POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Parameterization of robotic welding trajectories from a demonstration

Laurea Magistrale in Automation and Control Engineering - Ingegneria dell'automazione

Author: Giovanni Fracassi

Advisor: Prof. Paolo Rocco

Co-advisors: Prof. Andrea Maria Zanchettin, Ing. Isacco Zappa

Academic year: 2021-2022

---

1. Introduction

Industry 4.0 has led to an important change of perspective in the industrial scenario. Collaborative robots (cobots) are one of the key enabling technologies of this transformation and are designed to actively cooperate with humans. In Small-Medium Enterprises (SMEs), cobot integration is still challenging. In fact, in SMEs, the growing need for customized products in low volumes, and high variability, requires fast ways to program and re-program the cobot. Furthermore, the presence of non-expert workers leads to the necessity of very intuitive cobot programming techniques.

In this sense, Programming by Demonstration (PbD) is a very popular technique to program a cobot, because is very fast and intuitive. The demonstration can be performed by kinesthetic teaching, where the operator interacts with the robot through a graphical user interface in order to save specific waypoints along the trajectory. The robot program generated from the demonstration will be used to execute the task, but strictly under the same conditions defined in the demonstration. This thesis proposes a PbD method for welding task applications, developed with an ABB GoFa CRB 15000, called GoFa SmartWeld. The method aims at parameterizing a planar welding trajectory, taught by the operator in a single kinesthetic demonstration, extracting target positions, orientations, and velocities of the end effector. Then, the execution of the trajectory through a combination of linear and circular movements, and suitable tool reorientations, is proposed. Lastly, a methodology to adapt the skill parameters for multiple workpieces is examined. The method is then validated on three workpieces of different shapes.

2. State of the art

The parameterization of welding tasks has been investigated in the literature by several authors. One effective work has been proposed by Ferraguti et al [1], which have developed MyWelder, a collaborative robotic system useful for robot-assisted welding. The system is able to identify movement primitives in order to reconstruct complex robot movements. Those primitives will be available in the GUI, and the suitable welding targets are extracted by physically moving the robot to that specific target in kinesthetic teaching. This work does not present an automatic extraction of the welding targets, and parameters like speed or tool re-orientations are not automatically retrieved. Another relevant work is proposed by Takarics et al [2], which designed a computer vision-based approach, centered on the edge detection of the workpieces. Once the edges are detected, a cubic spline will fit these points, in order to build the entire welding path and retrieve the respective position coordinates. However, the method does not manage the welding velocity, which is a fundamental parameter for the quality of the welding execution.

A different field of research investigates the method of Dynamical Movement Primitives (DMP), to model the demonstration. DMPs model the demonstrated trajectory with non-linear, mechanical, second-order systems, that converge to an attractor. In particular, in the work of Kober et al., [3], the parameters of the DMPs will drive a hybrid impedance-force controller. The method extracts the suitable parameters defining a reference frame through a scoring system but requires multiple demonstrations, thus is not suitable for our purpose.

The last work proposed is in the field of Differentiable Programming, a method useful for a given program to optimize its parameters. Alt et al.[4] define a method that executes a gradient-based neural network inversion technique to infer skill parameters directly from data. Anyway, even if this method is suitable to optimize the given parameters of the task, it requires frequent execution of the task during the optimization phase, so its implementation in the industrial scenario is limited.

3. Methodology

The method is structured in three stages:

- Training stage, responsible for the recording of the trajectory from the demonstration;
- Processing stage, in which the parameterization procedure takes place;
- Execution stage, where the parameterized trajectory is executed.

The parameterization of the trajectory takes place by combining opportunely different sets of movements. The algorithm will develop different strategies in order to segment the trajectory in linear movements, circular ones, and tool re-orientations. Furthermore, for each movement, the relative skill parameters of target positions, tool orientations, and speed will be extracted. Each movement is related to a specific command of the robot. In particular, we will have [5]:

- MoveL, where the tool center point (TCP) moves linearly to a target position, with a specific orientation and speed;
- MoveJ, useful to re-orient the TCP. The target position, orientation, and speed of the TCP are required;
- MoveC, where the TCP moves circularly along an arc to a target position. This command requires the target position, orientation, and speed of the end effector in the mid-point and end point of the arc.

