A Probabilistic Approach to Reconfigurable Interactive Manufacturing and Coil Winding for Industry 4.0

Stefano Michieletto, Francesca Stival, Enrico Pagello

Intelligent Autonomous Systems Lab, Department of Information Engineering, University of Padova

---

Abstract

Robotic industry needs new, innovative, ideas to be globally competitive. Conventional industrial robots are not able to adapt to changes in the assembly processes. Flexible assembly applications are actually uncommon and only isolated attempts exploit industrial robots to perform tasks with variability in the parts. Variability aspects are emphasized when developing novel manufacturing applications involving human robot collaboration which are the foundation of Industry 4.0 systems. In this chapter, we will describe how variability can be considered and mathematically described as part of the problem to obtain a flexible robotic solution. The selected approach is based on a probabilistic representation of the task obtained starting from a set of demonstrations collected from humans. The chapter will illustrate the different steps leading to the complete learning framework. We will start by describing the strategies adopted during the data collection phase. From the raw data, the design of feature extraction procedures will be provided alongside with a set of preprocessing techniques used to remove noisy and incoherent information. The resulting dataset will be used to train a model of task by following a probabilistic approach. The output of the model will be exploited to actuate an industrial manipulator in the context of significant production scenarios. The robot motion strategies will be also analyzed depending on the level of flexibility requested from the specific use case. Two main use cases will be introduced: (i) the automatic assembly of a car door with its module, and (ii) robotize the manufacturing process of elec-

Email addresses: michieletto@dei.unipd.it (Stefano Michieletto), stivalfr@dei.unipd.it (Francesca Stival), enrico.pagello@unipd.it (Enrico Pagello)

tric machines, in particular winding of coils on stator or rotor cores. Each problem will be mathematically formulated by modeling both the robotic platform and the target to achieve. The influence of the scenario variability with respect to the computed robotic motion will be considered. The system flexibility will be evaluated by means of an extensive set of benchmarking tests by recording data and actuating robots in both simulated and real environments. Achievements will be compared with respect to state of the art solutions by defining a set of objectives and metrics. The goal is to measure the performance of the system, for example in minimizing time and energy needed to move the robot in the working space, in generating an effective human-robot interaction with low reaction time and high accuracy, and in providing an intuitive robot learning technique to easily let the human teach the robot new tasks. Dynamic online reconfigurability of the framework will be considered by testing its capability to deal with novel situations and new products. The integration of the proposed technologies with current robotic systems will be discussed and a solution based on the Robot Operating System (ROS) will be proposed to provide a good infrastructure for network communication as well as all the tools necessary to a modern distributed and heterogeneous system. The feasibility and cost effectiveness of the developed solutions will be taken into account in order to demonstrate the applicability of the proposed approach in actual industrial settings.

Keywords: Robotics, Robot Programming by Demonstration, Kinesthetic Teaching, Robot Learning, Industry 4.0, Automatic Assembly, Electric Motors

---

1. Introduction

During the last years, the dissemination of robots has exploded in many aspects of everyone lives. Up to now, we can meet robotic devices not only in the most advanced factories, but also in our houses. Nowadays, it is common to find a robot autonomously cleaning up a house, or assisting a surgeon during a medical operation. A key factor for the Industry 4.0 upgrade is the use of robots [1]. Nowadays, manipulators are employed for supply chains in which the same task should be accomplished several times in a repetitive manner. The main reasons are the price decrease and the boosted investments to develop technology in the field of robotics. All these new technologies have the goal of improving humans' quality of life, for example by reducing their workload, or by substituting the operator in dangerous and strenuous tasks. In the majority of the cases, human operators can understand easily how to perform the task even in complex situations, but they have not the expertise to program the robot. A useful solution could be obtained if the operator would be able to teach the robot how to perform a certain task, guiding the robot or showing himself what to do by using a Robot Learning by Demonstration paradigm [2]. Also known as Robot Programming by Demonstration, this paradigm aims to train robotic devices through human demonstrations in order to teach them how to perform a task [3] and many examples in the literature show the useful aspects of applying robot learning by demonstration techniques to provide an easy way to program robots [4].

Up to now, several research groups have developed different paradigms and techniques, but only a limited number of attempts have been exploited in real industrial environments. Myers et al. [5] wanted to automatically insert a PC card into a backplane slot on the motherboard treating forces/moments as the sensed inputs and robot velocities as the control outputs. Baroglio et al. [6] believe that the robots ability to gain profit from its experiences is crucial for fully exploiting its potential. They analyzed several approaches and tested them in a classical industry-like problem: insert a peg into a hole. The task was performed while recovering from error situations, in which, for instance, the peg is stuck midway because of a wrong inclination. Neto et al.[7] presented a way to program a robot showing it what to do by using gestures and speech. The gestures are extracted from a motion sensor, namely a Wii remote controller. The Japanese company Fanuc is developing robots that use reinforcement learning to train themselves [8]. Fanucs robot learns how to pick up objects while capturing video footage of the process. The new knowledge is used to refine a deep learning model that controls robot actions. It has been proved that after about eight hours the robot reaches up to 90 percent accuracy or above, almost the same as if it was programmed by an expert.

Usually, robots need a large number of demonstrations to learn how to perform a simple task. Selecting a specific Machine Learning algorithm could help to reduce the number of demonstrations in certain contexts, but otherwise improvements are limited. On the other hand, overcomplicated Machine Learning algorithms could end in overfitting. Consequently, the model could fail in predicting reliably future observations. The same problem may occur when data are limited and the model focuses on specific situations without the possibility to fit additional data. With overfitting, the framework lacks of abstraction and generalization capability and it will not be able to face even limited variations. Generalization is a key concept if the goal is to obtain a relation between robot movements and the objects to be manipulated during the task. Nevertheless, very simple algorithms coupled with an excessively wide dataset could lead to underfitting. The resulting model will not capture the common characteristics among the data providing poor predictive performance. A good way to incorporate variability is to consider actions performed by many different subjects [9].The same gesture can slightly change depending on who is doing the movement. The gesture remains correct, but the ways to perform it are almost infinite. However, the risk of underfitting is forestalled since there are common characteristics among the numerous ways different subjects perform the same movement.

In kinesthetic demonstration [10], the robot is physically guided through the task by the humans. An alternative approach could be learning from visual information [11]. In fact, the constraints characterizing the movement should be extracted from a sequence of images. Furthermore, industrial applications requires particular conditions on safety and efficiency. These aspects have to be taken into account when building the model to control a robotic device. The advent of Industry 4.0 brought new and innovative challenges [12] for robotics. The new concept of industry aims at reducing the waste, while maximizing the customization of the product, therefore a flexible and dynamic production line is essential. An efficient way to produce is necessary in modern factories, and the manufacturing system should be able to switch production in a very short time. The presence of intelligent and collaborative robots is a key factor for the fulfillment of these targets. Old-fashion robots are expensive devices, closed in a cage, repeating continuously the same task. Reprogram one of these robots takes time, money and requires the intervention of specialized programmers. New robots are lightweight and no longer closed in cages, since they are equipped with force sensors, aware of possible contacts with the surrounding world. The characteristics of collaborative robots (Cobots) make them ideal to be Programmed by Demonstration. This programming paradigm reduces the time needed to program the robot, since there is no requirement of specialized personnel. On the contrary, the machine will learn the task by observing the demonstrations performed by skilled workers which know well the tasks the robot should do. Furthermore, Cobots offer the possibility of having humans and machines working on the same workplace, an also to operate together to fulfill the same task. The closeness of the machine to humans arises several safety problems. The main issue is to avoid accidental contacts among humans and robots. Giving robot the capabilities perceive, understand and react to what happens in the environment will become essential in the factories of the future. In other words, the robot should be intelligent, capable of interpreting feedback from outside, and it will need the ability to understand and predict human movements.

