

Article

Online Computation of Time-Optimization-Based, Smooth and Path-Consistent Stop Trajectories for Robots

Rafael A. Rojas ^{1,2,*}, Andrea Giusti ² and Renato Vidoni ¹

¹ Faculty of Science and Technology, Free University of Bozen/Bolzano, Piazza Universita 1, 39100 Bolzano, Italy; renato.vidoni@unibz.it

² Fraunhofer Italia Research, Via A.-Volta 13A, 39100 Bolzano, Italy; andrea.giusti@fraunhofer.it

* Correspondence: rafael.rojas@unibz.it

Abstract: Enforcing the cessation of motion is a common action in robotic systems to avoid the damage that the robot can exert on itself, its environment or, in shared environments, people. This procedure raises two main concerns, which are addressed in this paper. On the one hand, the stopping procedure should respect the collision free path computed by the motion planner. On the other hand, a sudden stop may produce large current peaks and challenge the limits of the motor's control capabilities, as well as degrading the mechanical performance of the system, i.e., increased wear. To address these concerns, we propose a novel method to enforce a mechanically feasible, smooth and path-consistent stop of the robot based on a time-minimization algorithm. We present a numerical implementation of the method, as well as a numerical study of its complexity and convergence. Finally, an experimental comparison with an off-the-shelf stopping scheme is presented, showing the effectiveness of the proposed method.

Keywords: trajectory planning; optimization; robot stopping scheme; stopping trajectory



Citation: Rojas, R.A.; Giusti, A.; Vidoni, R. Online Computation of Time-Optimization-Based, Smooth and Path-Consistent Stop Trajectories for Robots. *Robotics* **2022**, *11*, 70. <https://doi.org/10.3390/robotics11040070>

Academic Editor: Dario Richiedei

Received: 23 May 2022

Accepted: 21 June 2022

Published: 1 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many robotic applications are subject to unforeseen situations that can endanger the robot itself, its environment or people sharing its workspace. When this happens, one possible solution is to eliminate the source of risk, i.e., the robot's motion. Rather than an arbitrary stop trajectory, it is desirable that the cessation of motion is satisfied at least three important characteristics of the original trajectory. First, it must be feasible with respect to the robot's actuator limits. Second, it has to satisfy the design criteria of the original trajectories. Finally, it has to preserve the original path in order to avoid possible collisions.

Among different design criteria for trajectories, the enforcement of smoothness through the minimization of the jerk has a significant place in the robotics community. This criterion goes back to [1,2] and has been reformulated under different settings [3–9]. Minimum jerk motions reduce the controller's tracking error [1], limit the vibratory content and the consequent mechanical wear on the robotic system [6] and successfully model human arm/hand motions [2,10,11]. In addition, it has been proven that these motions improve the human subjective acceptance of the robot [12], showing a potential application in collaborative robotics [8,9].

In the literature, path-consistent stop trajectories have been exploited to develop collaborative robotics applications under the speed and separation monitoring modality, in agreement with the ISO 10218 and ISO/TS 15066 [13–15]. In such a modality, the robot's motion is tuned through a perception system in order to ensure a protective separation distance from the human [16–18]. Non-path-consistent stop trajectories are used in [19,20].

In order to achieve path consistency, the authors of [21–26] implemented different techniques inside the control loop to dynamically choose the linear scaling of the desired trajectory. Multiple safety criteria are also considered in [27] and verified online to formally

ensure path-consistent fail-safe stops. Other applications of path-consistent stopping trajectories from the perspective of optimal control have been proposed for autonomous vehicles in, e.g., [28,29].

Here, we present a novel method for designing path-consistent, smooth stopping trajectories for robotic systems based on a time-minimization algorithm. In addition, we present a numerical formulation for the problem and a numerical analysis of the complexity and unicity of its solution. Finally, an experimental evaluation with a comparison with the off-the-shelf stop scheme implemented by `ros_control` [30] is presented. This work intends to provide a theoretical background, numerical and experimental insights for the future design of safety-rated, real-time, smooth emergency stop systems on top of smooth trajectories. Such systems have applications in traditional and collaborative robotics. The developed approach foresees: (i) planning the stopping trajectory by formulating a time minimization problem where the optimization variable is a smooth parametrization and the constraints are the dynamic feasibility and the smoothness of the stop; (ii) sending the feasible optimal trajectory to the controller which is responsible for actuating the stop.

In literature, minimum time path parametrization approaches have been proposed in [9,31–34]. In particular, the method developed in [32] has been implemented in [35] with the constraints of the ISO 15066. With respect to the literature, the proposed approach shows three main differences and novelties:

- With respect to [9,31–35] we present a simpler formulation that defines a two-parameter ad hoc set of smooth parametrizations. Such a set does not have a fixed end-point along the path as a constraint, allowing us to compute the required point of the curve where the motion stops. Moreover, our approach is compatible with off-the-shelf optimizers, avoiding the need to develop a custom optimizer to account for the special structure of these methods. In addition, we do not construct a piece-wise constant parametrization and provide an analysis of the construction of a fifth-order polynomial parametrization.
- With respect to control-based approaches [20,22–24], the proposed planning perspective ensures the smoothness of the motion and allows us to decouple the design of the controller from the planning of the stopping trajectory. This makes our approach compatible with any controller, which is responsible for tracking the desired stop trajectory and handling issues such as parametric uncertainties and unmodeled dynamics;
- With respect to [28,29], our methodology is computationally less expensive and does not formulate the optimal stopping problem as an Optimal Control Problem. On the contrary, we formulate an optimization problem on a custom set of parametrizations of the path. In doing so, we avoid the implementation of the differential and path consistency constraints, allowing a significant reduction in the size of the optimization problem.

The remainder of this paper is structured as follows. In Section 2, we provide a formulation of the problem we address. In Section 3 we choose a set of smooth parametrizations and provide the means to transform the original problem into an optimization problem in real variables. In Section 4, the details of the numerical evaluation aspects of our proposed solution are presented. In Section 5, we present numerical examples of our approach with a numerical analysis of the time performance and the uniqueness of the solution. Finally, in Section 6, we present an experimental comparison between our approach and an off-the-shelf stopping scheme.

