1st IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0 (ARCI' 2021), 3-5 February 2021, Chamonix-Mont-Blanc, France

Oral Topic: Robotics/Industrial Robotics

Velocity Planning of a Robotic Task Enhanced by Fuzzy Logic and Dynamic Movement Primitives

B. Maggioni [1], E. Marescotti [1], A. M. Zanchettin [1], D. Piga [2], L. Roveda [2]
[1] Politecnico di Milano, Milano, Italy
[2] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), Università della Svizzera Italiana (USI) IDSIA-SUPSI, Manno, Switzerland
E-mail: beatrice.maggioni@mail.polimi.it, elia.marescotti@mail.polimi.it, loris.roveda@idsia.ch

Summary: Many industrial tasks, such as welding and sealing, require not only a precise path reference, but also an advanced velocity planning in order to achieve the target quality for the final products. In this paper, a novel approach is proposed to perform robotic trajectory planning. The developed algorithm exploits Fuzzy Logic (FL) to relate the path features (such as curves or sharp edges) to the proper execution velocity. Such a computed velocity reference is then used as an input for Dynamical Movement Primitives (DMP), providing the reference signals to the robot controller. The main improved methodology features are: path-based velocity planning, extension of DMP to variable velocity reference and smoothing of the velocity reference including robot velocity/acceleration limits. The algorithm can be implemented in a collaborative framework, defining a compliant controller embedded into the DMP for online trajectory planning.

Keywords: autonomous robotics, collaborative robotics, DMP, Fuzzy Logic, trajectory planning.

1. Introduction

Within Industry 4.0 paradigm, industrial tasks are re-designed enhancing the automatization of the production lines. In such a context, robotics playes a fundamental role, in particular considering the human-centered solutions that can be implemented [1].

To relieve the operators from tedious and hard coding of each specific application, robots must be able to learn and perform a reference task, exploiting autonomous planners for motion generation. Such topic is critical in many applications, like sealing and welding [2,3], where trajectory planning and optimization is the main objective [4,5]. The aim is, therefore, to automatically assess high-accuracy performance in trajectory tracking to achieve the target task quality.

1.1 Related Works

Trajectory planning is a hot-research topic. In [6], a widely used algorithm for welding applications is described. The planner finds the optimized motion for both the robot end-effector and joints of a welding robots, but it doesn't set the velocity along the path. In [7], a sealing task is performed using global planning interpolation and trapezoidal speed profile, but without considering any coupling between the involved degrees of freedom and without a variable velocity. In [8], Dynamical Movement Primitives (DMP) are assessed for movement sequencing trajectory planning employing velocity continuity between blocks, but there is not a punctual characterization of the velocity in the single block.

Commonly, the traditional approaches for motion planning [9] do not address the problem of the punctual characterization along the path's natural coordinate. Indeed, such approaches optimize the motion reducing the execution time, but these procedures do not take into account the execution path geometry. Those algorithms work really well in open space movements, while failing in material deposition tasks in which it is fundamental to precisely define the optimal time with a direct correlation to the optimal quality of the final result [10]. The aim of the proposed work is to reduce the total task time by automatically imposing a proper execution velocity along the path natural coordinate (*i.e.*, considering the path geometry).

1.2 Paper Contribution

Taking as a reference an automatic sealing task (within H2020 CS2 ASSASSINN project), the here presented contribution aims to design a trajectory planner able to generate the robot's reference motion to properly manage the sealant deposition.

The task execution velocity, which strongly affects the material deposition, is the main design and control parameter. The velocity reference has to be managed considering the deposition path, taking into account its geometrical features (such as sharp edges, curves, etc.) to avoid a surplus/shortage of sealing material during the deposition, while smoothing out vibrations [10]. To do that, the trajectory planning problem must consider both geometrical path features and hardware limitations (robot velocity/acceleration limits).

The here presented paper proposes an adaptive path-based task execution velocity, with a combination of Fuzzy Logic (FL) and DMP methods for the path velocity definition and for the generation of the approximating smoothed trajectory.