In this chapter, we will describe how variability can be considered and mathematically described as part of the problem to obtain a flexible robotic solution by applying Robot Programming by Demonstration to a real industrial case. In particular, two main use cases will be introduced: (i) the automatic assembly of a car door with its module, and (ii) boost the production of electric motor coils, by automatizing the copper winding procedure. These cases were part of a European challenge aiming to encourage collaboration between academic and industrial counterparts [13]. This challenge aims to boost the collaboration between research and industrial partners, in order to achieve innovative results. The remaining of the chapter is structured as follow. Sec. 2 presents the general probabilistic approach and the mathematical formulation of the learning framework. The first case study regarding the automatic assembly of a car door with its module is described in Sec. 3. While, Sec. 4 enters into details of the second case study in which we robotized the manufacturing process for winding coils for stator or rotor cores in electric motors. Finally, in Sec. 5, we summarize the work and the achieved results.

2. Probabilistic framework

The learning framework proposed in this work aims at estimating the control model of a robotic device (output data) starting from human information (input data) to create flexible probabilistic approach for reconfigurable interactive manufacturing. Data undergo into a preprocessing phase in order to remove artifacts and noise from the signals. After preprocessing two main phases can be recognized, i.e. an offline and an online elaboration. During the offline phase, data are collected from many subjects while performing a certain task. In this phase, the information available for each trial should contain both input and output data. A probabilistic model is trained in order to represent the processed information with a limited set of parameters. The online phase considers data directly acquired from the environment, exploiting the model previously computed to estimate the corresponding robot motion. In the following, all the concepts introduced so far will be accurately described.

2.1. Signal processing

Selecting an extremely sophisticated machine learning technique does not guarantee high accuracy in data estimation. The aim of the preprocessing procedure is to obtain a set of significant features to estimate or in some cases predict robustly and online the movement performed by a subject. In fact, the process of filtering the significant information from input signals can affect the actual success of the entire framework. The preprocessing phase is essential to obtain a well-balanced combination of similarity and variability within the signal. On one hand, if the considered signals have nothing in common, the final model would not work properly. On the other hand, a certain amount of variability should be integrated in the system, in order to build a general model which can work with new, unseen data.

2.1.1. Smoothing and Normalization

Depending on the specific case study, the information collected can undergo to different kind of preprocessing. Although, it is very likely that data in input are very jagged and not good enough to be used to build good probabilistic models, since the great variability of the signal results in poor model performances. Therefore, a simple procedure to be applied to signals with the aim of obtaining better and more robust models consists of a combination of smoothing and normalization.

The smoothing function is based on a moving average filter. At the instant t, the average of S data points available within the windows is computed in order to smooth the data. This process is equivalent to lowpass filtering, with the response of the smoothing given by Eq. 1

$$\gamma_S(t) = \frac{1}{S+1} \sum_{s=1}^{S} \gamma(t-s) \tag{1}$$

The smoothing function performs a local regression using weighted linear least squares and a second degree polynomial model.

Differences in the amplitude and in the mean of the signal requires the application of a normalization technique. The normalization process ensures the regularization of the signal in order to obtain a more robust model. The normalization has been implemented in two different manners for training and testing phases. Since the training phase is executed offline, the normalization has been accomplished by using the relative maximum within the specific trial involved in the process. Instead, during the online testing procedure the information about the relative maximum is not available.,We needed to use a different method to be able to compute the normalization online. For obtaining this result, the mean of the relative maximums collected during the entire training set has been used as normalization factor.

2.2. Gaussian Mixture Model

Gaussian Mixture Model is a parametric probabilistic model that assumes all data points are generated from a mixture of a finite number of Gaussian distributions. These distributions completely characterize the model, therefore it is composed by a weighted sum of Gaussian components. In particular, three parameters for each Gaussian component are sufficient to represent the whole information: *mean*, *covariance* and *weight*. These parameters are estimated from training data using the iterative Expectation Maximization (EM) [14] algorithm. EM is a statistical algorithm that iteratively finds locally maximum likelihood parameters of a probabilistic model when equations can not be solved directly. The locally maximum likelihood is obtained repeating cyclically two phases.
Expectation (E) step creates a function for the expectation of the log-likelihood evaluated using the following estimate of the components parameters:

$$p_{k,j}(t+1) = \frac{\pi_k(t)\mathcal{N}(\zeta_j; \mu_k(t), \Sigma_k(t))}{\sum_{i=1}^{K} \pi_i(t)\mathcal{N}(\zeta_j; \mu_i(t), \Sigma_i(t))} \tag{2}$$

Maximization (M) computes parameters maximizing the expected log-likelihood found during last E step:

$$\pi_k(t+1) = \frac{1}{N}\sum_{i=1}^{N} p_{k,j}(t+1) \tag{3}$$

$$\mu_k(t+1) = \frac{\sum_{i=1}^{N} p_{k,j}(t+1)\xi_j}{\sum_{i=1}^{N} p_{k,j}(t+1)}$$

$$\Sigma_k(t+1) = \frac{\sum_{i=1}^{N} p_{k,j}(t+1)(\zeta_j - \mu_k(t+1))(\zeta_j - \mu_k(t+1))^\top}{\sum_{i=1}^{N} p_{k,j}(t+1)}$$

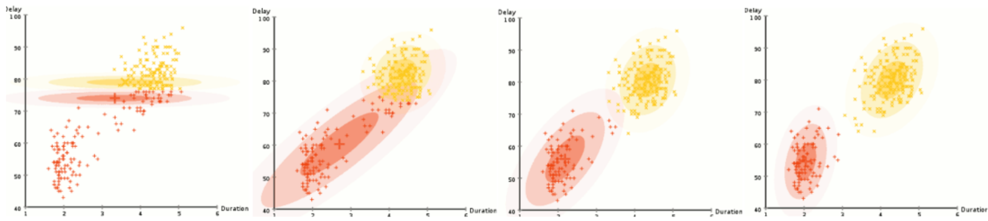

Figure 1: Example of the EM algorithm. The red and yellow ovals show how the algorithm adapt the parameters to fit the data (the red and yellow crosses)

The result is a continuously improving adaptation to the best representation of the input data as it is shown in Fig. 1. The EM loop stops when the increment of the log-likelihood $\mathcal{L} = \sum_{j=1}^{N} \log\left(p\left(\zeta_j|\theta\right)\right)$ at each iteration becomes smaller than a defined threshold ϵ, i.e. $\frac{\mathcal{L}(t+1)}{\mathcal{L}(t)} < \epsilon$.

A possible limitation in the learning process is the fact that EM requires a priori specification of the number of Gaussian components K. Selecting the correct K is a crucial task. On one hand, an overestimation of this parameter might lead to over-fitting and, consequently, to a poor generalization. On the other hand, an underestimation will result to poor predicting performances. Several entropy based model selection techniques has been proposed in literature to estimate this parameter (e.g. Bayesian Information Criterion (BIC) [15], Akaike Information Criterion (AIC) [16], Minimum Description Length (MDL) [17], and Minimum Message Length (MML) [18]). In our work, we choose a standard approach based on BIC (Eq. 4).

$$S_{BIC} = -2\mathcal{L} + n_p \log N \tag{4}$$

with $\mathcal{L} = \sum_{j=1}^{N} \log\left(p\left(\zeta_j|\theta\right)\right)$, the log-likelihood for the considered model θ; $n_p = (K-1) + K(D + \frac{1}{2}D(D+1))$, the number of free parameters required for a mixture of K components with full covariance matrix. Considering (a) H, number of subjects involved in the study; (b) n, number of trials per subject used to train the system, (c) T, number of repetitions of each trial, (d) $N = nT(H-1)$, total number of data samples. The number of subjects is decreased by one, since the model is trained on $H-1$ subjects and then tested on the excluded subject h. Then a single data $\zeta_j, 1 \leq j \leq N$ in input at the framework can be written like in Eq. 5.