2. Problem Formulation

The following approach considers an arbitrary trajectory of a robot described as a smooth curve $\mathbf{p}: J \rightarrow \mathbb{R}^n$, where $J = [0, t_f]$ such as a minimum jerk one. We are interested in calculating a stopping parametrization $s_{t_i}(t)$ at the instant t_i that will drive the robot smoothly from its state of motion at t_i to rest at T along the path of \mathbf{p} . The new stopping trajectory on the joint space on the interval $[t_i, T]$ will be

$$\mathbf{q} = \mathbf{p} \circ s_{t_i}, \quad (1)$$

where \circ is the function composition operator. We underline that (1) is the traditional path-trajectory decomposition. The excursion of the robot along the path before the stop is given by the value of the parametrization achieved at the stopping time $s_f = s_{t_i}(T)$. We identify the set of parametrizations with the set $C^{k_s}\text{-Diffeo}([t_i, T], [t_i, s_f])$, which is the set of C^{k_s} -diffeomorphisms [9], i.e., invertible C^{k_s} maps from $[t_i, T]$ to $[t_i, s_f]$ with C^{k_s} inverse. We use this notation to emphasize both the requirement of continuity of the parametrization up to its k_s -derivative, and the fact that parametrizations are one-to-one maps. In practice, the one-to-one requirement prevents that the robot goes back along the path. In other words, diffeomorphisms is the formal mathematical name of parametrizations in robots which do not go back along the path.

The stopping procedure must respect two types of constraints. On the one hand, the dynamics of the system are to be taken into account. During the stopping trajectory, it must be such that the generalized force on the i th-coordinate Q_i is feasible, i.e.:

$$\mathbf{Q} = \mathbf{M}(\mathbf{q}) \frac{d^2 \mathbf{q}}{dt^2} + \mathbf{C} \left(\mathbf{q}, \frac{d\mathbf{q}}{dt} \right) \frac{d\mathbf{q}}{dt} + \mathbf{g}(\mathbf{q}) \in \mathcal{Q}, \tag{2}$$

where \mathbf{M} is the mass matrix, \mathbf{C} is the matrix accounting for Coriolis and centrifugal effects, \mathbf{g} accounts for gravity, \mathbf{Q} are the joint torques and \mathcal{Q} is the set of admissible torques.

On the other hand, we desire to limit possible degradation of the smoothness of the original trajectory. We account for this with two complementary requirements: firstly, the resulting trajectory must be continuous up to its k_c -th (≥ 2) derivative; secondly, we introduce a measure of smoothness to be minimized and used for comparison. To do so, we introduce and use two different kinds of smoothness measures:

- The L^2 norm of the jerk, relative to the nominal path

$$\int_{t_i}^T \left\| \frac{d^3 \mathbf{q}}{dt^3} \right\|^2 dt - \alpha \int_{t_i}^{t_f} \left\| \frac{d^3 \mathbf{p}}{dt^3} \right\|^2 dt \leq 0, \tag{3}$$

- The max norm of the acceleration, relative to the nominal path

$$\max_{t \in [t_i, T]} \left| \frac{d^2 q_j}{dt^2}(t) \right| - \alpha \max_{t \in [0, t_f]} \left| \frac{d^2 p_j}{dt^2}(t) \right| \leq 0, \quad j = 1, \dots, n \tag{4}$$

where the term $\alpha > 0$ in (3)–(4) stands for a proportional factor and $\| \cdot \|$ is the euclidean norm.

In order to obtain the system dynamics along the path of \mathbf{p} during the stopping procedure, we have to substitute the following relations in (2)

$$\begin{aligned} \frac{d\mathbf{q}}{dt} &= \frac{d\mathbf{p}}{dt} \frac{ds_{t_i}}{dt} \\ \frac{d^2 \mathbf{q}}{dt^2} &= \frac{d^2 \mathbf{p}}{dt^2} \left(\frac{ds_{t_i}}{dt} \right)^2 + \frac{d\mathbf{p}}{dt} \frac{d^2 s_{t_i}}{dt^2}. \end{aligned} \tag{5}$$

We can conclude that our problem is that of finding a stopping smooth parametrization $s(t)$ capable of driving the robot from its nominal state of motion at the instant t_i to rest as fast as possible considering the constraints commented above. In other words,

$$\begin{aligned} \min_T & \int_{t_i}^T dt \\ \text{s.t.} & \quad (2) \text{ and either (3) or (4)} \\ & \quad \dot{s}(T) = 0 \\ & \quad \mathbf{q} \in C^{k_c}([0, T], \mathbb{R}^n) \\ & \quad s \in C^{k_s}\text{-Diffeo}([t_i, T], [t_i, s_f]), \end{aligned} \tag{6}$$

where the s.t. means “subject to”.

Finally, we underline that our approach is intrinsically path-consistent. In fact, because we directly design the parametrization of the path, it is not possible for our approach to produce positions that are outside the nominal path.

3. Smooth Stopping Parametrization

In order to obtain smooth stopping, we constrain s_{t_i} to the set of functions which minimize the following integral

$$\int_{t_i}^T \left(\frac{d^3 s_{t_i}}{dt^3} \right)^2 dt. \tag{7}$$

To simplify our notation, we introduce the following change of variable

$$\tau(t) = \frac{t - t_i}{T_s}, \tag{8}$$

where $T_s = T - t_i$ is the time the robot takes to stop.

The change of variable (8) transforms (7) into

$$\int_0^1 \left(\frac{d^3 \bar{s}_{t_i}}{d\tau^3} \right)^2 d\tau, \tag{9}$$

where \bar{s}_{t_i} is the parametrization of s_{t_i} in τ , i.e.,

$$s_{t_i} = \bar{s}_{t_i} \circ \tau(t). \tag{10}$$

The main advantage of formulating our problem in terms of \bar{s}_{t_i} instead of s_{t_i} , is that \bar{s}_{t_i} has a constant domain $[0, 1]$ and s_{t_i} a domain $[t_i, t_i + T_s]$ which varies along the optimization. Thanks to this, we can formulate the constraints of our problem in fixed points in the interval $[0, 1]$.