3.1. MoveL and MoveJ detection

The method divides the overall trajectory into segments, extracting for each segment the parameters related to its target positions, through subsequent steps. As a first step, the algorithm will evaluate all the segments of the trajectory that are relative to a linear movement. As an example, it's possible to refer to the trajectory depicted in Figure 1 The algorithm will extract the last point of the trajectory for each linear segment, like points A, C, and E in Figure 1. The complementary portions of the trajectory will be classified as something that is not associated with a linear movement of the TCP, and their last point will be extracted as well, like points B, and D in Figure 1. The final points evaluated in this step, related to linear segments, are the target positions required for each MoveL command. After the classification of linear segments, the algorithm must be able to spot correctly the segments of the trajectory where sharp edges and tool re-orientations take place, executing them as MoveJ. In fact, during a demonstration, in correspondence with a sharp variation of the shape of the workpiece, the teacher re-orients the tool. As an example, it's possible to refer to the trajectory in Figure 2 The algorithm will extract the last point of the

Figure 1: MoveL segments are depicted in blue, and complementary segments are depicted in red

Figure 2: MoveL segments are depicted in blue, MoveJ segments are depicted in red, and MoveC segments are depicted in green

trajectory for each MoveJ segment, like points B, D, and F in Figure 2.

3.2. MoveC detection

The complementary portions of the trajectory will be classified as something that is not associated with a linear movement of the TCP, nor with its re-orientation. Thus, these segments of trajectory will be classified as circular movements of the trajectory, or MoveC. However, MoveC segments can present consecutive MoveCs with different concavity. Thus, the algorithm will analyze the change of concavity in all the MoveCs segments, extracting the parameter of the target position of the trajectory point in which this concavity change takes place. MoveC command requires an additional parameter, which is the target position of each arc midpoint. To extract this parameter, we can not rely on the trajectory of the demonstration, since it is very difficult for the operator to move the robot end effector along a circumference while keeping the right orientation. Thus, an Ordinary Least Squares (OLS) procedure is performed, fitting a circumference on each MoveC segment of the parameterized trajectory [6]. The OLS allows us to find the coordinates of the center of the fitted circumference and its radius, given the position of the points (x_i, y_i) of the trajectory of the MoveC, as reported in equation 1.

$$\begin{bmatrix} x_1 & y_1 & 1 \\ \dots & \dots & \dots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} 2x_c \\ 2y_c \\ r^2 - x_c^2 - y_c^2 \end{bmatrix} = \begin{bmatrix} x_1^2 + y_1^2 \\ \dots \\ x_n^2 + y_n^2 \end{bmatrix} \tag{1}$$

Then, the arc evaluated will be parameterized, in order to find the position of its mid-point, on the fitted circumference.

3.3. Tool orientation and welding velocity extraction

The skill parameters of the orientation of the tool in the target positions are evaluated differently in relation to the given Move command. For MoveC commands, the situation is depicted in Figure 3. In particular, the tool will be always directed to the center of the circumference, as depicted on the right of Figure 3, or rotated by 180°with respect to the z axis of the tool reference frame. The desired orientation is obtained through consequent axis angle rotations of the robot reference frame, on the left of Figure 3. The first one is a rotation around the x axis of the robot reference frame by 90°, obtaining the reference frame in the center of Figure 3. The second one is a rotation around the z axis of $(x_0 - x)$ angle, obtaining the reference frame on the right of Figure 3. The $(x_0 - x)$ angle is the angle between the x axis of the robot reference frame, called x_0, and the x axis of the tool reference frame, when directed to the center of the circumference. For MoveL and MoveJ, instead, we can refer to Figure 4

