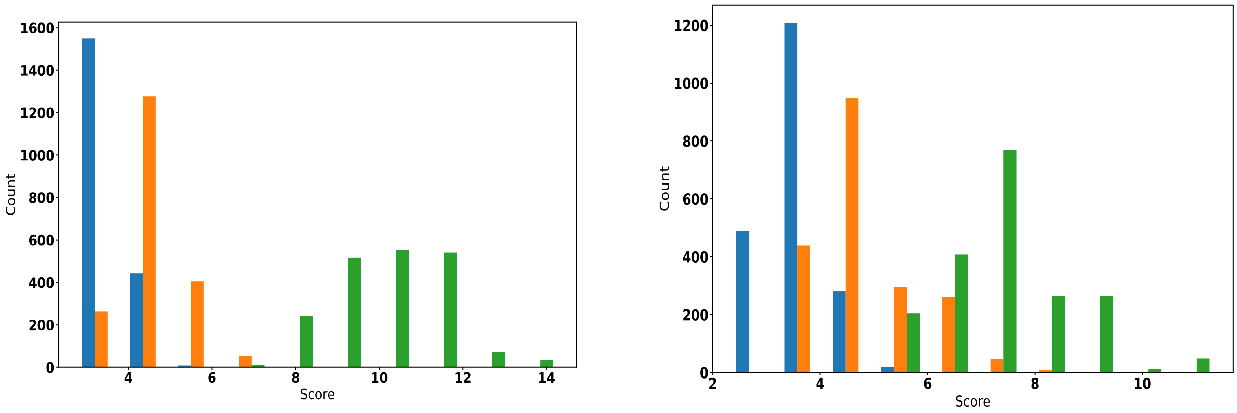
In the second setup, the environment is more complicated. The robot must avoid an added simple wall embedded between the bins from the previous setup (Figure 18a). This situation is a non-standard situation in industry. However the goal of these experimental setups is to monitor the impact on path quality when the planner must avoid more obstacles on the scene. The third setup exhibits the worst traversability, as there is an extra wall inserted (Figure 18b). Start and goal states are the same as in the first setup.



(**a**) (**b**)

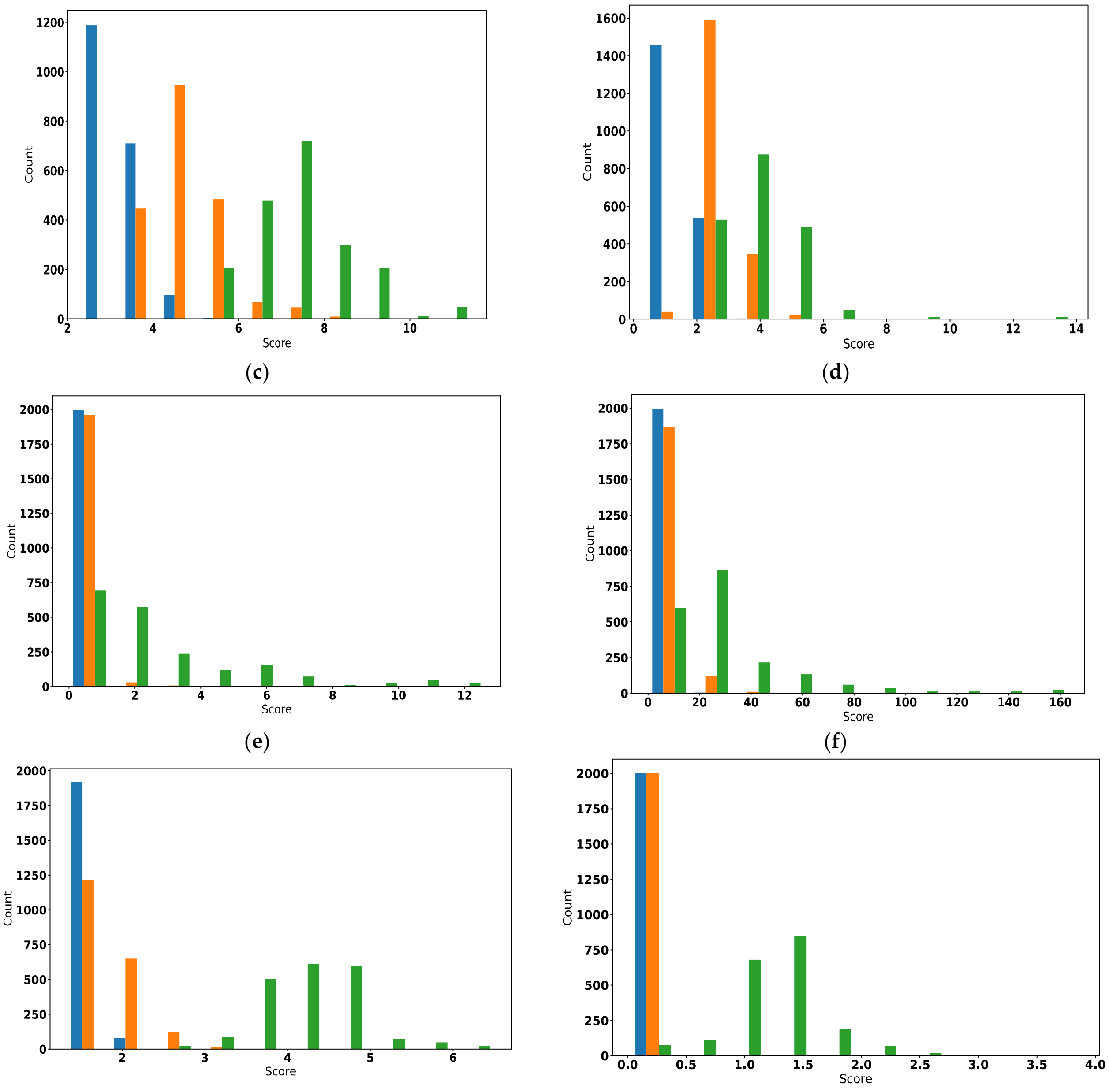
Figure 18. Experimental setups in which extra obstacles are added: (**a**) the second experiment; (**b**) the third experiment.