It is known that the minimizers of (9) are of the form [8]

$$\bar{s}_{t_i}(\tau) = \sum_{k=0}^5 a_k \tau^k. \tag{11}$$

We underline that by choosing $s_{t_i} = \bar{s}_{t_i} \circ \tau$ we have $k_s = \infty$ and transformed (6) into a constrained optimization problem in real variables, i.e., the coefficients of (11) and T_s . They must hold two different kinds of boundary conditions. On the one hand, they must ensure a smooth transition between the nominal and stopping trajectory. By taking $k_c = 2$ we have

$$\begin{aligned} \mathbf{p}(t_i) &= \mathbf{q}(t_i) \\ \frac{d\mathbf{p}}{dt}(t_i) &= \frac{d\mathbf{q}}{dt}(t_i) \\ \frac{d^2\mathbf{p}}{dt^2}(t_i) &= \frac{d^2\mathbf{q}}{dt^2}(t_i). \end{aligned} \tag{12}$$

On the other hand, they must ensure the stop condition

$$\begin{aligned} \frac{d\mathbf{p}}{dt}(T) &= 0 \\ \frac{d^2\mathbf{p}}{dt^2}(T) &= 0. \end{aligned} \tag{13}$$

After substituting the relations (1), (5) and (10) into (12) and (13), we get the boundary conditions on \bar{s}_{t_i} , which ensures the smooth transition from the nominal to the stop trajectory and the stop condition

$$\begin{aligned} \bar{s}_{t_i}(0) &= t_i \\ \frac{d\bar{s}_{t_i}}{d\tau}(0) &= T_s & \frac{d\bar{s}_{t_i}}{d\tau}(1) &= 0 \\ \frac{d^2\bar{s}_{t_i}}{d\tau^2}(0) &= 0 & \frac{d^2\bar{s}_{t_i}}{d\tau^2}(1) &= 0. \end{aligned} \tag{14}$$

The constraints (14) allow us to define our parametrization as

$$\begin{aligned} \bar{s}_{t_i}(\tau) &= (-3T_s + 6s_f - 6t_i)\tau^5 \\ &+ (8T_s - 15s_f + 15t_i)\tau^4 \\ &+ (-6T_s + 10s_f - 10t_i)\tau^3 \\ &+ T_s\tau + t_i. \end{aligned} \tag{15}$$

Note that (15) reduces the number of variables of our problem to two, namely the time to stop T_s and s_f . Another advantage of (15) is that it allows us to write the diffeomorphism in terms of these variables, as follows. Firstly, we note that the parametrization constraints read as follows [9]

$$\frac{ds}{dt} > 0. \tag{16}$$

Secondly, as we desire to cease the robot’s motion, we restrict our attention to those parametrizations that slow down the robot with the following constraint

$$\frac{d^2s}{dt^2} < 0. \tag{17}$$

Finally, we note that (17) implies (16). In fact, if the velocity becomes negative, it has to be constrained to accelerate in order to respect (14). As a consequence, the constraint (17) account for both the diffeomorphism and deceleration constraints. In our variables (T_s and s_f), (17) reads

$$\begin{aligned} 3T_s - 5s_f + 5t_i &> 0 \\ -2T_s + 5s_f - 5t_i &> 0 \end{aligned} \tag{18}$$

We underline that the implementation of (16) alone would require an analysis of the roots of a fourth-order polynomial and the subsequent implementation of non-linear constraints.

4. Numerical Implementation

In order to approximate the constraints as well as the required integrals, we take a sufficient dense grid of Gauss–Lobatto points $\{t_j\}_{j=1}^{n_c} \subset [t_i, T]$ [36,37]. Thus, we enforce (2) at these points and approximate (3)–(4) as follows

- The L^2 norm of the jerk, relative to the nominal path

$$\left[\sum_{j=1}^{n_c} w_i \left\| \frac{d^3\mathbf{q}}{dt^3}(t_j) \right\|^2 \right] - \alpha \int_{t_i}^{t_f} \left\| \frac{d^3\mathbf{p}}{dt^3} \right\|^2 dt \leq 0, \tag{19}$$

where w_i are the Gauss–Lobatto weights [37].

- The max norm of the acceleration, relative to the nominal path

$$\left| \frac{d^2q_k}{dt^2}(t_j) \right| - \alpha \max_{t \in [0, t_f]} \left| \frac{d^2p_k}{dt^2}(t) \right| \leq 0, \quad k = 1, \dots, n \tag{20}$$

Then, the problem (6) may be written as

$$\begin{aligned} \min \quad & T_s \\ \text{s.t.} \quad & (2), (18) \text{ and either } (19) \text{ or } (20). \end{aligned} \tag{21}$$

We implement the presented approach using the GSplines library [38] for curve representation and the IPOPT optimizer [39] using the options in Table 1.

The linear solver ma27 was chosen, as it was faster for this particular problem. The option `fast_step_computation` allows us to avoid fine tuning of the solution from the linear solver intended for larger problems. To introduce in the problem the analytical derivatives while using numerical approximations of the hessian, we set `jacobian_approximation` to `exact` and `hessian_approximation` to `limited-memory`.

Table 1. IPOPT options used for the optimization of (21).

<code>linear_solver</code>	<code>ma27</code>
<code>fast_step_computation</code>	<code>yes</code>
<code>jacobian_approximation</code>	<code>exact</code>
<code>hessian_approximation</code>	<code>limited-memory</code>

5. Convergence and Complexity Analysis

In order to evaluate the convergence and the complexity of our proposed methodology, we follow a numerical approach. To do so, we optimize (21) for a set of randomly generated problems, on the set of minimum jerk trajectories. Each problem is constituted by

1. Either one of the dynamic models of the UR3 from Universal Robots [40] and the robot Franka Emika [41] picked randomly.
2. A random minimum jerk trajectory computed as described in [8].
3. A random desired stopping time t_i .
4. Either one of the constraints (19)–(20) picked randomly.

To test the uniqueness of the solution, we solve (21) for each problem using three different families of initial guesses given by

$$s_f = \frac{3}{5}T_s + t_i \tag{22}$$

$$s_f = \frac{1}{2}T_s + t_i \tag{23}$$

$$s_f = \frac{2}{5}T_s + t_i, \tag{24}$$

with $T_s = \gamma(t_f - t_i)$ for $\gamma \in \{0.01, 0.05, \dots, 1\}$. For each family in (24) and each value of γ we compute the solution (T_s, s_f) of (21). Then, we pick one of these solutions randomly (T_s^*, s_f^*) and compute the error $e = \sqrt{(T_s^* - T_s)^2 + (s_f^* - s_f)^2}$. The unfeasible initial guesses are discarded. We found that, for each problem, all the feasible initial guesses converge to the same solution (T_s^*, s_f^*) . In Figure 1 we observe the convergence of the optimizer for different initial guesses. Here, the iterations of the optimizer for three random problems are reported on the columns. Each color corresponds to a different initial guess. In the top row, we show the iterations in the plane (T_s, s_f) for three initial guesses and depict the constraints (17) with thick black lines. The bottom row shows the convergence as the error e (computed with respect to the unique solution found (T_s^*, s_f^*)) with respect to the number of iterations.