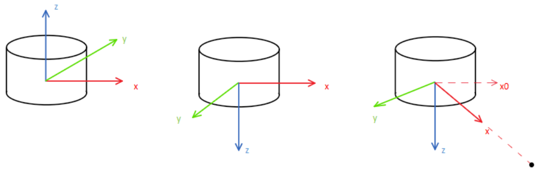

Figure 3: Orientation of the tool reference frame in MoveC

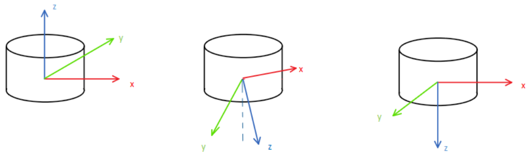

Figure 4: Orientation of the tool reference frame in MoveL and MoveJ

In particular, the z axis of the reference frame in the center of Figure 4 will be aligned with the negative direction of the z axis of the robot reference frame, depicted on the left in Figure 4. The final tool orientation is depicted on the right of Figure 4

The velocity instead will be evaluated through a mean calculation on the overall array that contains the end effector velocities, for all the points of the demonstration, evaluated with differential kinematics. Velocity is a fundamental parameter for the welding task, because of its impact on the overall quality of execution. Hence, our method, in the execution stage, will evaluate the mean value of the velocity taught by the operator in the demonstration.

4. Adaptation of skill parameters to multiple workpieces

In PbD, once the demonstration via waypoints has been performed, the robot program generated will be valid only on the same scene of the demonstration, with the same objects in the same positions and orientations. Thus, a possible extension of the method of the parameterization of the trajectory to a scene where are present multiple workpieces, of the same shape as the demonstrated one, but in different orientations and positions, are evaluated. These orientations and positions must be specified in input to the method. The skill parameters related to the target positions and orientations of each segment of the trajectory can be roto-translated in each object reference frame on the scene. For example, we can refer to Figure 5 and 6.

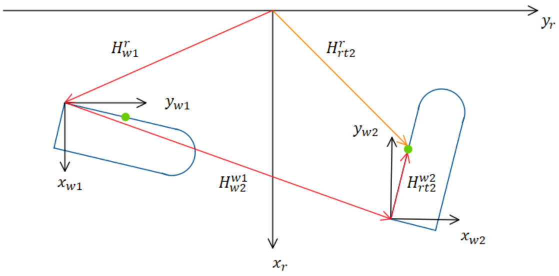

Figure 5: Multiple workpieces on the scene

The workpiece on which the demonstration has been performed is the one on the left of Figure 5, and its object reference frame is (x_{w1}, y_{w1}, z_{w1}). The set of equations that govern the roto-translation procedure for the robtarget depicted in green in Figure 5, is the following:

$$\begin{cases} H^{W2}_{RT2} = H^{W1}_{RT1}, & \text{(2a)} \\ H^{R}_{RT1} = H^{R}_{W1} \times H^{W1}_{RT1}, & \text{(2b)} \\ H^{R}_{RT2} = H^{R}_{W1} \times H^{W1}_{W2} \times H^{W2}_{RT2}, & \text{(2c)} \end{cases}$$

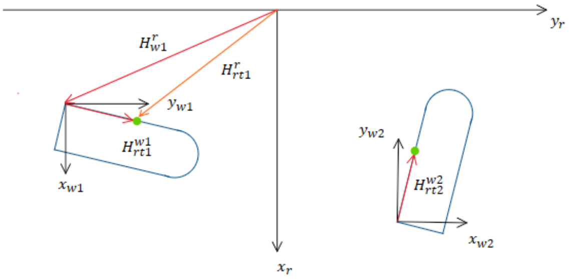

Figure 6: Evaluation of H^{R}_{RT1} matrix

Since the workpieces on the scene have the same shape, equation 2a holds. In fact, in each object reference frame, the position and orientation of the considered robtarget are the same for each workpiece. Thus it is enough to evaluate H^{W1}_{RT1}, inverting the equation 2b. A brief description of the relationships presented in equation 2b is depicted in Figure 6. The value of the robtarget of the workpiece on the right of Figure 5, evaluated in the robot reference frame, can be calculated in equation 2c. The method requires in input the pattern of the workpieces on the scene, represented by the matrix H^{W1}_{W2}. In particular, the parameters required are the orientation and the position of each object reference frame, with respect to the one on which the demonstration has been performed. In the example proposed, the aforementioned reference frame is (x_{w1}, y_{w1}, z_{w1}).