$$\begin{aligned} \zeta_j &= \{\xi(t), \alpha(t)\} \in \mathbb{R}^D \\ \xi(t) &= \{\xi_c(t)\}_{c=1}^{C}, \\ \alpha(t) &= \{\alpha_g(t)\}_{g=1}^{G}. \end{aligned} \tag{5}$$

where $C = |\xi|$, number of channels considered; $\xi(t) \in \mathbb{R}^C$, the set of values assumed from all the considered channels at the time instant t, with $\xi_c(t) \in \mathbb{R}$, the value assumed from the c^{th} channel at the time instant t; $G = |\alpha|$, number of joint bending angles; $\alpha(t) \in \mathbb{R}^G$, the set of values assumed from the considered joint bending angles at the time instant t, with $\alpha_g(t) \in \mathbb{R}$, the value assumed from g^{th} joint bending angle at the time instant t; and $D = C+G$, the dimensionality of the problem. The final resulting probability density function is computed as:

$$p\left(\zeta_j\right) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\zeta_j; \mu_k, \Sigma_k\right) \tag{6}$$

with π_k priors probabilities; $\mathcal{N}\left(\zeta_j; \mu_k, \Sigma_k\right)$ Gaussian distribution; μ_k mean vector of the k-th distribution; Σ_k covariance matrix of the k-th distribution, K number of Gaussian components.

2.3. Incremental Gaussian Mixture Model

The construction of the probabilistic model is a time consuming task. Furthermore, often we are not interested in the building of a new model from scratch. On the contrary, in some cases we want to update the model, adding the information from new demonstrations, in order to make the model fit better on a specific task, without losing the knowledge, the robustness and the generality acquired from previous examples. For these reasons, we implemented an incremental version of Gaussian Mixture Model (GMM), namely Incremental Gaussian Mixture Model (IGMM), able to update the model as new demonstrations are received from operators. We implemented and tested the *Generative method* described in [19].

The first step consists of building a GMM with the classic EM algorithm as described in Sec. 2.2. When new data are available ξ_i, they undergo the following passages:

1. Synthetic data are stochastically generated with by performing a regression on the current GMM. The generated data are a compact representation of the previous data distribution.

2. A new GMM is computed on the whole set composed by new data ξ_i and the stochastically generated ones.
3. A learning rate $\alpha \in [0,1]$ is introduced to modulate the contribution from the new data and the stochastically generated ones. $\alpha = \frac{\tilde{N}}{\tilde{N}+N}$, with $\tilde{N}$ number of new datapoints available, and N number of datapoints from previous demonstrations.
4. Given $n = n_1 + n_2$ number of samples for the iterative learning procedure, with $n_1 \in \mathbb{N}$ number of trials from the new observations, and $n_2 \in \mathbb{N}$ number of trials generated from the previous model. The new training set is then defined by:
   $\xi_{i,j} = \tilde{\xi}_j$, if $1 < i \leqslant n_i$
   $\xi_{i,j} = N(\hat{\mu}_j, \hat{\Sigma}_j)$, if $n_i < i \leqslant n$
   $\forall j \in \{1, ..., T\}$, with T number of timestamps, with $n_1 = [n\alpha]$ ([.] nearest integer function).
5. The training set of n trials is used to refine the model by updating the current set of parameters (π_k, μ_k, Σ_k) by using the EM algorithm.

2.4. Gaussian Mixture Regression

The regression rpocess has the goal of continuously estimating the robot joints bending angles. The Gaussian Mixture Regression (GMR) provides a smooth generalized version of the signal starting from the GMM. GMR estimates the joints angles $\hat{\alpha}$ and their covariance from the Electromyography (EMG) ξ (and eventually accelerometers φ) signals known a priori, respectively using Eq. 7 and Eq. 8.

$$\hat{\alpha} = E\left[\alpha\left|\xi, \varphi\right.\right] = \sum_{k=1}^{K} \beta_k \hat{\alpha}_k \tag{7}$$

$$\hat{\Sigma}_s = Cov\left[\alpha\left|\xi, \varphi\right.\right] = \sum_{k=1}^{K} {\beta_k}^2 \hat{\Sigma}_{\alpha,k} \tag{8}$$

with $\beta_k = \frac{\pi_k \mathcal{N}\left(\xi,\varphi\left|\mu_{p,k},\Sigma_{p,k}\right.\right)}{\sum_{j=1}^{K}\mathcal{N}\left(\xi,\varphi\left|\mu_{p,j},\Sigma_{p,j}\right.\right)}$, the weight of the k^{th} Gaussian component through the mixture; $\hat{\alpha}_k = E\left[\alpha_k\left|\xi, \varphi\right.\right] = \mu_{\alpha,k} + \Sigma_{\alpha p,k}\Sigma_{p,k}^{-1}\{\{\xi,\varphi\} - \mu_{p,k}\}$, the conditional expectation of α_k given $\{\xi,\varphi\}$; $\hat{\Sigma}_{\alpha,k} = Cov\left[\alpha_k\left|\xi,\varphi\right.\right] = \Sigma_{\alpha,k} + \Sigma_{\alpha p,k}\left(\Sigma_{p,k}\right)^{-1}\Sigma_{p\alpha,k}$, the conditional covariance of α_k given $\{\xi,\varphi\}$.

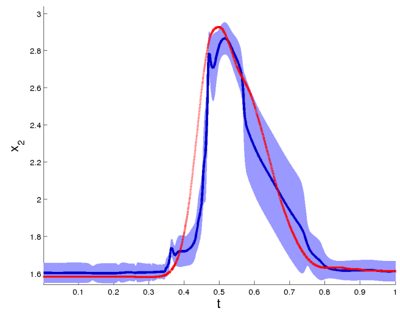

Figure 2: GMR estimation and variance (blue) VS actual motion.

Assuming that the parameters (π_k, μ_k, Σ_k) defining the k^{th} Gaussian component are decomposed as follows:

$$\mu_k = \{\mu_{p,k}\, \mu_{\alpha,k}\} \qquad \Sigma_k = \begin{bmatrix} \Sigma_{p,k} & \Sigma_{p\alpha,k} \\ \Sigma_{\alpha p,k} & \Sigma_{\alpha,k} \end{bmatrix} \tag{9}$$

in which the mean and the covariance of the known a priori information $p = \{\xi, \varphi\}$ have been represented respectively with μ_p and Σ_p. Thus, the model is completely defined by the Gaussian components composed solely by weights, means and covariances obtained by means of the EM algorithm. Subsequently, the information composing the model allows us to calculate a generalized motion $\hat{\zeta} = \{\xi, \varphi, \hat{\alpha}\}$ (Fig. 2).

2.5. System effectiveness

Besides the specific metrics established for each case study, a measure widely used for evaluating the goodness of the predicted measure [20] [21] is the correlation coefficient $\rho_{\alpha,\hat{\alpha}}$. This value is calculated between the predicted output $\hat{\alpha}$ and the real one α (Eq. 10), and it gives a measure of the model performances by means of the statistical relationships between different signals and different subjects. In particular, the correlation coefficient is a measure of the degree of linear dependence between two variables, and it is based on the covariance ($Cov(\alpha, \hat{\alpha})$) and the standard deviations (σ_α and $\sigma_{\hat{\alpha}}$) of the considered variables. The resulting formula is reported in Eq. 10.

$$\rho_{\alpha,\hat{\alpha}} = \frac{Cov(\alpha, \hat{\alpha})}{\sigma_\alpha \sigma_{\hat{\alpha}}} \tag{10}$$

The correlation coefficient can assume all the values between 1 and -1, where 1 is total positive correlation and indicates a perfect direct linear relationship (correlation), 0 is no correlation, and -1 is total negative correlation, or a perfect decreasing linear relationship (anticorrelation). The closer the coefficient is to either -1 or 1, the stronger is the correlation between the variables, while the closer it is to zero, the weaker is the correlation. When the correlation reaches zero the variables are independent.