Our proposed criteria evaluate the environment complexity. Control pseudo-cost criteria use the same constants, calculated in Section 3.1 “Control Pseudo-Cost”. In Section 3.2. “Jerk-based Criteria,” the threshold for the criterion Joint Jerk Peaks 0.4 rad could not determine any peaks in experiments. Therefore, the threshold was decreased to 0.08 rad in these experimental setups, which means that the criterion is more sensitive to finding pseudo-jerk peaks. In the case of Cartesian jerk peaks, the threshold of 0.002 m is preserved. The histograms in Figure 19a–h show the score of each criterion. The results are described in the Section 5 “Results”.



(**a**) (**b**)

Figure 19. *Cont.*



(**c**) (**d**)

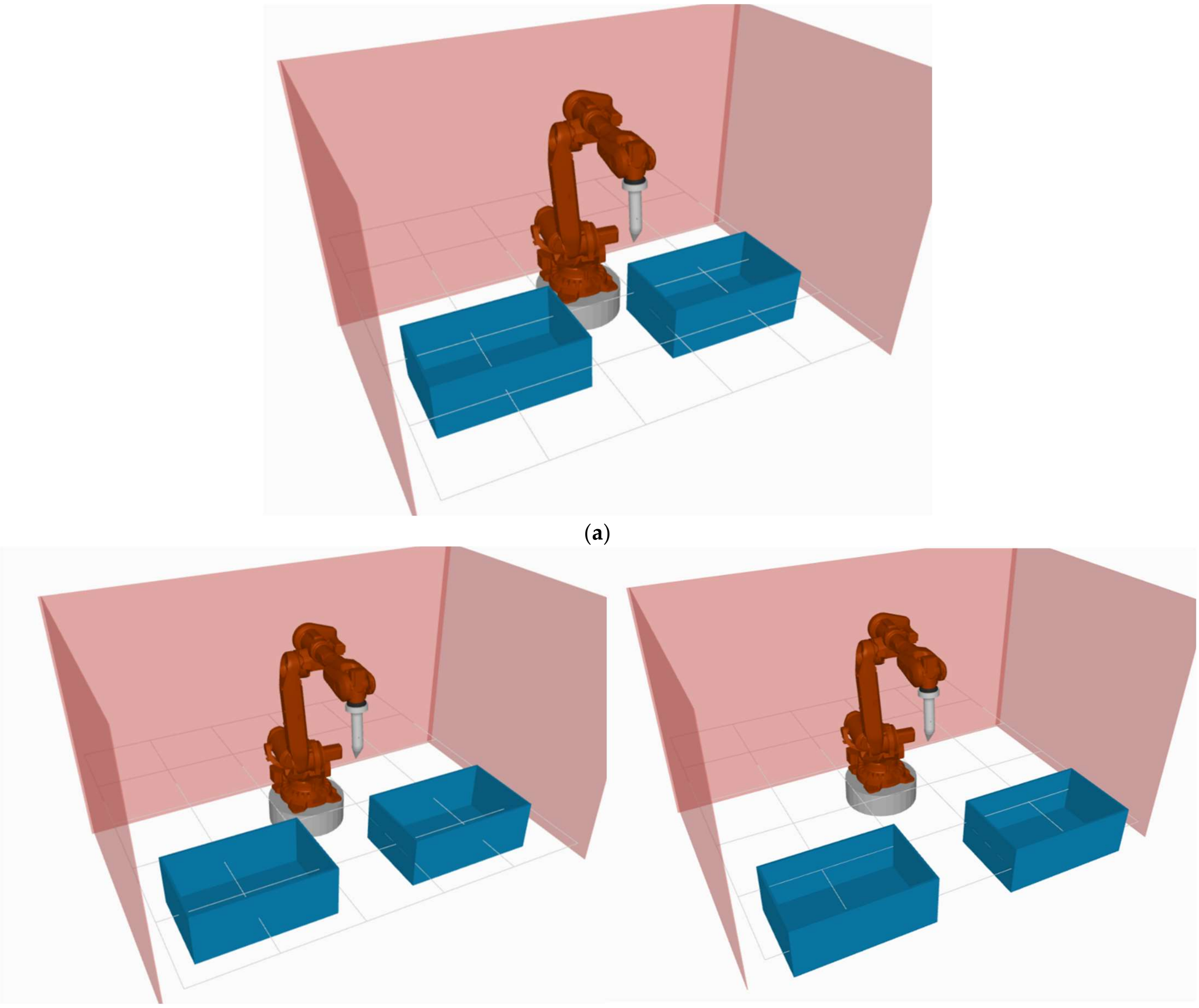
(**e**) (**f**)