The velocity, instead, will not be modified, remaining the same for each trajectory on the scene, since it's a very important parameter in the welding task. Thus, all trajectories will be executed with the mean value of the velocity taught by the operator during the demonstration.

5. Experimental validation

The experimental validation is structured in three main steps, and each one is executed on a different workpiece. It's important to point out that the parameterization depends on the correctness of the demonstration performed by the operator. In fact, for example, if the operator moves the end effector linearly in a circular portion of the shape of the workpiece, the algorithm will parameterize this part of the trajectory as a linear segment. So the performances of the algorithm will be conditioned on the "quality" of the demonstration. The goal of the validation is to prove the capability of the method summarized in the following claims:

- Claim one: correctness in the trajectory parameterization
- Claim two: correctness in the execution of the welding, also for a linear pattern of workpieces
- Claim three: correctness in the execution of the welding of multiple workpieces with different orientation

Three different experiments, each one related to one respective claim, can be defined. Experiment 1, which aims to validate claim one, is depicted in Figure 7.

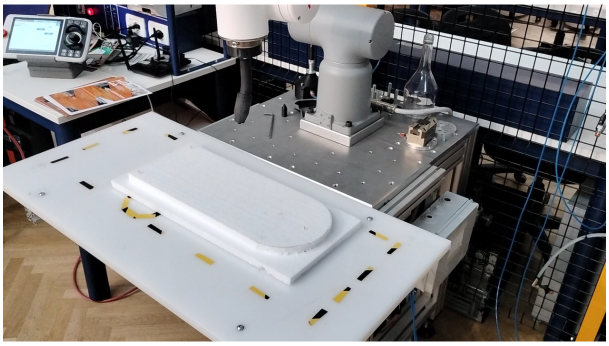

Figure 7: workpiece number 1

The goal of this validation step is to evaluate the magnitude of possible parameterization errors like a wrong position identification of the arc mid-point of a specific MoveC, or a mismatch in the evaluation of the parameterized segment like a move command entirely missing. Experiment 2, which aims to validate claim two, is depicted in Figure 8.

Experiment 3, which aims to validate claim three, is depicted in Figure 9.

For Stage One, the algorithm parameterizes correctly the trajectory in 19/50 demonstrations. Anyway, as mentioned above, these performances are dependent on the correctness of the demonstration performed by the operator.

For Stage Two, the results are shown in Figure 10.

In particular, 76% of the executions are correct, instead, 24% present multiple errors. Thus, 25% of the wrong executions are caused by the sixth joint going out-of-range, 58.3% is caused by a wrong orientation of the tool along the geometry of the workpiece, and 16.7% is caused by a wrong parameterization of the trajectory.