Another common measure of the effectiveness of GMM-based systems is Normalized Mean Square Error (NMSE). This function measures the goodness of fit between test and reference data. NMSE (Eq. 11) costs vary between -∞ (bad fit) to 1 (perfect fit).

$$\text{NMSE}(t) = 1 - \left\| \frac{\hat{\alpha}(t) - \alpha(t)}{\hat{\alpha}(t) - \mu_t(\alpha)} \right\|^2 \tag{11}$$

where t is the temporal instant from the beginning of the trial; $\hat{\alpha}(t)$ is the estimated output at the instant t; $\alpha(t)$ is the groundtruth at instant t; $\mu_t(\alpha)$ is the mean along the time of considered quantity. In this work, we used both methods as an initial feedback for understanding if a model could work or not in an actual test with the robot.

3. Automatic assembly

The main objective in this industrial case is to perform the automatically assembly of a car door with its module. The evaluation of the developed techniques has been carried out in a dedicated facility for Industry 4.0 in Stuttgart with the constraints imposed by real industrial cases to guarantee success and repeatability of the tasks to achieve in a quantifiable way. The set of sensors available in the facility was predetermined without the possibility of adding new sensors or modifying the already existent ones. We had to to work with a previously installed system as a part of a factory supply chain partially obsolete and outdated. Industrial settings were composed of complex and challenging workcell layouts with changing illumination and tight workplaces. Moreover, it was a dynamic environment with people working alongside robots in a collaborative manner and uncertain position of parts to be assembled. Starting from the Robot Programming by Demonstration paradigm, we designed a framework to learn novel robot trajectories and configurations on the fly from human demonstrations.

Such approach is quite novel in research, and still few attempts are available in the literature. As suggested by Chen *et al* [22], flexible assembly applications are actually uncommon and only a small portion of industrial robot are used to perform tasks with variability in the parts. In fact, conventional industrial robots are not able to adapt to changes in the assembly processes. On the other hand, Goya *et al* [23] indicated flexibility in the automotive manufacturing as one of the more competitive weapons in the economical analysis of North American automotive industry. They proposed the possibility of switch easily and with a lower risk from a production line to an other as main advantage in future achievements with respect to foreign competitors. The reduced risk should permit industries to invest in low volume-high risk products, since the money and time loss would be minor and the production line would remain the same.

While hardware and firmware composing the robotic system were fixed, we had the possibility to use Robot Operating System (ROS) as programming tool. ROS is now the standard de facto in research robotic frameworks, but it has been only recently accepted as a tool for industry. The importance of ROS has been expressed in [24] by Tavares *et al.* They analyzed a pick and place task by combining several layers of control. Using ROS in developing industrial applications gives the possibility of efficiently divide layers in standardized and compact blocks able to interact one with each other to autonomously correct errors during the accomplishment of the task. In our solution, we took advantage of the modularity and standardization ROS characteristics to fuse together visual and robot learning techniques in order to face the variability in the system configuration. Stability and reliability of each method have to be enhanced to meet the requirements in terms of elapsed time and hardware compatibility.

The human-robot interface has been designed to easily teach the system with novel door assembly combinations by starting from previous works. Vakanski *et al* [25] suggested to take advantage of robots ability of learning from what surrounds them, transferring skills to a robot thanks to multiple demonstration of the same skill under similar conditions. Vakanski's idea was to learn the robot trajectory by observing a subject moving a tool. Instead, in our solution the person is actually moving the robot to acquire information about the desired motion. The robot has to infer a generalized trajectory obtained extracting relevant features from the demonstrations.

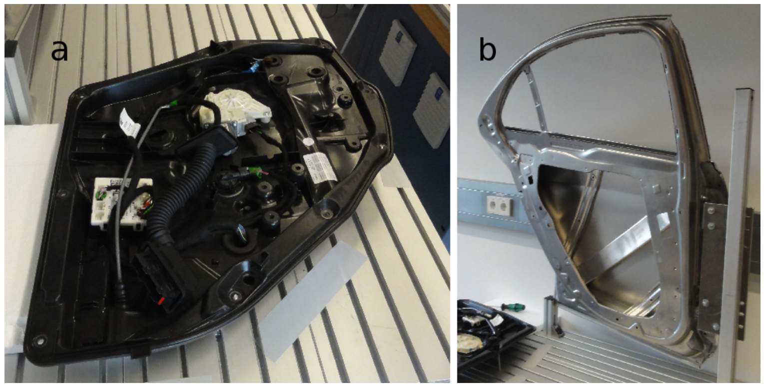

Figure 3: a) Plastic module, b) Car door where the module has to be inserted.

3.1. Task and System Description

This paragraphs will introduce the setup of the cell and the objectives to achieve for the correct automatic assembly of a car door with its module. The module and the door are represented on Fig. 3. The positions of door and module could vary of few centimeters in translation and few degrees in rotation in each direction. The testbed is composed by three tasks:

1. Pick and insert door module: in the Pick phase we have to locate the door module by using visual and force information, pick it up and reach a reference position. Then the robot has to place the module into the door, and come back to reference position without detaching it.
2. Screw door module: consists of detecting, picking up, inserting three screws into three relevant holes to fix the module on the door.
3. Teach and assemble unknown door: the whole assembly has to work for a novel pair of module and door.

The hardware available at the facility in Stuttgart consisted of a lightweight collaborative robot (Universal Robots UR10) equipped with three sensors: a 3D sensor camera (PMD CamBoard Nano), a stereo camera (VRMagic D3), and a 6D force-torque sensor (Robotiq FT150). A vacuum gripper, composed by 6 suction cups, and a screwing tool (Weber Pluto 6D) were available and they could be automatically attached or detached from the robot flange by using a tool changing rack (Schunk SWS011). We replicated the setting in our laboratory, using as a basis the same lightweight collaborative robot, the Universal Robots UR10, but equipping it with different sensors. A PMD

CamBoard Pico has been used as 3D camera, a pair of Philips SPZ5000 webcams have been calibrated to work as stereo camera, while no force sensor has been mounted on the robot. A vacuum gripper and a screwing tool have been built by means of 3D printed materials to mimic the functionality available in the original system. The tools could be manually changed from one to another. A central working station connected through Gigabit Ethernet collects data coming from sensors and going to controllers. ROS is used as framework to enable interaction with the system. Topics, services, and actions coming from the system, as well as device drivers, have been established in advance to be able to use them in both settings. Positions and orientations of every relevant frame are published and updated by using the ROS TF package. A rough virtual model of the environment was also available and objects were coherently placed in the scene depending on the published positions. 2D and 3D cameras are provided with a default intrinsic and extrinsic calibration.

3.2. Methodology

The proposed tasks are connected to 3 main constraints: (i) limited time available for developing the solution, flexibility needed to deal with position tolerances and unknown modules and doors, (iii) usability and reliability of the teaching procedure. These characteristics lead us to propose a solution able to face both known and unknown door assembly in a very similar manner. In fact, different modules and doors rely on similar structures, and these features can be used as input for the framework. We want to extract these common characteristics to simplify and speed-up the new module and door identification. In order to successfully solve the previously described tasks we used the following pipeline: (a) learn the relative positions of each screw hole in both module and door; (b) learn the gripping and inserting trajectories through human demonstrations; (c) identify both module and door real positions by visual inspection; (d) pick and place of the module by transposing the learned motion to real position; (e) identify screw positions, (f) screwing.

In the learning phase, we collect template images for each part to identify. The idea is to extract relevant visual information in order to recognize them during the part inspection in order to build a coarse virtual model of the environment. A combination of Robot Learning from Kinesthetic Demonstration and Inverse Kinematics is used to learn how to pick and place the module. The variability and robustness of the system are granted by collecting several repetitions of the same action, performed by different subjects.