(**g**) (**h**)

Figure 19. Histograms depicting the score for each criterion (blue—no.1, orange—no.2, green—no.3): (**a**) joint distance; (**b**) Cartesian distance; (**c**) robot displacement; (**d**) orientation change; (**e**) joint jerk peaks; (**f**) Cartesian jerk peaks; (**g**) control pseudo-cost; (**h**) duration.

4.2. The Impact on the Path Planner of the Bin's Position

This experimental setup is focused on the bin position and the behavior of the path planner as well as how the quality of the generated paths is evaluated in different workspaces. The movement is similar to the first experimental setup and the robot moves from bin A to bin B. The experiment consists of three scenarios, and in each scenario, the bins are reallocated on the *x*-axis of the robot's coordinate system. The layout of each scenario is illustrated in Figure 20a–c.



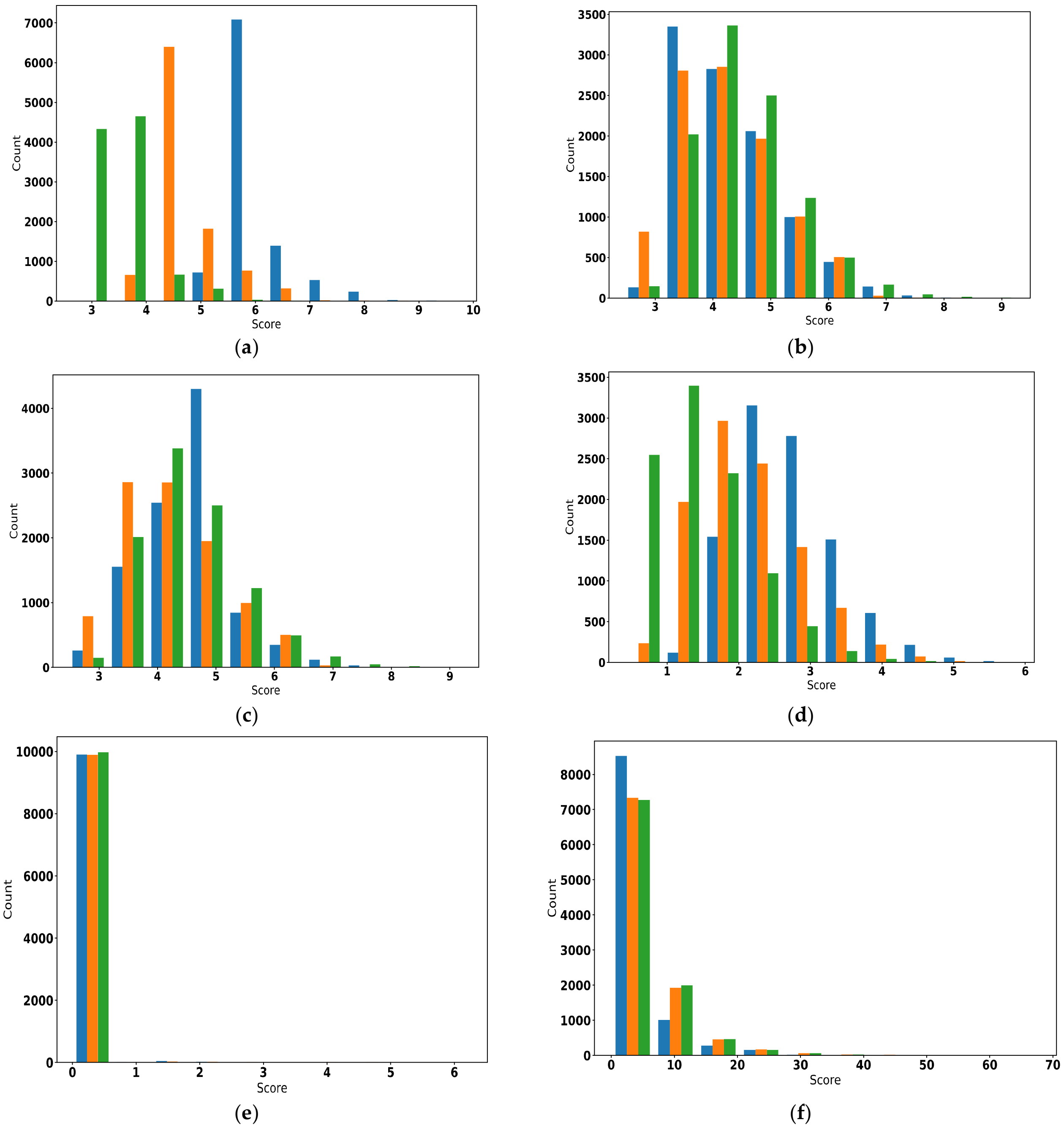
(**a**)

(**b**)

(**c**)

Figure 20. Experimental setups: (**a**) the first setup; (**b**) the second setup; (**c**) the third setup.