Fig. 1: experimental setup: Franka Emika Panda robot with a Makita caulking gun connected through a custom flange.

The FL relates the path features to the proper execution velocity, ensuring a proper sealant extrusion quality. The computed velocity is then used as input for the DMP, so that it is possible to generate the reference velocity for the robot controller [11].

While the FL methodology has been selected due to its capabilities in experimental I/O mapping [12], DMP were selected for their capabilities in trajectories representation and time/space scaling [13].

Simulations and experiments have been performed by means of a custom setup, depicted in Figure 1. Achieved results highlight the trajectory planning capabilities of the proposed framework, considering a complex reference path. A comparison with standard DMP (*i.e.*, defining an almost constant velocity reference along the path) has been performed.

2. Methodology

The algorithm structure of the proposed approach, depicted in Figure 2, is composed by trajectory pre-processing steps, where FL is exploited, and by the trajectory execution step in which the DMP are carried out using a novel modified input approach.

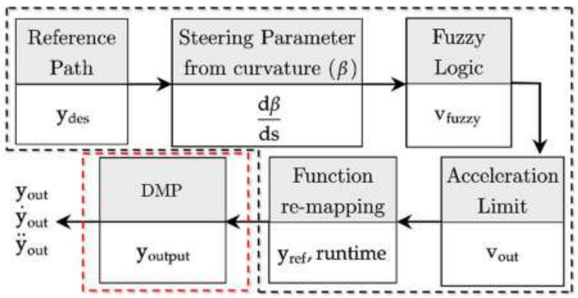

Fig. 2: trajectory planning framework. Pre-processing block is highlighted in **B**. Execution block is highlighted in **R**.

2.1 Pre-Processing Analysis

The algorithm firstly re-samples the path, $\mathbf{y}_{des}$, in order to have all the teaching points equally spaced along the path natural coordinate.

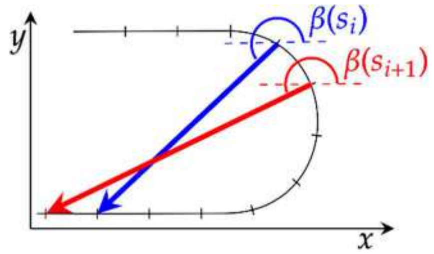

Fig. 3: example of vector definition during a path curve. In the example: n = 7.

Then, the automatic path recognition is performed, through the generation and analysis of a parameter called "*steer*", which defines the local path curvature. It is defined as follows (Figure 3):

- vector $\mathbf{v}_{i,i+n}$ is defined, connecting the considered path point with another one *n* steps forward;
- vector $\mathbf{v}_{i+1,i+1+n}$ is defined, connecting the *next* point with the *n+1* steps forward point;
- angles $\beta(s_i)$ and $\beta(s_{i+1})$ are computed (between the horizontal axis and $\mathbf{v}_{i,i+n}$ / $\mathbf{v}_{i+1,i+1+n}$, respectively);
- *steer* is defined as: $steer_i = \beta(s_{i+1}) - \beta(s_i)$.

The absolute value of the *steer* parameter is then used as an input to the FL: each value locally describes a certain path feature (*i.e.* straight lines have *steer = 0*). The FL relates, therefore, the path geometry to the reference velocity through the generation of an experimental I/O non-linear law (Figure 4). The velocity imposed by the FL is the maximum one at which a critical path's feature can be executed with proper sealant extrusion quality.

Entering the acceleration limit block (Figure 2), the planned trajectory in output from the FL is corrected accordingly to the end-effector linear acceleration/deceleration limits, by relaxing the time intervals at which each spatial point is reached. The two cases, acceleration and deceleration overshoots, are depicted in Figure 5:

- *if* $a(s_{i+1}) > a_{max}$ *then* set $a(s_{i+1}) = a_{max}$, such that $v_{out}(s_{i+1}) < v_{fuzzy}(s_{i+1})$ and the new time instant is longer;
- *if* $a(s_{i+1}) < -a_{max}$ *then* fix $v_{out,back}(s_{i+1}) = v_{fuzzy}(s_{i+1})$, and lower the previous computed velocity values, $v_{out,old}(s_i, s_{i-1}, \ldots)$, up to the convergence of related deceleration values (backward).