A visual system is used also for finding the screws and pick them up and fixing the module. A Template Matching approach is used for the screwing operation. Again, a door inspection is performed looking for screw holes positions. Once a matching has been identified, the system will align the screw with the hole by using the previously acquired template in order to perform the insertion.

3.2.1. Learning Phase

The learning phase mainly involves how to correctly pick the module and insert it in the door. These operations could be very easy or critical depending on the context and on the complexity of parts to be assembled. The module is composed by several elements including some flexible cables possibly assuming different configurations while the task evolves. The gripper provided for performing the picking action has a fixed base, while suction cups attached on its extremities can slightly change in position. Positioning the gripper in a consistent manner obtaining a robust layout is essential to assure a safe pick. This operation has to be performed in advance, since the surface of the module vary a lot and it is not always smooth. Moreover, the module is quite heavy while the suction cups do not have great gripping power in case of unbalanced loads. Selecting a wrong gripping position can determinate a loss in vacuum system and the consequential module falling due a displacement in weight or position of the suction cups. Once the module is properly picked, it has to be placed into the door. A series of coupling pins should be inserted into slots in the door to hold up the module, while avoiding cables to obstruct the movement. The motion should be executed precisely by placing the cables, and proceeding diagonally to insert the first coupling pin. Finally, the module should be straighten, and the rest of the pins could be set into place. Pinching the cable or failing to place a coupling pin lead to incorrect insertion and consequential falling of the module. Obviously, the placing trajectory is really dependent from the initial gripping position. Therefore, a successful picking does not correspond necessarily to a good performance in placing the module.

The described constraints could be easily met when the task is performed by a human being. Indeed, people have the capabilities to understand the task, test the selected strategy in few trials, and move the robot accordingly to fulfill the objectives. Nevertheless, recording a single execution is not enough for achieving a smooth generalized trajectory able to take into account to the intrinsic variability in the tasks. In order to obtain such mo-

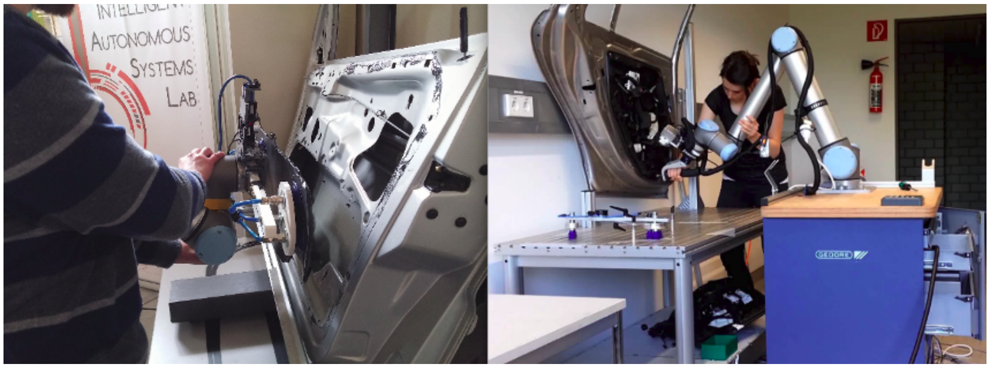

Figure 4: The robot is physically guided to learn the motion through Kinesthetic Teaching.

tion, we used a Robot Learning by Demonstration paradigm able to build a robust model of the movement starting from a series of demonstrations. In particular, we recorded data directly from the robot while a subject is free to physically guide the manipulator following the desired path (Fig. 4). In this way, it is possible to let the robot learn a novel task in little time and without the need of additional staff for robot programming.

In order to avoid unnecessary variability in the motion and reduce the number of examples we decided to keep human demonstrations as short as possible. Short trajectories are computed quicker resulting in a more standardized movement, while allowing a simpler and consequently safer robot activity. We mixed together Probabilistic Robot Learning with Inverse Kinematics to take advantage from both of them. The robot reaches fixed and safe positions close to the targets by using an Inverse Kinematics engine obtaining better performances in both reliability and time. The last part of the movement, namely the most complex one, is performed by using inferred trajectories computed through Robot Learning. Ten repetitions of the movement performed by different subjects has been recorded from an arbitrary selected initial position. The angles assumed by each joint while the robot is manually controlled have been considered for building the probabilistic model. Since door and module positions are not known a priori, a visual feedback is used to compute the actual configuration of the system before proceeding with the real picking and placing motions.

The raw data recorded from robot encoders have to be preprocessed in order to be able to generate a good probabilistic model. As a first step, they have been filtered to remove artifacts, such as periods in which all the joints

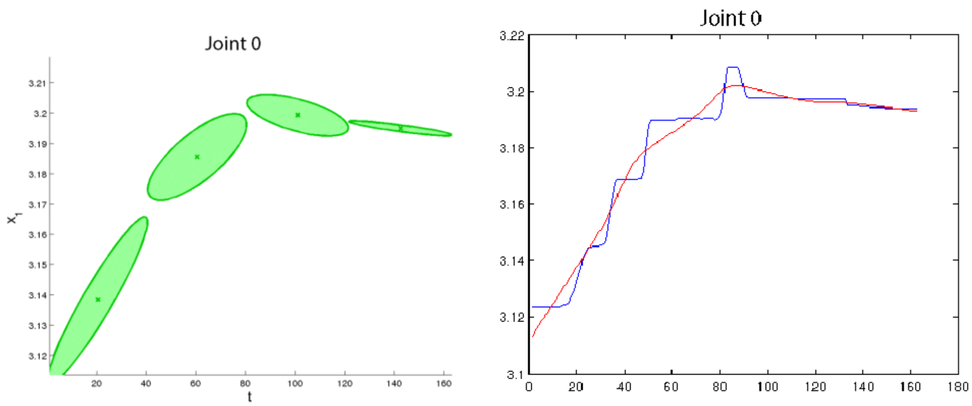

Figure 5: Modelization of Joint1 with GMM and continuous estimation of Joint1 angle retrieved with GMR.

were still. Doing so all the data not correlated with the movement have been eliminated, maintaining exclusively motion information. This process leaded to more robust and smooth trajectories while speeding up the creation of the model. Once the model is built, the final motion is estimated thanks to a regression technique.

Staring from GMM as accurately described in Sec. 2 and naming n the number of repetitions of the task, and T the number of observations acquired during each trial, the total number of data samples is $N = nT$. A single data in input at the framework $\zeta_j, 1 \leq j \leq N$ is described in Eq. 12.

$$\begin{aligned} \zeta_j &= \{t, \alpha(t)\} \in \mathbb{R}^D \\ \alpha_x(t) &= \{\alpha_g(t)\}_{g=1}^{G}. \end{aligned} \tag{12}$$

with G, number of joint bending angles; $\alpha_g(t) \in \mathbb{R}$, the value assumed from g^{th} joint bending angle at the time instant t; $\alpha(t) \in \mathbb{R}$, the set of values assumed from the considered joint bending angles at the time instant t; $D = G + 1$, the dimensionality of the problem. The resulting probability density function is computed as shown in Eq. 6. The Gaussian Mixture Regression (GMR) provided a smooth generalized version of every joint angle starting from the GMM (Fig. 5). Every joint angle $\hat{\alpha}$ and its covariance are estimated from the known a priori time instant t respectively using Equation 13 and 14.

$$\hat{\alpha} = E\left[\alpha \,|t\right] = \sum_{k=1}^{K} \beta_k \hat{\alpha}_k \tag{13}$$

$$\hat{\Sigma}_s = Cov\left[\alpha \left| t \right.\right] = \sum_{k=1}^{K} {\beta_k}^2 \hat{\Sigma}_{\alpha,k} \tag{14}$$

with β_k, the weight of the k^{th} Gaussian component through the mixture, $\hat{\alpha}_k$, the conditional expectation of α_k given t, $\hat{\Sigma}_{\alpha,k}$, the conditional covariance of α_k given t. The parameters (π_k, μ_k, Σ_k) defining the k^{th} Gaussian component are decomposed as shown in Eq. 9. The described framework could be used with known setting as well as with novel unknown door-module pairs. It gives good results both in time needed to teach the tasks and in robustness in reaching the goals. Nevertheless, it is hugely dependent from the information provided by the visual counterpart system.