Our criteria evaluate the impact of the change in the layout and the impact of the bins' position. Control pseudo-cost, joint jerk peaks, and Cartesian jerk peaks use the same constants and parameters as were used in the previous section of this research paper. The histograms in Figure 21a–h show the score of each of these criteria. The results are described in Section 5 "Results".

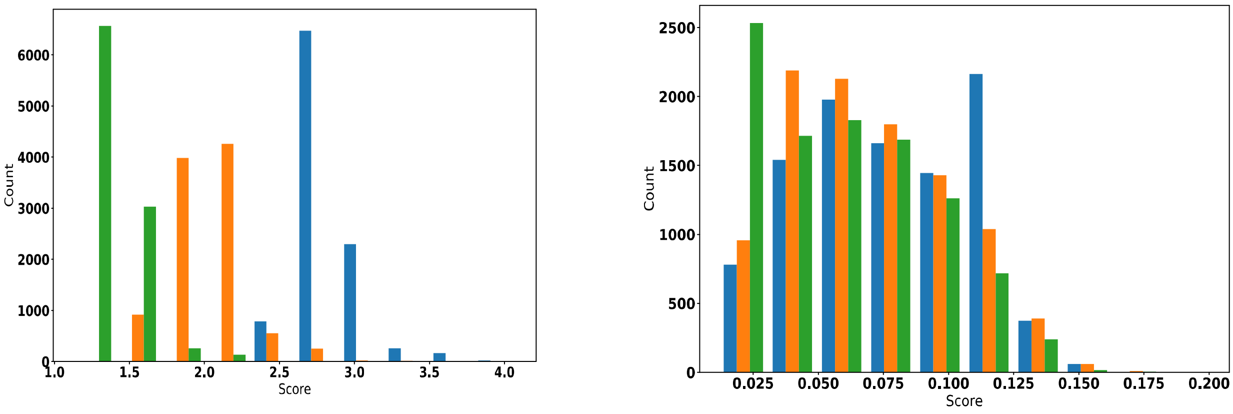


(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 21. *Cont.*



(**g**) (**h**)

Figure 21. Histograms depicting the score for each criterion (blue—no.1, orange—no.2, green—no.3): (**a**) joint distance; (**b**) Cartesian distance; (**c**) robot displacement; (**d**) orientation change; (**e**) joint jerk peaks; (**f**) Cartesian jerk peaks; (**g**) control pseudo-cost; (**h**) duration.

5. Results

The criteria joint distance, Cartesian distance, orientation change, and robot displacement are based on the robot positions within the planned path. To validate these, they were compared with the positions on trajectory executed by a real robot. The differences between the planned and executed trajectories are recapitulated in Table 3, lines 1–4, where the averages and variances for each criterion respectively are shown. These four criteria correlate and we can, therefore, accept them as valid.

Table 3. Recapitulation of criteria validation.

Criterions	Average	Variance
Joint Distance [rad]	-5.75×10^{-3}	3.76×10^{-4}
Cartesian Distance [m]	-4.09×10^{-3}	2.22×10^{-5}
Orientation Change [rad]	-1.24×10^{-2}	3.09×10^{-3}
Robot Displacement [m]	-3.75×10^{-3}	2.52×10^{-5}
Control Pseudo Cost [1]	-2.20×10^{-2}	1.14×10^{-3}
Control Pseudo Cost [2]	8.08×10^{-2}	5.6×10^{-3}
Joint Jerk Peaks [rad]	0.3344	0.8290
Cartesian Jerk Peaks [m]	5.1×10^{-3}	0.6036

[1] The results of experiment no. 2 in Section 3.1. [2] The results of experiment no. 5 in Section 3.1.

The criterion control pseudo-cost is not directly compared with the measured robot positions, however the KUKA robot program provides the measurement of energy consumption as each trajectory is being executed. Firstly, the control pseudo-cost constants of each robot link have been calculated using the brute force method on one dataset consisting of 30 different trajectories carried out between the same start and goal states. As demonstrated on other datasets, the constants and the criteria are applicable to similar movements. The results of the second dataset are recapitulated in Table 3 below. In the table, line 5 depicts the average and the variance of differences between the consumed energy and the criterion score. Notably, the average and the variance increased significantly in line 6 when the robot was performing different movements. For more detailed results from further datasets, consult the section "Control Pseudo-Cost" above.