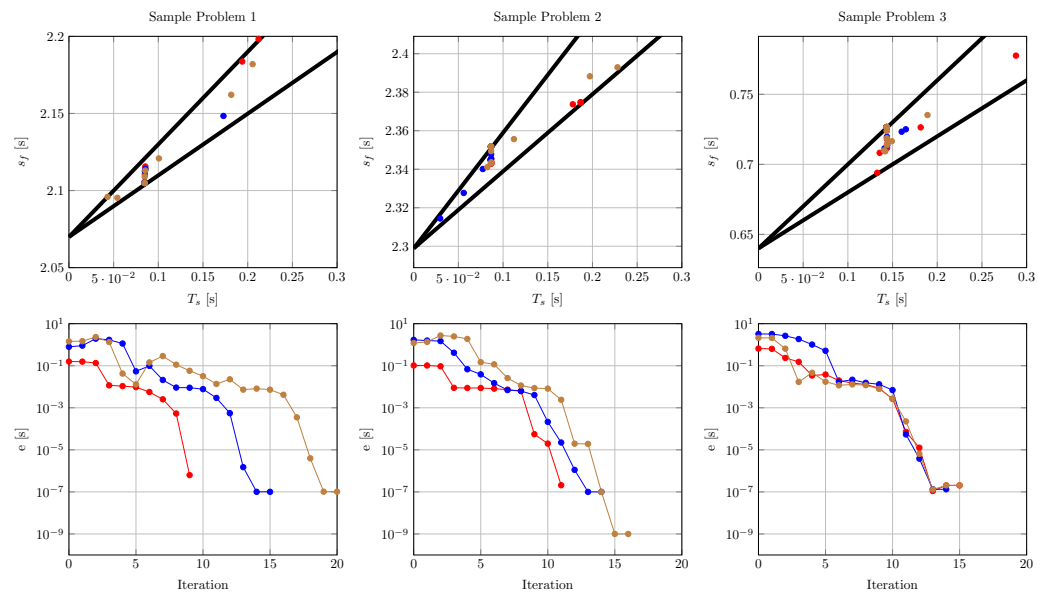


Figure 1. Evolution and convergence of the optimization algorithm for three different initial guesses in three randomly sampled problems. Each color represents a different initial guess.

In Figure 2, we present the cumulative distribution of the time employed by IPOPT to optimize (21) for the random problems described above in different computers. This experiment was performed using the same Docker [42] image in all computers. We achieved convergence for all samples, which took less than 20 ms. Moreover, in the best case, we achieved time performances less than 10 ms for more than 99% of the samples. In Figure 3, we present the cumulative distribution of the number of iterations required by the optimization process.

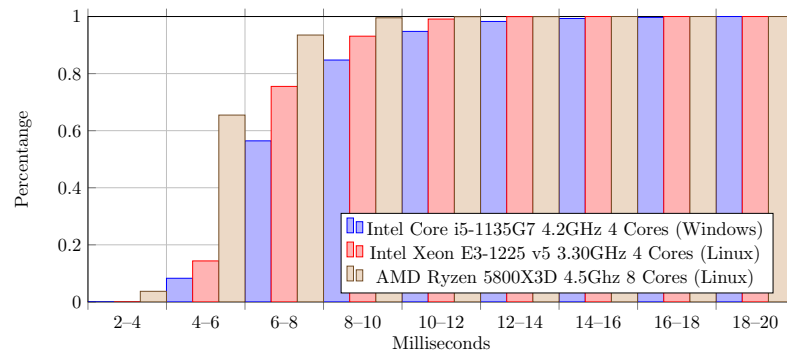


Figure 2. Cumulative distribution of the optimization time employed by different workstations.

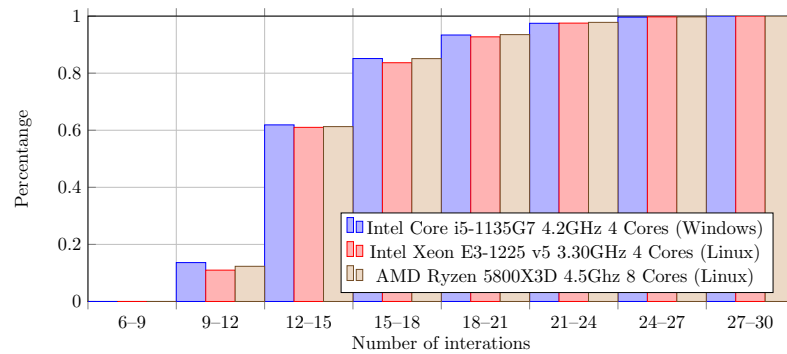


Figure 3. Cumulative distribution of number of iterations required by the optimization problem in different workstations.

6. Experimental Comparison

Our experimental setup is depicted in Figure 4, and consists of a Franka Emika Panda arm controlled with `ros_control` [30] performing a random minimum-jerk trajectory. For the realization of the experiments, we implemented our proposed approach within a ROS package compatible with `ros_control` [43]. The `ros_control` package provides the possibility of regulating the smoothness of this stop procedure through the ROS parameter `stop_trajectory_duration`, which sets the duration of the stopping trajectory. The larger this parameter is, the smoother the motion stop. We call this mechanism the off-the-shelf stop. This controller exposes a ROS action which listens for a complete goal trajectory on the joint space, which is subsequently performed at the specified start time. It also provides two different pre-emption mechanisms that we use to stop the robot. Firstly, when a cancel request is received the control enters in position hold mode and stops the robot by trying to place it in the last known position of the trajectory. Secondly, when a new trajectory goal is received, the controller takes the useful parts of the currently executed and new trajectories and combines them, as described in [30]. This is achieved by considering the desired starting time of the new trajectory and the current time. We used the second mechanism to enforce a path-consistent stop by sending the stopping segment of the trajectory as a new goal, i.e., the segment on the interval $[t_i, t_i + T_s]$ with a starting time at t_i . To take into account the time consumed by the computational effort and the network delays, we considered a time delay d between the time t'_i when the stop signal is received and the time $t_i = t'_i + d$. We chose $d = 60$ ms for our setup.