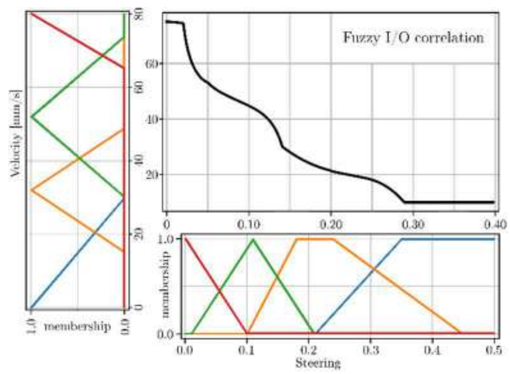

Fig. 4: FL input (*steer*) and output (*velocity*) shape functions. The Fuzzy rules are represented by the colors (*i.e.*, low *steer*, **R**, corresponds to high *velocity* and vice versa). In **B** the I/O non linear correlation is reported.

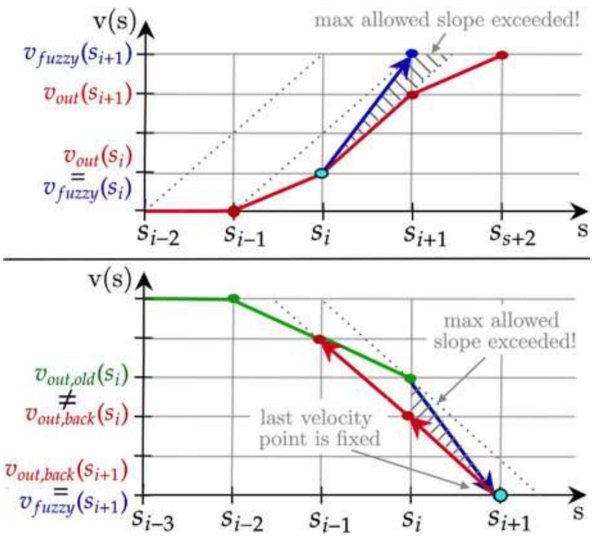

Fig. 5: time-shift for the acceleration limit: in (**B**&**G**) the computed *v(s)* that exceeds the limit, while in (**R**) the new (feasible) velocity reference.

It must be noticed that in the deceleration case, $v_{out,back}(s_{i+1})$ cannot be increased to reduce the deceleration (as done for the acceleration), so that the maximum punctual velocity does not exceed the one computed by the FL controller.

2.2 Path Execution

The computed trajectory $\mathbf{y}_{ref}$ is finally converted to be a time function, rather then a natural coordinate used in the pre-processing stages of the algorithm. Then, it is used as an input to the DMP framework, which approximates the path with smoothing features and providing continuity to the velocity.

The novel contribution proposed by this paper is related to the capabilities of the modified DMP to make use of the input path (which has to be reproduced) to achieve a final task execution with a punctual velocity characterization along the path natural coordinate (*i.e.*, adapting the execution velocity).

The provided algorithm can be executed offline, just by sending the reference positions to the robot controller, but it also allows to define an on-line human-robot collaborative framework by embedding a compliant controller to manage external interactions sending real-time signals to the robot and to perform trajectory error recovery [14]. The trajectory $\mathbf{y}_{out}$ is finally fed into the robot position controller commanding the Cartesian end-effector signals using the built-in C++ library, *lib_franka*.

3. Results

The proposed framework has been successfully tested both in numerical simulations with a Python code and experimentally on a Franka Emika Panda robot (Figure 1).