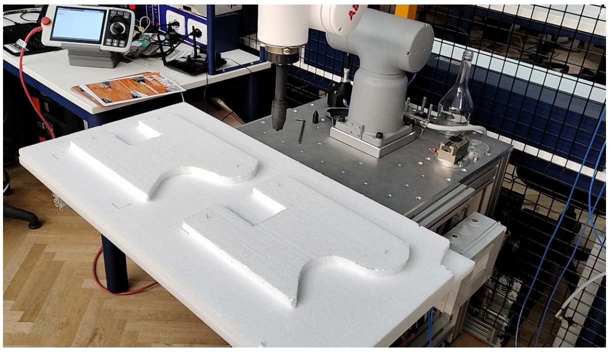

Figure 8: workpiece number 2

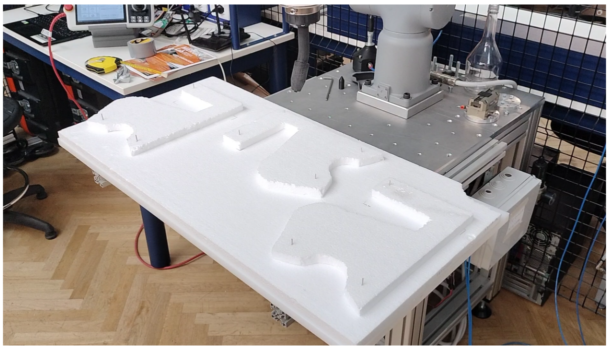

Figure 9: workpiece number 3

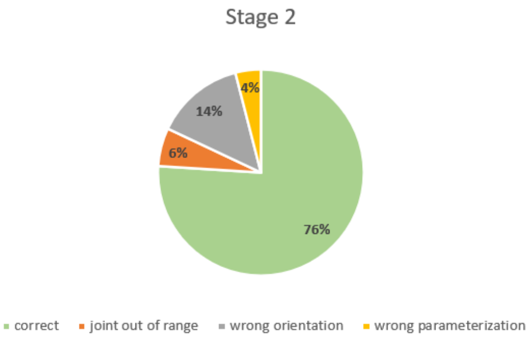

Figure 10: Validation Stage Two

For Stage Three of the validation, the results are shown in Figure 11.

In particular, 80% of the executions are correct, and 20% present errors. In fact, 40% of the wrong executions are caused by the sixth joint going out-of-range, 50% is caused by a wrong orientation of the tool along the geometry of the workpiece, and 10% is caused by a wrong parameterization of the trajectory.

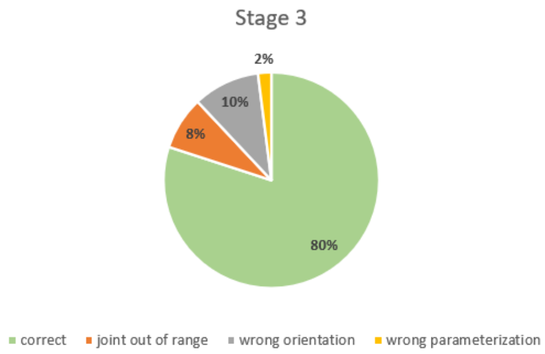

Figure 11: Validation Stage Three

6. Conclusions

The proposed methodology aims to realize a parameterization of a welding trajectory, recorded from a one-shot demonstration of a welding task. A method to extract skill parameters of target positions, orientation, and velocity has been formalized, and the execution of the task through suitable robot movements has been developed. Furthermore, the adaptation of the extracted skill parameters on a scene that presents multiple workpieces of the same shape, but with different positions and orientations, has been investigated .

Future studies may focus on the integration of a suitable vision system on the robot, in order to automatically evaluate the number of workpieces present on the scene, and their position and orientation.

References

[1] Federica Ferraguti, Valeria Villani, and Chiara Storchi. Mywelder: A collaborative system for intuitive robot-assisted welding. *Mechatronics*, 89:102920, 2023.

[2] Bela Takarics, Peter T Szemes, Gyula Németh, and Peter Korondi. Welding trajectory reconstruction based on the intelligent space concept. In *2008 Conference on Human System Interactions*, pages 791–796. IEEE, 2008.

[3] Jens Kober, Michael Gienger, and Jochen J Steil. Learning movement primitives for force interaction tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3192–3199. IEEE, 2015.

[4] Benjamin Alt, Darko Katic, Rainer Jäkel, Asil Kaan Bozcuoglu, and Michael Beetz. Robot program parameter inference via differentiable shadow program inversion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4672–4678. IEEE, 2021.

[5] ABB Robotics. Technical reference manual: Rapid instructions, functions and data types. *ABB Robotics*, 2014.

[6] Ian D Coope. Circle fitting by linear and nonlinear least squares. *Journal of Optimization theory and applications*, 76:381–388, 1993.