It was found that the jerk-based criteria—joint jerk, joint max jerk, Cartesian jerk, and Cartesian max jerk do not correlate with the executed trajectory. However we observed a different correlation, namely that the pseudo-jerk peaks calculated on a planned path corresponded to a higher density of measured robot positions in time. This correlation is discussed in more depth in the section "Jerk-Based Criteria". While the joint jerk calculates the sum of jerks along the whole path and the joint max jerk uses just one maximum jerk for the entire path, the experiments proved it would be more advantageous as combined criteria, termed joint jerk peaks and Cartesian jerk peaks respectively. These two criteria can be defined by (13) in joint space and by (14) in Cartesian space and their goal is to find all local pseudo-jerk peaks. The statistical indicators of the differences between the function's output and the jerks calculated from the measured robot positions are shown in Table 3, in lines 7 and 8.

These criteria are applied, and their usage is demonstrated in Section "Criteria Usage" in two experimental setups. In the first experiment three different environments with increasing levels of complexity are compared, and the presented criteria are used to evaluate the planned paths generated to navigate these environments. As expected, the simplest environment achieves the best scores while with increasing complexity, i.e., more obstacles, we see the scores dropping. Naturally, the third and the most challenging environment has the worst scores. In several attempts, the path planner fails to find a collision-free path. The scores are compared and recapitulated in Table 4 below.

Table 4. Recapitulation of the impact of environment complexity per each criterion.

	Criterion	Experiment No.1		Experiment No.2		Experiment No.3	
		Average	Variance	Average	Variance	Average	Variance
1	Joint Distance [rad]	3.699	0.1511	4.9824	0.51754	10.456	1.544
2	Cartesian Distance [m]	3.6277	0.2882	5.107	0.7206	7.6689	1.3505
3	Orientation Change [rad]	1.6659	0.199	2.8288	0.3863	4.6727	1.64228
4	Robot Displacement [m]	3.6315	0.2800	5.1134	0.7182	7.753	1.307
5	Control Pseudo Cost	1.6349	0.0208	2.0629	0.0870	4.3652	0.3606
6	Joint Jerk Peaks [rad]	0.0022	0.0103	0.0575	0.1551	2.824	9.454
7	Cartesian Jerk Peaks [m]	2.42	10.5699	6.7953	60.34	31.2399	943.78
8	Duration [s]	8.02×10^{-2}	4.5×10^{-4}	8.6×10^{-2}	9.2×10^{-4}	1.44	0.2

In the second experimental setup, the behavior of the path planner and the quality of the generated paths are evaluated in different workspaces. On the surface, the robot performs a movement similar to that in the first experimental setup. The difference in this case comes in the position of the bins which are moved further along the X-axis of the robot's coordinate system in each scenario. The goal is to evaluate the impact of this change on the resulting paths. Each experiment has a 100% success rate of path computation. The Cartesian distance and robot displacement criteria have the same results in each scenario, and the position of bins does not change the length of the Cartesian path. The same result can be seen for the duration criterion, which means that the computation time does not dependent on the change in the reachable position of the bins. However, the criterion joint distance shows that the trajectories are longer when the bin is placed closer to the robot base. This behavior is echoed by the outcomes observed for the orientation change criterion. They display a frequent change in the orientation of the tool point in the scenario where the bins are placed closer to the robot base. The scores for the criteria joint jerk peaks and Cartesian jerk peaks prove that the change of the bin position has no impact on path jerks. Our results show that most of the paths have no significant jerk, and are therefore mostly smooth. These results are recapitulated in Table 5.

Table 5. Recapitulation of the impact of bin position per each criterion.

	Criterion	Experiment No.1		Experiment No.2		Experiment No.3	
		Average	Variance	Average	Variance	Average	Variance
1	Joint Distance [rad]	6.2858	0.3452	4.9181	0.4432	3.729	0.3086
2	Cartesian Distance [m]	4.797	0.730	4.328	0.7535	4.737	0.897
3	Orientation Change [rad]	2.974	0.4137	2.2203	0.518	1.804	0.368
4	Robot Displacement [m]	4.810	0.707	4.337	0.7435	4.7405	0.894
5	Control Pseudo-Cost	2.9122	0.0518	2.0887	0.0584	1.4711	0.0431
6	Joint Jerk Peaks [rad]	0.049	0.1156	0.0651	0.1909	0.0148	0.0347
7	Cartesian Jerk Peaks [m]	5.2656	30.308	5.43026	47.276	4.8859	38.443
8	Duration [s]	0.082	0.0011	0.078	0.0011	0.0719	0.0010