3.2.2. Module and Door Identification

As before, we tried to adopt the same base procedure for both known and unknown objects. We decided to implement a reliable and adaptive identification system working for both door and module. A stereo camera has been used for obtaining a visual servoing procedure able to align the robot with respect to an object. A simple Image-Based Visual Servoing (IBVS) is really hard to implement due to lack of common robust set of visual features to use. On other hand, a Position-Based Visual Servoing (PBVS) approach supposes to know a priori a model of the object, not provided in this challenge.

However, our algorithm compensates this lack with the 3D knowledge coming from the stereo camera. The approach is composed by a training phase and an iterative query phase. In the training phase, the robot is placed into a known position ${}^{w}\xi_{cd}$, where both cameras could see the same region of interest inside the object. A stereo pair template is acquired and a keypoints extraction is performed by using *FAST* algorithm [26, 27] for the detection and *ORB* [28] as descriptors. A sparse triangulation is computed after a keypoints matching between left and right templates by using intrinsic and extrinsic camera parameters. In this way, a model representing the object is obtained from this set of 3D points. Finally, the object pose with respect to the camera frame ${}^{cd}\xi_o$ is estimated with a *Virtual Visual Servoing* algorithm [29] based on the 3D information of the object.

In the query phase, we used a single-camera approach. For each iteration, a template matching with the desired template has been performed in order to identify a valid area where extracting keypoints on the current frame (Fig. 6).

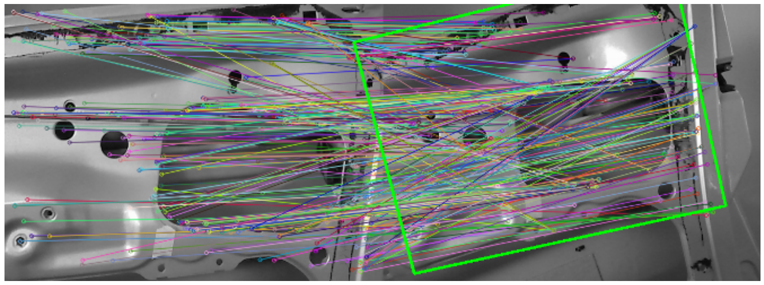

Figure 6: Keypoints extraction for the current frame with respect to the desired template.

Then, we matched current and corresponding keypoints by using the available 3D coordinates in the trained model in order to obtain the current camera pose with respect to the object ${}^{c}\xi_{o}$. The displacement between the current and the desired position could be easily computed as described in Eq. 15.

$$ {}^{c}\xi_{cd} = {}^{c}\xi_{o} * {}^{o}\xi_{cd} = {}^{c}\xi_{o} * {}^{cd}\xi_{o}^{-1} \tag{15} $$

After the robot movement the effective displacement could be different due to noise presence (Eq. 16).

$$ {}^{c}\tilde{\xi}_{cd} = {}^{c}\xi_{cd} + \Delta\xi \tag{16} $$

The procedure has been iterated to satisfy the condition in Eq. 17.

$$ \Delta\xi \leq \Delta\xi_{max} \tag{17} $$

where $\Delta\xi_{max}$ is related to the desired accuracy. A simple template matching in 2D has been used to find the screw and hole positions to perform the screwing task.

3.3. Results

GMM has been used as probabilistic model to predict angles of $G = 6$ robot joints. Good results can be achieved with $k = 10$ Gaussian components. Using few Gaussian components cause the generation of a too general model, unable to handle the variability of the signal. Contrariwise, if k is big the final model will be too complex. Experimentally, for module and door identification, an average of 6 iterations were needed to reach a precision of $1mm$ for translation and $0.2deg$ for rotation. The training phase allowed us to perform a PBVS task in an easy way, without any actual model of the

Figure 7: The robot is able to insert the plastic module into the door, both in Stuttgart and the laboratory facility.

object or a priori information. Therefore, the identification process for unknown objects becomes quite simple and immediate. In fact, the operations needed to perform the tasks have been restricted to (a) update the templates collected for the stereo camera pairs, and (b) recompute ${}^{cd}\xi_o$ through keypoints matching and triangulation. The query phase remained unchanged. We were able to correctly pick and place the module and the framework is robust to module shifts Fig. 7.

4. Manufacturing of electric motors

This task aims to increase the competitiveness in the European electric vehicles and motors manufacturing. Automation is already applied at different levels in this field, nevertheless it is facing strong competition from Countries with low labor cost. The interest for electric motors has increased in the last years to reduce the use of fossil fuels for environmental reasons, with the ideal goal to eliminate non-renewable energy resources in few decades. Increasing process efficiency would strengthen a very critical sector for Europe, as it is expected to garner $22.32 billion by 2022, registering a CAGR of 3.7% during the forecast period 2016-2022 [30].

Our aim is to reduce costs and increase flexibility with the following contributions: (a) important reduction of setup time and costs of the winding machine, thanks to the simplicity and flexibility of the proposed approach; (b) increase in the quality of the final motors, thanks to the increased amount of copper that the robot will be able to insert in each coil with respect to manual winding; (c) possibility to parallelize the winding operations, dramatically increasing production rate; (d) decreased number of defected cores,

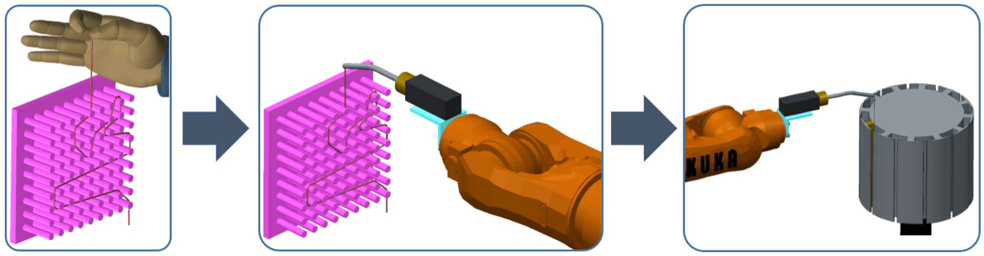

Figure 8: The 3 phases approach adopted for this case study, with (i) the human interface at the left, (ii) the lab scenario in the center, and (iii) the industrial scenario at the right.

thanks to an advanced quality inspection system; (e) reduction of environmental impact of the production process, thanks to a reduction of wasted copper wire.

An automatic system for coil winding has to be affordable to a wide range of users: from small-medium enterprises (SMEs), producing small batches of motors and frequently changing products design, to big companies, having a market request of several thousand standard units. The low flexibility of automated winding machines [31], i.e. the time and costs required to switch from one design to another, coupled to their high cost (up to 100k Euros), force small manufacturers (especially SMEs) to employ human operators in this task. The handcrafted job is obviously much more flexible, but more expensive (because of labor cost and equipment), and for the worker it is distressing, frustrating and repetitive. Few attempts of robotic cell for coil winding have been made [31]. We aim at achieving the product flexibility required for this business sector by developing an interactive robotic cell for this task. Such a reconfigurable cell has been provided with learning capabilities. The cell is suitable for winding the coils of several kind of electric machines, starting from the information of a simple teaching interface that can be easily used by operators without specific skills in robotics. This goal has been gradually reached passing through 3 main phases: (i) human interface, (ii) lab scenario, (iii) industrial scenario (Fig. 8). The need of an alternative power supply system for cars will be a crucial issue in the next years. Up to now, important steps forward have been made in the electrical motors. Factories like Tesla produce cars whose motors performances are comparable with traditional motors ones. Here, the goal is the development of an automatic tool, able to create autonomously an electric motor. Up to now, there are industrial machines able to wind up coils. These machines are very expensive and they are not flexible.