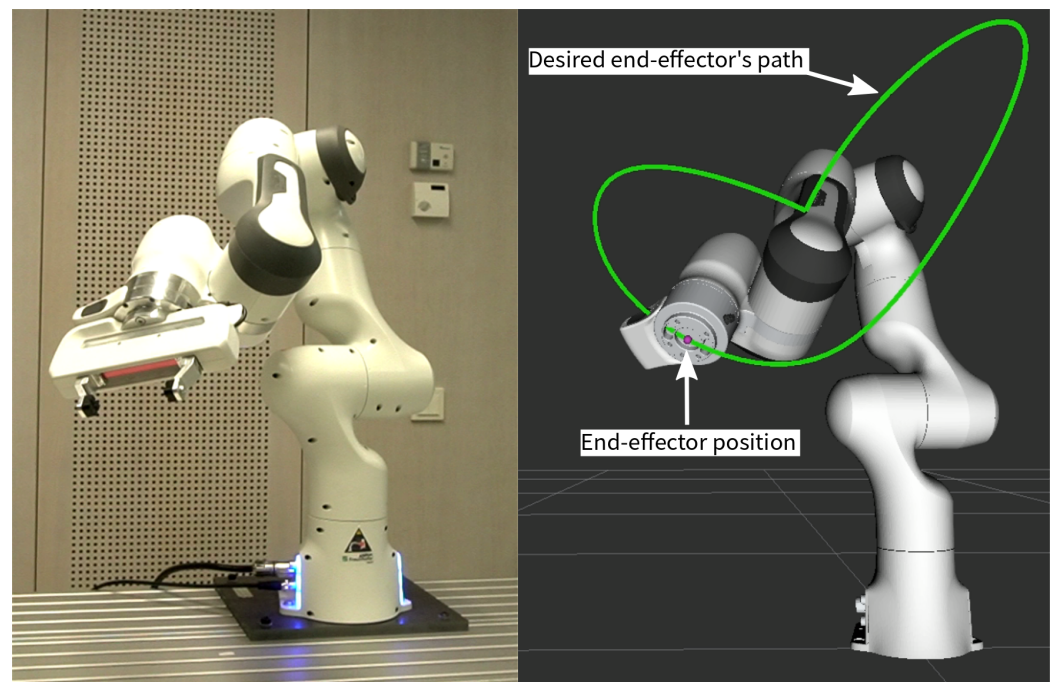


Figure 4. Experimental setup: (left) Robot Franka Emika Panda used in the experiments; (right) example of randomly generated path.

In Figures 5–8, we present two different comparisons between the off-the-shelf stop and our approach with a bound on the acceleration. In all the figures, we depict the desired trajectory in black, the trajectory which stops with off-the-shelf stop in red, our approach with the smoothness degradation constraint (4) in green and our approach with the smoothness degradation constraint (3) in blue. For all experiments, we took $\alpha = 5.0$ for the smoothness degradation constraints (4)–(3). In Figures 5 and 7 we present a comparison in the joint space and the instant when the stop signal is issued is highlighted as a vertical dashed line. Figures 6 and 8 present our comparison in the Cartesian space with the instant when the stop signal is issued highlighted as a black dot.

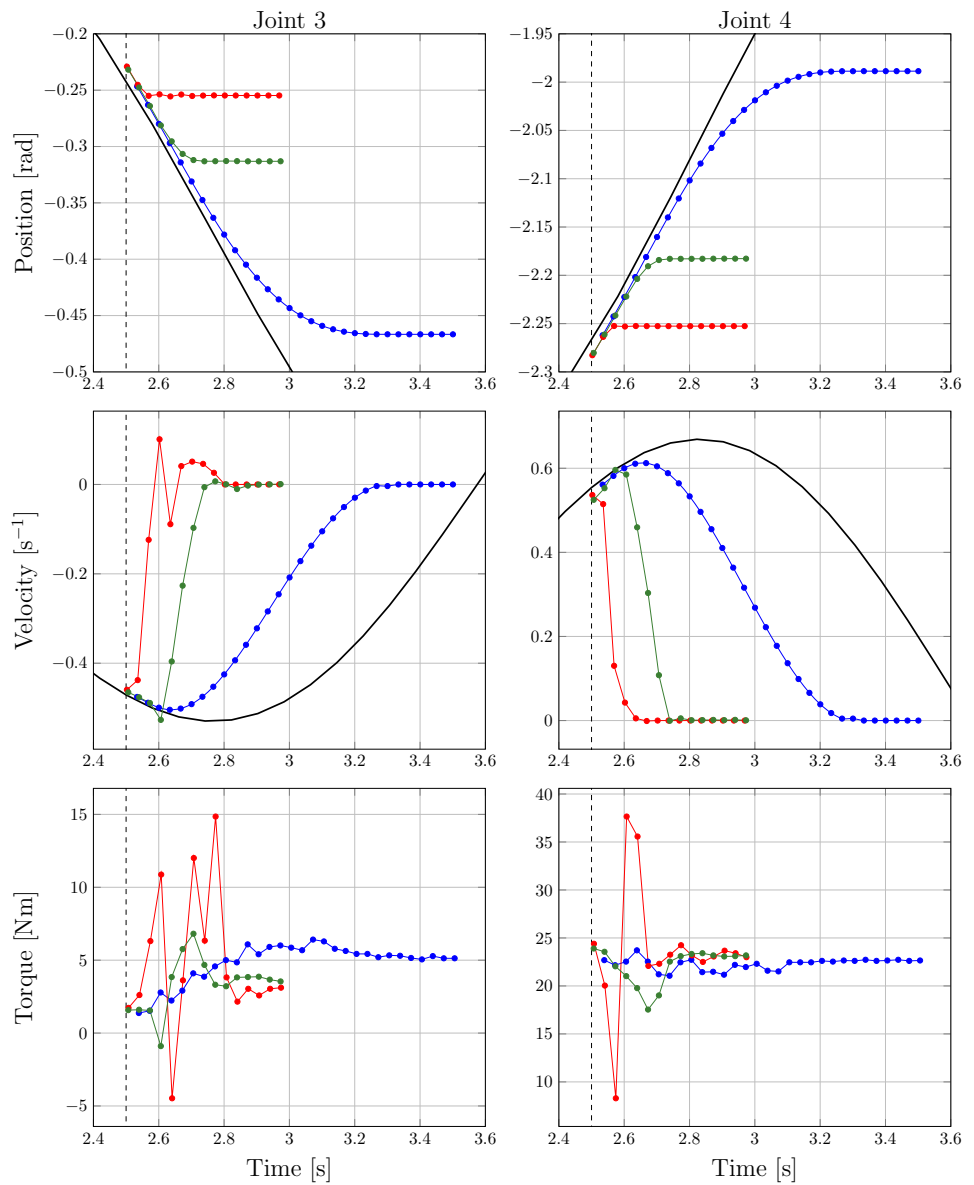


Figure 5. Comparison in the joint space of the off-the-shelf stop (red) with the path-consistent stops with smoothness degradation constraint (4) (green) and (3) (blue). The desired trajectory is depicted in black. The dashed vertical line is the instant when the stop is requested.