6. Discussion

In this paper, we presented a complex trajectory-evaluation framework based on seven criteria. Firstly, we introduced nine hypotheses, mostly based on well-known trajectory metrics. However, following the outcome of our experiments and subsequent modifications, we formulated a total of seven criteria to be used in comparison of trajectories. Each hypothesis was meticulously tested on a real robot, and thus proved to be a valid criterion. There is abundant research into the planning time and success rate of path planners which are viewed as important benchmarks for many industrial applications. Our research, however, focused on the quality of the path planners' output, namely a path defined as a set of robot states. When looking at the minimization of trajectory length, there are several criteria that need to be considered. A well-known approach is to assess the length of a trajectory by finding the sum of joint differences, as demonstrated in [8]. Similarly, [9] employs the tactic to compute the cumulative sum of the tool point's Euclidean distances between each waypoint. Therefore, these criteria (joint distance, Cartesian distance, orientation change, and robot displacement) were included in our trajectory-evaluation framework and were assessed experimentally using a real robot. We confirmed our expectations that these criteria are valid, given that the planned trajectories correspond with the robot-executed trajectories (Table 1). The aim of the research was to consider advanced metrics, energy consumption being one of those. The hypothesis we formulated posits that it is possible to predict which trajectory consumes more energy. Our findings show that by selecting a combination of weights for the control pseudo-cost criterion (4) we can approximate the energy consumption. We investigated the correlation between this criterion's score and the measured consumed energy during the trajectory execution on the real robot. The results confirmed this hypothesis: the higher the criterion's score, the more energy a given trajectory will consume. However, it is important to note that the set of weights is only suitable for sets of similar movements (Experiments no. 1–4 in Section 3.1). The difference between real consumed energy and the criterion's score becomes a less reliable prediction tool in case of significantly different movements (Experiment no. 5 in Section 3.1). This difference is clearly illustrated in a box plot in Figure 8. Another key metric to consider is the smoothness of a trajectory, as smooth motion puts less stress on mechanical units. Smoothness can be computed as the accumulated sum of linear and angular accelerations of the end effector, as presented in [7]. An optimal time-jerk trajectory planning, as suggested in [26], minimizes jerk which is defined as the cumulative sum of time-derivation of accelerations on each joint. Unlike the aforementioned studies, we presented a different approach of trajectory comparison. Our approach considers only robot positions, neglecting properties such as time-parametrization, velocities or accelerations, which are all unknown. We propose that many path planners generate trajectories primarily as positions only, while velocities or accelerations may not necessarily be computed. With this hypothesis in place we formulated the criteria joint jerk, joint max jerk, Cartesian jerk, and Cartesian max jerk which were all based on the computation of the pseudo-jerk. However, the experiment detailed in Section "Jerk-Based Criteria" showed that these criteria are not suitable for trajectory comparison without the trajectories being time-parametrized. Figures 9 and 10 demonstrate the lack of correlation between pseudo-jerk and real jerk. However, we were able to observe a different phenomenon. The local peaks of the pseudo-jerk (Figure 11) showed correlation with observed robot "slowdowns" (Figure 12). Therefore, we designed new criteria, the joint jerk peaks and Cartesian jerk peaks which can be used as a prediction tool for which trajectory will take more time to be executed. Finally, we demonstrate the possible applications of these criteria in Section 4 "Criteria Usage". Due to the proven validity of our criteria, these can now be simply applied in trajectory comparison, without the need for executions on a real robot. This presents a major advantage, as trajectories can now be compared in simulation mode, foregoing the need to collect any measurements from real robot performances. Furthermore, we applied these criteria experimentally to investigate what impact does a change in environment have on a path planner. Our assumption was a straightforward one, namely that the generated trajectories will be of a worse quality in a more complex environment (Table 4). In line with our expectations, the most complicated environment (Experiment no. 3) rendered the highest criterion scores, meaning the quality of trajectories here was the worst.

7. Conclusions

Due to the non-deterministic behavior of randomized path planning algorithms, it is difficult to compare the resulting paths between multiple implementations or changes in configuration.

Path planners usually generate paths as sets of robot states in joint space. Trajectory attributes (e.g., velocity and acceleration) could not be calculated by every path planner, but they could be calculated by a robot control system during their execution. In our work, we focused on paths containing only position data. It is important to note that trajectory analysis with regard to dynamics would constitute an entirely different task, as this type of path does not contain all the necessary information.

In this article, we presented a novel approach for the evaluation and comparison of paths computed by randomized path planners. Nine hypotheses, mostly based on well-known trajectory metrics, were suggested and evaluated with experiments. On the basis of the test results, we formulated seven criteria that can be used for the comparison of the computed paths with respect to multiple production demands.

The evaluation of the criteria joint distance, Cartesian distance, orientation change, and robot displacement confirmed that the planned paths correspond with the executed trajectories. We can confirm that these represent a valid metric for path comparison, and they will be valid in all situations except for situations when the robot executes a completely different trajectory than requested.

The criterion control pseudo-cost is used as a prediction of which paths could consume more energy. This criterion simplifies the dynamics of each robot link by replacing it with a simple constant. For every link, these constants have been calculated using the brute force method on a reference dataset, consisting of 30 different trajectories between the same start and goal states. Our assumption was that the constants are applicable to similar motions and for other groups of motions the constants should be recalculated. Two types of experiment were conducted to verify this hypothesis (described in Section 3.1 Control Pseudo-Cost). For experimental evaluation, we measured the real energy consumption of the robot during the trajectory execution and compared it with the predicted results. Our results confirmed this hypothesis. The difference between consumed energy and criterion scores is more accurate in the group of similar motions. The error is increased in the case of significantly different motions.