The electric machines manufactures have to deal with uncertain sales volumes. As follow, the batch sizes in the manufacturing process are varying. Furthermore, continuous efforts are taken by the industries R&D departments to develop optimized electric machines with increased efficiency, increased power density, decreased manufacturing cost, etc. This also leads to currently uncertain motor designs for its manufacturing processes. In the product lifetime its design may change several times. Also there is a trend in the manufacturing industry to work with minimum or even zero stocks. The products will be manufactured after receiving the orders. The development of flexible production technologies that can be adapted to varying motor constructions is an existing concern as long as manufacturing uncertainties still exist. The process related to the coils manufacturing and theirs transfer/insertion into the stator are addressed.

The concept of a flexible production will use a needle winding technique. The production process is divided into coils manufacturing and insertion of these on the stator. The coils are wound on frames, after which they are mounted onto the stator. For this particular application the winding process is restricted to concentrated windings. However, distributed windings or even complex winding schemes are achievable by winding individual coils. The proposed production process will have the potential to allow three dimensional shapes of the coils and complex winding schemes. Thanks to its flexibility, the process can easily adapt to new developed motors, without the need of expensive and time-consuming changing in the layout. Particularly, the removal of auxiliary special wire guides implies a reduction of setup times of the winding machine by 50%. The robotic-based system does not require any machining of new fittings for every new production batch. This will lead to an additional reduction of setup costs by 70%, mainly in terms of effort.

During the project, we faced the following challenges: (i) teach the robot how to properly wind the coils of stator/rotors; (ii) robotize the manufacturing process of electric machines, in particular the winding of coils on stator or rotor cores; (iii) detect and report non-compliances in the process of the coil winding. The selected electric motors have the following features: (a) frameless torque motors designed to be compact, high performance and cost effective; (b) allow direct coupling with the payload, eliminating parts of mechanical transmission; (c) maintenance free; (d) high energy NdFeB magnets maximize torque density. Main applications for the proposed motors are electric vehicles, machine tools, laser scanning and printing, motion simulators, rotary stage, robots, tracking systems.

4.1. Task and System Description

As anticipated, the case study has been divided in 3 phases.

4.1.1. Human-Robot interface

The first phase is focused on developing a solid learning system and a reliable human-robot interface. Indeed, the objectives of this phase were agreed in order to put the basis of the following ones. The approach will be used during tests in lab and industrial scenarios to finally wind up the stator coils of an electric machine. In this phase, the robot has to learn an arbitrary path. The operator teaches the selected path moving a tool in a natural manner to deploy the wire through a pin table, for a relatively low number of demonstrations. The user guides the copper wire through the pole grid by using a tool designed ad-hoc to maintain the wire constantly stretched, and prevent the possibility of knotting. The system records the covered trajectories by using a camera network composed by both 2D and 3D cameras. The camera network system has also the capability to monitor the workspace by detecting and tracking humans for safety and collaboration purposes. The idea is to have the robot to stop as soon as a human being is detected in a danger zone around the machine by the cameras. In order to demonstrate the consistency of the approach used, the system has to analyze a trajectory decided by a person not knowing the system. To select a pole to pass through, the operator has to roll up the wire twice around each choice.

4.1.2. Lab scenario

In the lab scenario (Fig. 9, the key robot action is the winding motion around a fixed point as an initial step towards the final goal to wind the coil of a real electric motor. Previous expertise in learning systems has been exploited to teach the robot how to unroll a wire following a specific path in order to pass a wire through a peg grid composing different possible routes, as shown by the operator. A 6 DoF robot manipulator, equipped with the same custom wire deployment system used by the human, has to replicate the motion of the operator and unroll a wire along the path taught in the previous phase. The result is considered correct if the robot is able to replicate the pole sequence in the exact same order selected by the operator. Both the operator and the robot starts from a fixed position. Moreover, the tool has to maintain the tension of the wire, while allowing the user to detach the tool for

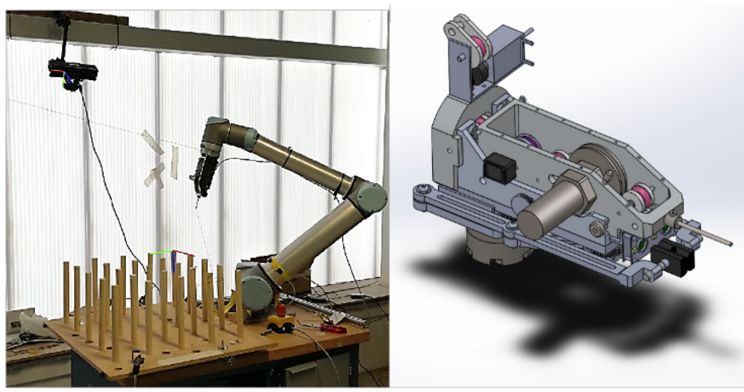

Figure 9: System setup for the lab scenario.

demonstrating the task and to attach it back on the robot when finished. No sensor is integrated in the end effector, since the only aim of the robot is to copy the operator motion. As the robot deploys the wire, particular attention has to been kept in order to prevent the wire from getting stuck or break. To do so, it is important that the robot performs a human-like movement, choosing a smart way for moving from one pole to another, avoiding useless change of direction or turnabout. These observations have been considered while developing the learning by demonstration framework since they are fundamental in the following phase.

4.1.3. Industrial scenario

In the industrial scenario, the objective is to plan the path for winding a coil to be mounted on a real stator composing an electric motor. The system takes into account the coils dimensions (height, width, and depth), the number of turns in the coil, the wire thickness and allowed tension. The planning is based on a learning framework previously developed in [11] [32] to improve the system developed in the previous phase to compute the trajectory to be covered by the robot tool and deeply explained in [33] and [34]. The learning system has been trained with several examples generated by using an initial set of human demonstrations, with the idea to improve the internal model in an iterative manner and increase the performances of the whole winding procedure. Of course, it is still possible to refine the computed trajectory by teaching the robot a better route through human demonstration. The idea is to enable the system to wind up a coil to be mounted on a stator never seen before. The parameters of pole dimensions, number of turns in the coil,

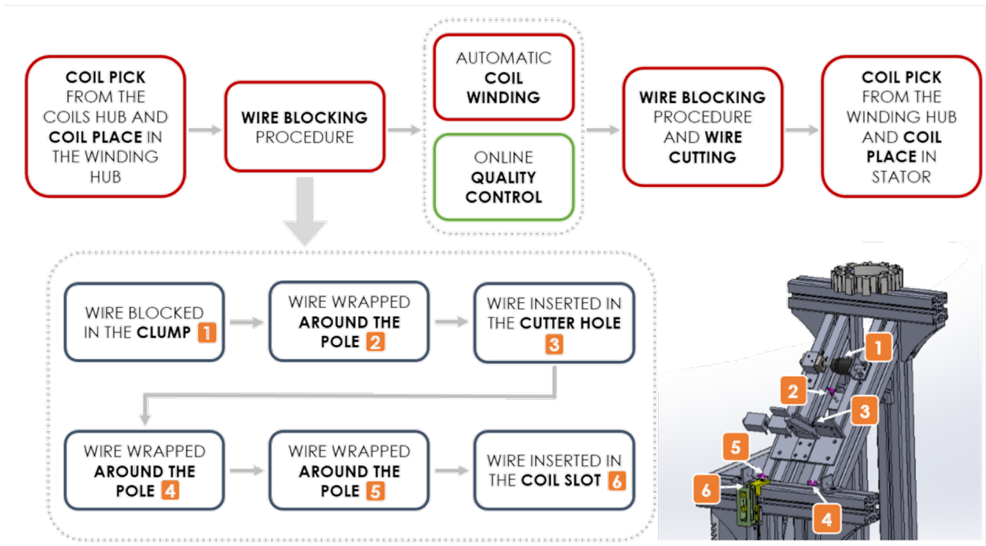

Figure 10: Sequence of automated winding procedure.