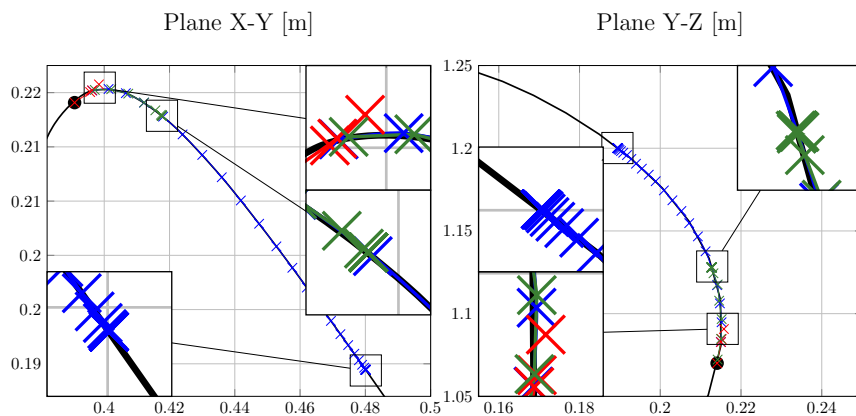


Figure 6. Comparison in the Cartesian space of the off-the-shelf stop (red) with the path-consistent stops with smoothness degradation constraint (4) (green) and (3) (blue). The desired path is depicted in black.

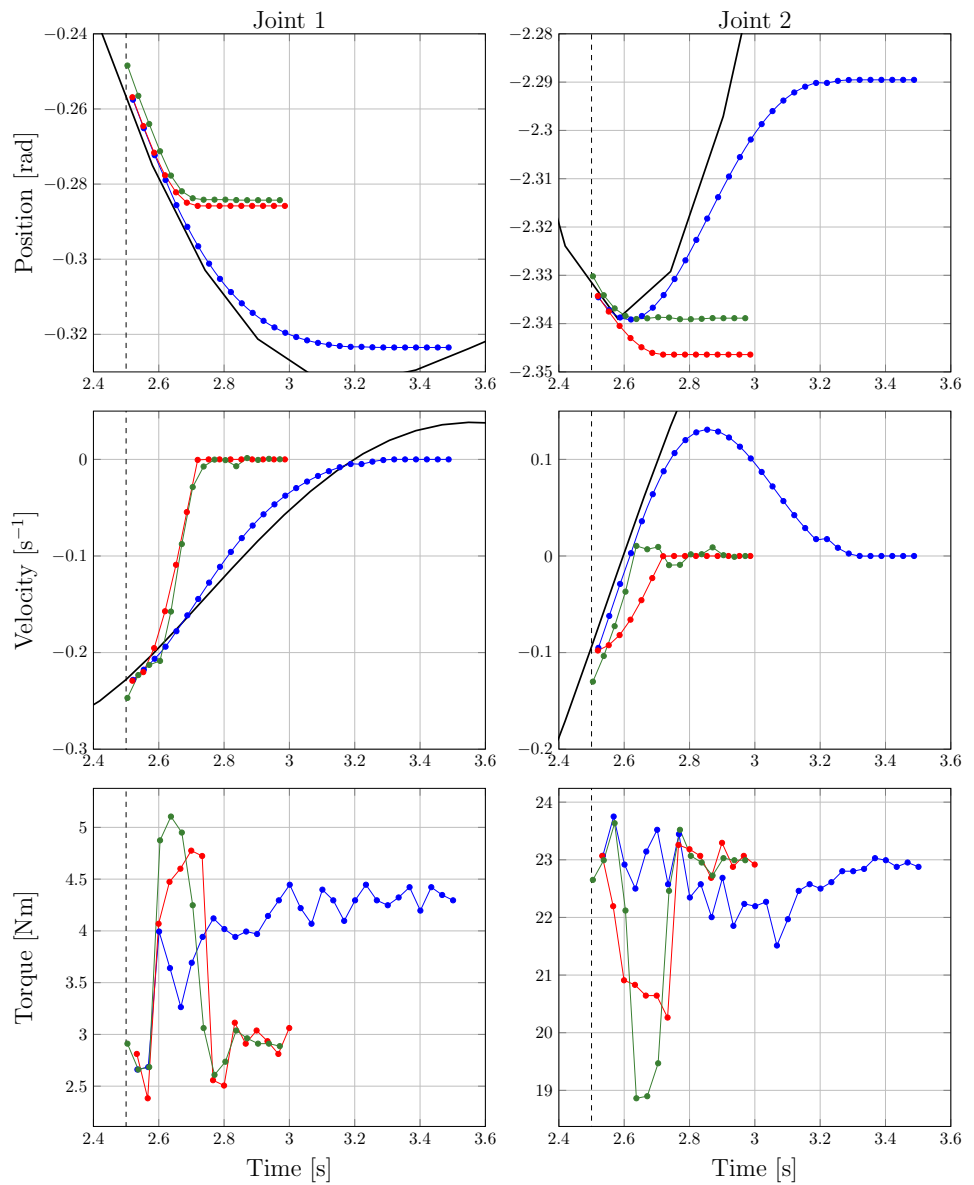


Figure 7. Comparison in the joint space of the off-the-shelf stop (red) with the path-consistent stops with smoothness degradation constraint (4) (green) and (3) (blue). The desired trajectory is depicted in black. The dashed vertical line is the instant when the stop is requested.

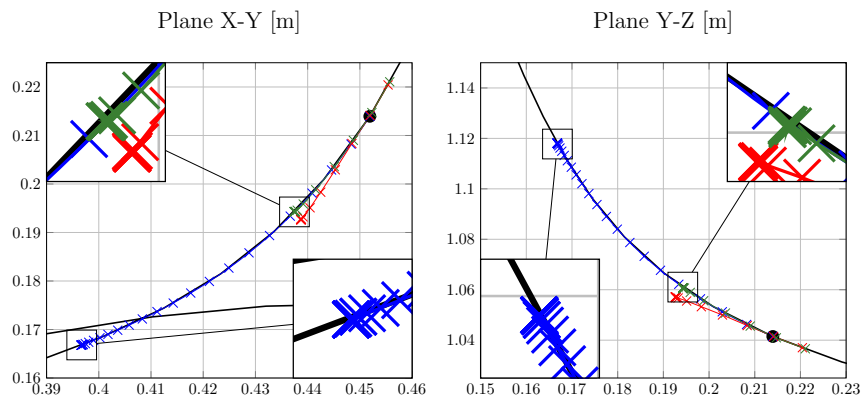


Figure 8. Comparison in the Cartesian space of the off-the-shelf stop (red) with the path-consistent stops with smoothness degradation constraint (4) (green) and (3) (blue). The desired path is depicted in black.