Criteria based on the jerk, namely joint jerk, joint max jerk, Cartesian jerk, and Cartesian max jerk, cannot be directly compared with the executed robot motions due to the missing information in trajectories such as velocity and acceleration. However, in our experiments, we noticed that the local peaks of the predicted pseudo-jerk correlate with observed “slowdowns” of the robot. For this reason, we designed the new joint jerk peaks criterion for the joint space and its corresponding counterpart for the Cartesian space, Cartesian jerk peaks. These new criteria could be used as a prediction of which path takes more time. This hypothesis was successfully confirmed in our experiments.

The proposed criteria can serve as a tool for deciding which path planner generates better paths. Likewise, the environment complexity has an impact on path planners and the criteria could assist in the decision of which layout is better or which path planner is appropriate for the chosen layout. Our experiments confirmed that if the environment is more complex, the quality of trajectories is worse.

In our future work, we will focus on in-depth comparison of path planners in different production types with the help of the criteria-based framework presented in this article. Furthermore, it must be stated that the criteria joint jerk peaks and Cartesian jerk peaks have been validated just on the one robot model and, therefore, we would like to investigate the behavior of jerk peaks on other robots as well.

Author Contributions: Conceptualization, P.B. and M.D. (Martin Dekan); methodology, M.D. (Martin Dekan); software, M.D. (Martin Dekan) and M.D. (Michal Dobiš); validation, M.D. (Michal Dobiš); formal analysis, M.D. (Martin Dekan) and M.D. (Michal Dobiš); investigation, M.D. (Martin Dekan) and M.D. (Michal Dobiš); resources, P.B.; data curation, M.D. (Michal Dobiš); writing—original draft preparation, M.D. (Michal Dobiš); writing—review and editing, P.B. and F.D.; visualization, M.D. (Michal Dobiš); supervision, P.B.; project administration, P.B.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by VEGA, grant number VEGA 1/0049/20 and by the European Regional Development Fund, Grant Number 313012P386 and by Horizon 2020, Grant Number 824964.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was partially sponsored by Photoneo company. (http://www.photoneo.com, accessed on 16 January 2022). VEGA 1/0049/20 and DIHˆ2 supported this work. Operational Program Integrated Infrastructure created this publication for the project: "Robotic workplace for intelligent welding of small-scale production," code ITMS2014+: 313012P386, and it is co-sponsored by the European Regional Development Fund.).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Kavraki, L.E.; Svestka, P.; Latombe, J.-C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
2. Lavalle, S.M. Rapidly-Exploring Random Trees: A New Tool for Path Planning. 1998. Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.1853 (accessed on 14 February 2022).
3. Zucker, M.; Ratliff, N.; Dragan, A.D.; Pivtoraiko, M.; Klingensmith, M.; Dellin, C.M.; Bagnell, J.A.; Srinivasa, S.S. CHOMP: Covariant Hamiltonian optimization for motion planning. *Int. J. Robot. Res.* **2013**, *32*, 1164–1193. [CrossRef]
4. Kalakrishnan, M.; Chitta, S.; Theodorou, E.; Pastor, P.; Schaal, S. STOMP: Stochastic trajectory optimization for motion planning. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011.
5. Rodriguez, C.; Suárez, R. Comparison of motion planners in an environment with removable obstacles. In Proceedings of the 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg, 08–11 September 2015; pp. 1–7. [CrossRef]
6. Paulin, S.; Botterill, T.; Lin, J.; Chen, X.; Green, R. A comparison of sampling-based path planners for a grape vine pruning robot arm. In Proceedings of the 2015 6th International Conference on Automation, Robotics and Applications (ICARA), Queenstown, New Zealand, 17–19 February 2015; pp. 98–103.
7. Magyar, B.; Tsiogkas, N.; Brito, B.; Patel, M.; Lane, D.; Wang, S. Guided Stochastic Optimization for Motion Planning. *Front. Robot. AI* **2019**, *6*, 6. [CrossRef]
8. De Maeyer, J.; Demeester, E. Benchmarking framework for robotic arc welding motion planning. *Procedia CIRP* **2021**, *97*, 247–252. [CrossRef]
9. Larsen, L.; Kim, J.; Kupke, M.; Schuster, A. Automatic Path Planning of Industrial Robots Comparing Sampling-based and Computational Intelligence Methods. *Procedia Manuf.* **2017**, *11*, 241–248. [CrossRef]
10. Dobiš, M.; Dekan, M.; Beňo, P.; František, D.; Miroslav, K. The Comparison of Motion Planners for Robotic Arms. *J. Control. Eng. Appl. Inform.* **2021**, *23*, 87–94.
11. Chettibi, T.; Lehtihet, H.E.; Haddad, M.; Hanchi, S. Minimum cost trajectory planning for industrial robots. *Eur. J. Mech. A/Solids* **2004**, *23*, 703–715. [CrossRef]
12. Gandhi, P.; Jain, A.K. Optimization Performance of a Robot to Reduce Cycle Time Estimate. *Int. J. Sci. Res. (IJSR)* **2013**, *2*, 357–363.
13. Trnka, K.; Božek, P. Optimal Motion Planning of Spot Welding Robot Applications. *Appl. Mech. Mater.* **2012**, *248*, 589–593.
14. Mumm, J.; Mutlu, B. Human-robot proxemics: Physical and psychological distancing in human-robot interaction. In Proceedings of the 6th International Conference on Human-Robot Interaction—HRI'11, Lausanne, Switzerland, 6–9 March 2011. [CrossRef]
15. Farouki, R.T.; Li, S. Optimal tool orientation control for 5-axis CNC milling with ball-end cutters. *Comput. Aided Geom. Des.* **2013**, *30*, 226–239. [CrossRef]
16. Toh, C.A. study of the effects of cutter path strategies and orientations in milling. *J. Mater. Process. Technol.* **2004**, *152*, 346–356. [CrossRef]