The numerical analysis focuses on the approximation of a taught path, which has been executed with both standard and modified DMP approaches (Figure 6). Considering a proper tuning of the DMP, it is possible to achieve in both cases a proper trajectory reproduction. However, considering the standard DMP, the acceleration has unwanted sharp peaks which would cause vibrations in the real experiments. Moreover, in this case the total task time must be fixed a priori, without an optimization with respect to the path length and geometry. Conversely, the novel modified DMP approach permits to control the velocity along all the path, and consequently to assign an optimized total runtime as function of the path complexity. To compare the results, the standard runtime has been set to be equal to the optimized time of the modified input approach (resulting in a different velocity profile).

In Figure 6 it is possible to see that the testing path (Figure 6a) is performed at constant velocity if the standard DMP approach is considered (Figure 6b), and with modulated velocity using the modified DMP approach (Figure 6c): the straight lines (orange) are executed at a higher velocity with respect to the small curves (light blue).

Several testing paths have been studied in order to check the consistency of the algorithm. In Figure 7 and in Figure 8 a complete generic path which has different complex geometries (small radius curves, sharp edges and a sawtooth profile) is reported. The experimental executions show the sealant extrusion comparison between standard DMP approach (Figure 7) and the modified DMP approach (Figure 8). The standard execution shows large vibrations in correspondence of the sharp directional changes (*i.e.*, in the sawtooth) and the overall extrusion is not uniform, reducing the final quality. Instead, the result of the modified DMP approach smooths out most of the vibrations, achieving a higher quality deposition along the path.

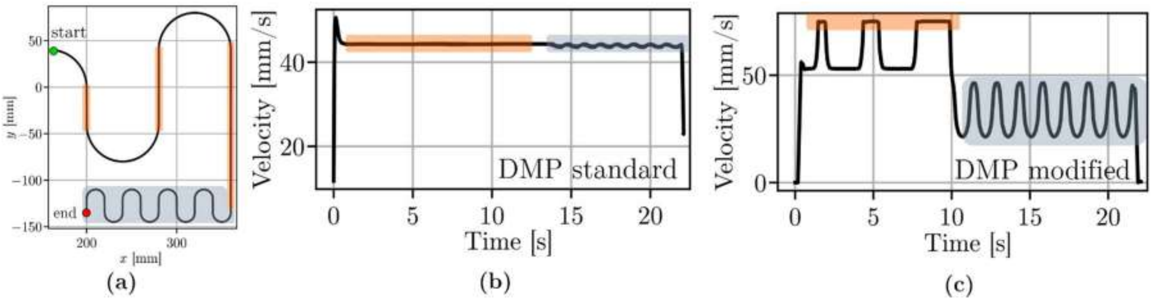

Fig. 6: two different executions of the path shown in (a): in (b) the output velocity from a classical DMP formulation is presented, while in (c) the newly proposed method permits a continuous modulation of the velocity reference.

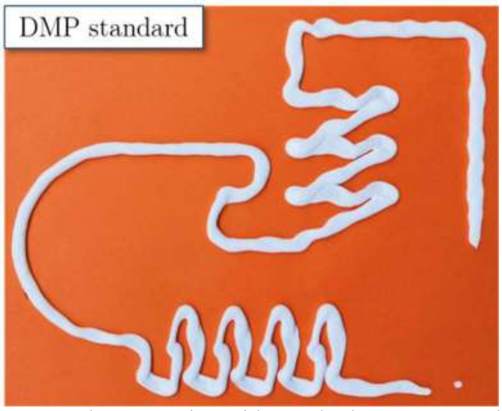

Fig. 7: sealant extrusion with standard DMP approach.