In our first comparison, depicted in Figures 5 and 6, we show that although it is possible to achieve small deviations from the desired path with a traditional stopping mechanism, this comes at the price of large torque variations and subsequent oscillations. Here, we set our off-the-shelf stop to 50 ms and it can be seen in Figure 6 that our proposed approach remains on the desired path. Meanwhile, the off-the-shelf stop presents a small deviation from it. However, in Figure 5, we observe that the off-the-shelf stop shows evident oscillations in the velocity plot and large variation in the torques; meanwhile, our approach is smoother and requires fewer torques. Finally, we note that although our approach implementing (4) shows a larger displacement after the stop signal than the off-the-shelf stop, it takes a similar amount of time to bring the robot to rest.

In our second comparison, depicted in Figures 7 and 8, we show that although it is possible to achieve a smooth stop of the motion with a traditional method, this comes at the price of evident deviations from the desired path. Here, we set our off-the-shelf stop to 200 ms, and it can be seen in Figure 8 that our proposed approach remains on the desired path and the off-the-shelf stop presents an evident deviation from it. However, in Figure 7, we observe that the off-the-shelf stop has a similar smoothness and slightly smaller torques than our approach implementing (4).

In both comparisons, we observe that the stopping trajectory with smoothness degradation constraint (3) is the smoothest, with fewer torque requirements. However, it is also the one with the larger excursion along the path after the stop request. From this, we can conclude that the smoothness degradation constraints (3) is more conservative than (4), which achieves larger torques and shows minor excursions along the path after the stop request. We underline that we can achieve different trade-offs between the smoothness and the time to stop the robot by choosing a different value of α . The final time value, computed by the algorithm presented here, depends on the particular smoothness measure that we wish to preserve.

7. Conclusions and Outlook

In this work, we presented a time-optimization approach capable of stopping a robot while preserving the path of the nominal motion and limiting the smoothness degradation. This was achieved by choosing an a priori set of smooth parametrizations and writing our problem as a constrained optimization problem in two real variables. A numerical analysis showed that our implementation converges to a unique solution in finite time (less than 20 ms) on the set of minimum jerk trajectories. We presented an experimental comparison with the off-the-shelf stop scheme implemented by `ros_control` and showed the effectiveness of our approach.

Future work will focus on the following different aspects. The authors will: (i) deepen the theoretical aspects related to the complexity of the presented approach, as well as evaluating the effects of the α parameter; (ii) implement the approach in both classical and collaborative robotics where emergency and safety stops have to be implemented; (iii) study the possibility of implementing this scheme on general trajectory parametrization problems, such as time minimization.

Author Contributions: Conceptualization, all; methodology and formal analysis R.A.R.; supervision, A.G., R.V.; writing—original draft, R.A.R.; writing—review and editing, all. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Unibz ID project “Automated Process Planning in Cyber Physical Production Systems of Smart Factories” (Acronym: SMART-APP—Cost centre: TN2011).