17. Su, H.; Yang, C.; Ferrigno, G.; De Momi, E. Improved Human–Robot Collaborative Control of Redundant Robot for Teleoperated Minimally Invasive Surgery. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1447–1453. [CrossRef]
18. Tölgyessy, M.; Dekan, M.; Duchoň, F.; Rodina, J.; Hubinský, P.; Chovanec, L. Foundations of Visual Linear Human–Robot Interaction via Pointing Gesture Navigation. *Int. J. Soc. Robot.* **2017**, *9*, 509–523. [CrossRef]
19. Cohen, B.; Chitta, S.; Likhachev, M. Search-based planning for dual-arm manipulation with upright orientation constraints. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 24–27 May 2012; pp. 3784–3790. [CrossRef]
20. Lavalle, S.M. *Planning Algorithms*; Cambridge University Press: New York, NY, USA, 2014; pp. 153–206.
21. Gregory, J.; Olivares, A.; Staffetti, E. Energy-optimal trajectory planning for robot manipulators with holonomic constraints. *Syst. Control. Lett.* **2012**, *61*, 279–291. [CrossRef]
22. Mohammed, A.; Schmidt, B.; Wang, L.; Gao, L. Minimizing Energy Consumption for Robot Arm Movement. *Procedia CIRP* **2014**, *25*, 400–405. [CrossRef]
23. Prado, J.; Marques, L. Energy Efficient Area Coverage for an Autonomous Demining Robot. In Proceedings of the ROBOT2013: First Iberian Robotics Conference, Madrid, Spain, 28–29 November 2013; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2014; pp. 459–471. [CrossRef]
24. Carabin, G.; Scalera, L. On the Trajectory Planning for Energy Efficiency in Industrial Robotic Systems. *Robotics* **2020**, *9*, 89. [CrossRef]
25. Vidussi, F.; Boscariol, P.; Scalera, L.; Gasparetto, A. Local and Trajectory-Based Indexes for Task-Related Energetic Performance Optimization of Robotic Manipulators. *J. Mech. Robot.* **2021**, *13*, 1–12. [CrossRef]
26. Huang, J.; Hu, P.; Wu, K.; Zeng, M. Optimal time-jerk trajectory planning for industrial robots. *Mech. Mach. Theory* **2018**, *121*, 530–544. [CrossRef]
27. Galloway, K.C.; Becker, K.P.; Phillips, B.; Kirby, J.; Licht, S.; Tchernov, D.; Wood, R.J.; Gruber, D.F. Soft Robotic Grippers for Biological Sampling on Deep Reefs. *Soft Robot.* **2016**, *3*, 23–33. [CrossRef]
28. Pettersson, A.; Davis, S.; Gray, J.O.; Dodd, T.J.; Ohlsson, T. Design of a magnetorheological robot gripper for handling of delicate food products with varying shapes. *J. Food Eng.* **2010**, *98*, 332–338. [CrossRef]
29. Rubio, F.; Llopis-Albert, C.; Valero, F.; Suñer, J.L. Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory. *Robot. Auton. Syst.* **2016**, *86*, 106–112. [CrossRef]
30. Liu, H.; Lai, X.; Wu, W. Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. *Robot. Comput. Manuf.* **2013**, *29*, 309–317. [CrossRef]
31. Pan, J.; Chitta, S.; Manocha, D. FCL: A general purpose library for collision and proximity queries. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 24–27 May 2012; pp. 3859–3866.
32. KUKA. System Software KUKA System Software 8.3 Operating and Programming Instructions for System Integrators. 2015. Available online: http://www.wtech.com.tw/public/download/manual/kuka/krc4/KUKA%20KSS-8.3-Programming-Manual-for-SI.pdf (accessed on 14 February 2022).