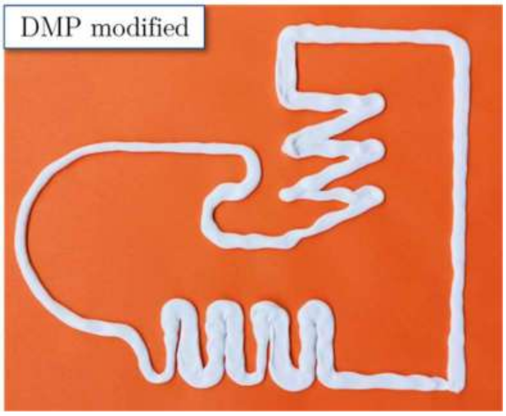

Fig. 8: sealant extrusion with modified DMP approach.

4. Conclusions

The presented paper proposes a framework for autonomous trajectory planning, being able to take into account geometrical path features for velocity definition, while also considering velocity/acceleration limits. The planned trajectory shows an execution velocity that is dependent on the geometrical path characteristics, making possible to reduce the total execution time, while obtaining the target quality for the specific deposition task.

The presented algorithm needs a manual tuning of the FL controller in order to characterize the velocity shape functions referred to some reference geometries. Current work is devoted to optimize such shape functions using a pairwise preference-based algorithm, which exploits operator's judgements [15].

While the here presented trajectory planning is performed offline, the proposed DMP framework is under implementation for real-time trajectory planning, embedding a compliant controller into the DMP framework to manage external interaction (*e.g.*, for human-robot collaboration purposes and disturbance recovery).

Acknowledgements

The work has been developed within the project ASSASSINN, funded from H2020 CleanSky 2 under grant agreement n. 886977.

References

[1] R. Galin and R. Meshcheryakov. 'Automation and robotics in the context of Industry 4.0: the shift to collaborative robots'. In: IOP Conference Series: Materials Science and Engineering. Vol. 537. 3. IOP Publishing. 2019, p. 032073.

[2] J. Ogbemhe and K. Mpofu. 'Towards achieving a fully intelligent robotic arc welding: a review'. In: Industrial Robot: An International Journal (2015).

[3] Z. Chen et al. 'An Optimized Trajectory Planning for Welding Robot'. In: Materials Science and Engineering 324.1 (2018), p. 012009.

[4] L. Biagiotti and C. Melchiorri. Trajectory planning for automatic machines and robots. Springer Science & Business Media, 2008.

[5] Chen, Yuqing, Loris Roveda, and David J. Braun. "Efficiently computable constrained optimal feedback controllers." IEEE Robotics and Automation Letters 4.1 (2018): 121-128.

[6] J. De Maeyer et al., Cartesian Path Planning for Arc Welding Robots: Evaluation of the Descartes Algorithm, in: *22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1-8.

[7] L. Anderlucci, Smooth trajectory planning for anthropomorphic industrial robots employed in continuous processes, *Politecnico di Torino*, 2019.

[8] T. Kulvicius et al., Joining movement sequences: modified dynamic movement primitives for robotics applications exemplified on handwriting, in: *IEEE Transactions on Robotics 28.1*, 2011, pp. 145-157.

[9] B. Siciliano et al. Robotics: modelling, planning and control. Springer Science & Business Media, 2010.

[10] L. Anderlucci. 'Smooth trajectory planning for anthropomorphic industrial robots employed in continuous processes'. In: Politecnico di Torino (2019).

[11] P. Pastor et al. 'Learning and generalization of motor skills by learning from demonstration'. In: 2009 IEEE International Conference on Robotics and Automation. IEEE. 2009, pp. 763–768.

[12] T. J. Ross et al., Fuzzy Logic with Engineering Applications, *John Wiley & Sons Ltd*, 2004.

[13] A. J. Ijspeert et al., DMP: Learning Attractor Models for Motor Behaviors, in: *Neural Computation 25*, 2013, pp. 328–373.

[14] M. Karlsson et al. 'Two DoFs control for trajectory tracking and perturbation recovery during execution of dynamical movement primitives'. In: IFAC-PapersOnLine 50.1 (2017), pp. 1923–1930.

[15] Bemporad, Alberto, and Dario Piga. "Active preference learning based on radial basis functions." arXiv preprint arXiv:1909.13049 (2019).