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kyriakopoulos, K.J.; Saridis, G.N. Minimum jerk path generation. In Proceedings of the International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 24–29 April 1988; pp. 364–369.
2. Flash, T.; Hogan, N. The coordination of arm movements: An experimentally confirmed mathematical model. *J. Neurosci.* **1985**, *5*, 1688–1703. [[CrossRef](#)] [[PubMed](#)]
3. Simon, D.; Isik, C. A trigonometric trajectory generator for robotic arms. *Int. J. Control* **1993**, *57*, 505–517. [[CrossRef](#)]
4. Piazzzi, A.; Visioli, A. Global minimum-jerk trajectory planning of robot manipulators. *IEEE Trans. Ind. Electron.* **2000**, *47*, 140–149. [[CrossRef](#)]
5. Gasparetto, A.; Zanotto, V. A new method for smooth trajectory planning of robot manipulators. *Mech. Mach. Theory* **2007**, *42*, 455–471. [[CrossRef](#)]
6. Gasparetto, A.; Lanzutti, A.; Vidoni, R.; Zanotto, V. Validation of minimum time-jerk algorithms for trajectory planning of industrial robots. *J. Mech. Robot.* **2011**, *3*, 031003. [[CrossRef](#)]
7. Boscariol, P.; Gasparetto, A.; Vidoni, R. Planning continuous-jerk trajectories for industrial manipulators. *Eng. Syst. Des. Anal. Am. Soc. Mech. Eng.* **2012**, *3*, 127–136.
8. Rojas, R.A.; Garcia, M.A.R.; Wehrle, E.; Vidoni, R. A Variational Approach to Minimum-Jerk Trajectories for Psychological Safety in Collaborative Assembly Stations. *IEEE Robot. Autom. Lett.* **2019**, *4*, 823–829. [[CrossRef](#)]
9. Rojas, R.A.; Vidoni, R. Designing Fast and Smooth Trajectories in Collaborative Workstations. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1700–1706. [[CrossRef](#)]
10. Meirovitch, Y.; Bennequin, D.; Flash, T. Geometrical invariance and smoothness maximization for task-space movement generation. *IEEE Trans. Robot.* **2016**, *32*, 837–853. [[CrossRef](#)]
11. Oguz, O.S.; Zhou, Z.; Glasauer, S.; Wollherr, D. An Inverse Optimal Control Approach to Explain Human Arm Reaching Control Based on Multiple Internal Models. *Sci. Rep.* **2018**, *8*, 5583. [[CrossRef](#)]
12. Kühnlenz, B.; Kühnlenz, K. Reduction of Heart Rate by Robot Trajectory Profiles in Cooperative HRI. In Proceedings of the ISR 2016: 47th International Symposium on Robotics, Munich, Germany, 21–22 June 2016; pp. 1–6.
13. ISO. *ISO/TC 299: ISO 10218-1: 2011; Robots and Robotic Devices—Safety Requirements for Industrial Robots—Part 1: Robots*. International Organization for Standardization: Geneva, Switzerland, 2011.
14. *ISO/TC 299: ISO/TS 15066: 2016; Robots and Robotic Devices—Collaborative Robots*. International Organization for Standardization: Geneva, Switzerland, 2016.
15. Haddadin, S. *Towards Safe Robots: Approaching Asimov's 1st Law*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 90.
16. Lacevic, B.; Rocco, P.; Zanchettin, A.M. Safety Assessment and Control of Robotic Manipulators Using Danger Field. *IEEE Trans. Robot.* **2013**, *29*, 1257–1270. [[CrossRef](#)]
17. Zanchettin, A.M.; Ceriani, N.M.; Rocco, P.; Ding, H.; Matthias, B. Safety in human-robot collaborative manufacturing environments: Metrics and control. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 882–893. [[CrossRef](#)]
18. Althoff, M.; Giusti, A.; Liu, S.B.; Pereira, A. Effortless creation of safe robots from modules through self-programming and self-verification. *Sci. Robot.* **2019**, *4*, eaaw1924. [[CrossRef](#)] [[PubMed](#)]
19. Scalera, L.; Giusti, A.; Vidoni, R.; Di Cosmo, V.; Matt, D.; Riedl, M. Application of Dynamically Scaled Safety Zones Based on the ISO/TS 15066:2016 for Collaborative Robotics. *Int. J. Mech. Control* **2020**, *21*, 41–49.
20. Safeea, M.; Neto, P.; Bearee, R. On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case. *Robot. Auton. Syst.* **2019**, *119*, 278–288. [[CrossRef](#)]
21. Pupa, A.; Arrfou, M.; Andreoni, G.; Secchi, C. A Safety-Aware Kinodynamic Architecture for Human-Robot Collaboration. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4465–4471. [[CrossRef](#)]
22. Zanchettin, A.M.; Rocco, P.; Chiappa, S.; Rossi, R. Towards an optimal avoidance strategy for collaborative robots. *Robot. Comput.-Integr. Manuf.* **2019**, *59*, 47–55. [[CrossRef](#)]
23. Merckaert, K.; Convens, B.; Wu, C.J.; Roncone, A.; Nicotra, M.M.; Vanderborght, B. Real-time motion control of robotic manipulators for safe human–robot coexistence. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102223. [[CrossRef](#)]
24. Ferraguti, F.; Bertuletti, M.; Landi, C.T.; Bonfè, M.; Fantuzzi, C.; Secchi, C. A control barrier function approach for maximizing performance while fulfilling to iso/ts 15066 regulations. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5921–5928. [[CrossRef](#)]
25. Glogowski, P.; Lemmerz, K.; Hypki, A.; Kuhlenkötter, B. Extended Calculation of the Dynamic Separation Distance for Robot Speed Adaption in the Human-Robot Interaction. In Proceedings of the 2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2–6 December 2019; pp. 205–212.
26. Byner, C.; Matthias, B.; Ding, H. Dynamic speed and separation monitoring for collaborative robot applications—concepts and performance. *Robot. Comput.-Integr. Manuf.* **2019**, *58*, 239–252. [[CrossRef](#)]
27. Beckert, D.; Pereira, A.; Althoff, M. Online verification of multiple safety criteria for a robot trajectory. In Proceedings of the 56th Annual Conference on Decision and Control (CDC), Melbourne, Australia, 12–15 December 2017; pp. 6454–6461.
28. Wang, L.; Wu, Z.; Li, J.; Stiller, C. Real-Time Safe Stop Trajectory Planning via Multidimensional Hybrid A*-Algorithm. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–7.

29. Svensson, L.; Masson, L.; Mohan, N.; Ward, E.; Brenden, A.P.; Feng, L.; Törngren, M. Safe stop trajectory planning for highly automated vehicles: An optimal control problem formulation. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 517–522.
30. Chitta, S.; Marder-Eppstein, E.; Meeussen, W.; Pradeep, V.; Tsouroukdissian, A.R.; Bohren, J.; Coleman, D.; Magyar, B.; Raiola, G.; Lüdtke, M.; et al. ros_control: A generic and simple control framework for ROS. *J. Open Source Softw.* **2017**, *2*, 456. [[CrossRef](#)]
31. Pham, Q.C. A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *IEEE Trans. Robot.* **2014**, *30*, 1533–1540. [[CrossRef](#)]
32. Shiller, Z.; Lu, H.H. Computation of path constrained time optimal motions with dynamic singularities. *J. Dyn. Syst. Meas. Control* **1992**, *114*, 34–40. [[CrossRef](#)]
33. Pham, H.; Pham, Q.C. A new approach to time-optimal path parameterization based on reachability analysis. *IEEE Trans. Robot.* **2018**, *34*, 645–659. [[CrossRef](#)]
34. Ma, J.W.; Gao, S.; Yan, H.T.; Lv, Q.; Hu, G.Q. A new approach to time-optimal trajectory planning with torque and jerk limits for robot. *Robot. Auton. Syst.* **2021**, *140*, 103744. [[CrossRef](#)]
35. Palleschi, A.; Hamad, M.; Abdolshah, S.; Garabini, M.; Haddadin, S.; Pallottino, L. Fast and Safe Trajectory Planning: Solving the Cobot Performance/Safety Trade-Off in Human-Robot Shared Environments. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5445–5452. [[CrossRef](#)]
36. Benson, D.A.; Huntington, G.T.; Thorvaldsen, T.P.; Rao, A.V. Direct trajectory optimization and costate estimation via an orthogonal collocation method. *J. Guid. Control Dyn.* **2006**, *29*, 1435–1440. [[CrossRef](#)]
37. Trefethen, L.N. *Approximation Theory and Approximation Practice*; SIAM: Philadelphia, PA, USA, 2013; Volume 128.
38. Rojas, R.A. GSplines Source Code. 2022. Available online: <https://github.com/rafaelrojasmiliani/gsplines> (accessed on 20 June 2022).
39. Wächter, A.; Biegler, L.T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **2006**, *106*, 25–57. [[CrossRef](#)]
40. Robots, U. Universal Robots: User Manual. Available online: <https://www.universal-robots.com/download/manuals-cb-series/user/ur3/315/user-manual-ur3-cb-series-sw315-english-international-en/> (accessed on 22 May 2022).
41. Emika, F. Panda’s Instruction Handbook. Available online: https://dentec.pl/pliki/Artykul/855_dentec-franka-instrukcja-uzytownika.pdf (accessed on 1 June 2022).
42. Merkel, D. Docker: Lightweight linux containers for consistent development and deployment. *Linux J.* **2014**, *2014*, 2.
43. Rojas, R.A. Opstop Source Code. 2022. Available online: https://github.com/rafaelrojasmiliani/opstop_cpp (accessed on 20 June 2022).