wire thickness and desired tension are provided as input to the robot by the operator, without the need of specific sensors to identify them. Based on the given information, the system chooses the proper coil from a coil hub and the robot tool gripper picks and places it on the adjustable winding stage. Later, the tool clamps its wire to the winding stage and starts winding the coil. A set of basic quality inspection protocols, based on turns count, wire tension and wire round distribution unity have been introduced, in order to guarantee a high standard of the winding process. A tension sensor has been integrated into the robot end effector in order to control the wire tension. The output of the sensor has been used to close the loop in the controller, adjusting the joint trajectories to match the desired output. This feature allows the robot to keep the wire tension as much as possible within the prescribed range, in order to avoid picks in the tension and reach optimal performances of the winded coil. Finally, the robot gripper picks the wound coil and places it on the empty stator slot. The process is summarized in Fig. 10. The robot working process has to be able to adapt to different sizes of poles and input parameters to control the winding process. The system does not need human intervention in wire handling, online quality control, pick and place. In fact, the robotic platform has been provided with automatic wire clamping and cutting, sensors for determining wire tension and wire rounds position, and

Table 1: Specifications of the motors produced by the manufacturer during the project.

Params	Unit	M1	M2	M3	M4	M5
Ext. diam.	mm	128	178	178	252	252
Inner diam.	mm	80	120	120	160	160
Act. length	mm	30	20	30	30	50
Rated power	W	1600	2400	3100	3000	4500
Conn. torque	N m	13	24	30	36	54
Peak torque	N m	30	53	69	82	124
Rated speed	rpm	1200	1000	1000	800	800
Noload speed	rpm	1500	1350	1200	900	850
Inertia	Kg m^2	0.09	0.03	0.032	0.055	0.06
Weight	Kg	3.7	5.65	6.65	8.5	10
Phase conn.		Y	Y	Y	Y	Y
N. of poles		14	20	20	20	20

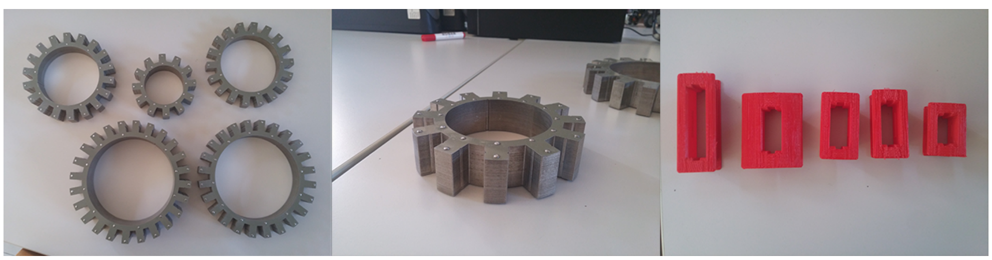

Figure 11: Stators and coils composing the motors used in the project.

a custom gripper for coil pick and place from the hub to the winding stage and from winding stage to the stator.

We worked on five different types of electric motors in order to designed and optimize a flexible production of the coil winding procedure. A stator consists of a laminated steel core in whose slots is located a three phase star connected winding. A rotor consists of a magnetic steel ring on which there are placed high energy permanent magnets. Applications for the proposed motors are electric vehicles, machine tools, laser scanning and printing, motion simulators, rotary stage, robots, tracking systems. The specifications of the proposed outer rotor frameless motors are reported in Tab. 1, while Fig. 11 shows some of the real components used in this scenario.

The system takes into account the coils dimensions (height, width, and depth), the number of turns in the coil, the wire thickness and allowed ten-

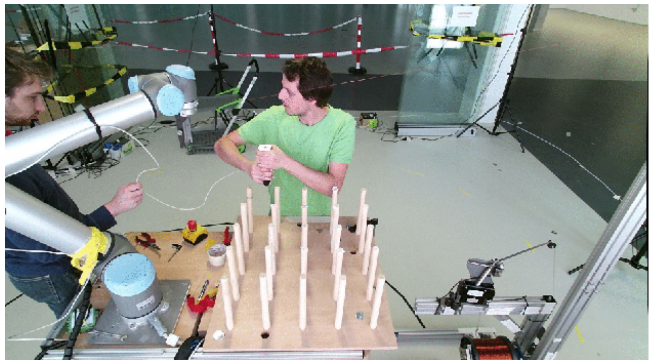

Figure 12: Human demonstration.

sion. These characteristics improve the system capabilities to compute the trajectory to be covered by the robot tool. In fact, the considered features are used to plan the path for winding a coil never seen before by the system. Of course, it will still be possible to refine the computed trajectory by teaching a better route through human demonstration. Novel demonstrations can be acquired by the learning system to iteratively improve its internal model and increase the performances of the whole winding procedure. The parameters of pole dimensions, number of turns in the coil, wire thickness and desired tension are provided as input to the robot by the operator, without the need of specific sensors to identify them. Based on the given information, the system chooses the proper coil from the coils hub and the robot tool gripper picks and places it on the adjustable winding stage. Later, the tool clamps its wire to the winding stage and starts winding the coil. A set of basic quality inspection protocols, based on turns count, wire tension and wire round distribution unity have been introduced, in order to guarantee a high standard of the winding process. A tension sensor has been integrated into the robot end effector in order to control the wire tension. The output of the sensor will be used to close the loop in the controller, adjusting the joint trajectories to match the desired output. This feature allows the robot to keep the wire tension as much as possible within the prescribed range, in order to reach optimal performances of the winded coil. Finally, the robot gripper picks the wound coil and places it on the empty stator slot.

4.2. Methodology

In our solution, we made Robot Learning and Inverse Kinematics work together in order to accomplish a more general and robust solution. The

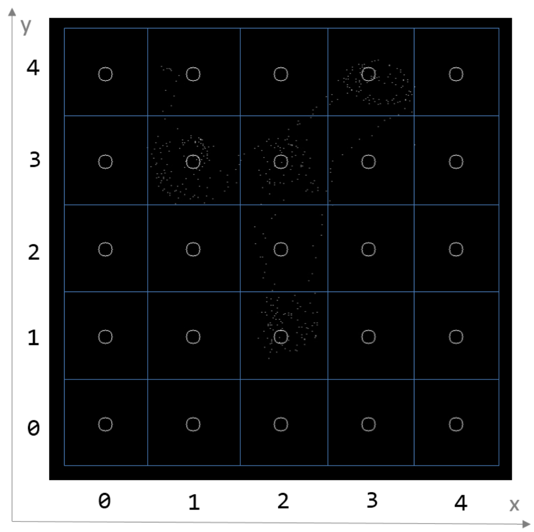

Figure 13: Trajectory grid.

basic idea is to take advantage of the human capability to find a solution by simply looking at the problems, while leaving to the robotic system to handle mathematical computation. For example, it is very easy for people to find the path to pass through a series of poles, while it is very difficult (or even useless) for them to compute the robot joint positions to guide the robot end effector along the same trajectory without interfering with the copper wire. Therefore, the set of useful information extracted from the trajectories performed by humans has been described by using a GMM [10], while a GMR has been used for retrieving an unified smoothed trajectory for the robot TCP. The learned trajectory has been translated from Task Space, in the tracking system reference frame, into robot Joint Space by means of a inverse kinematic engine.

During this work, we considered mainly three aspects in order to make the robot correctly reproduce the operator actions. Two of them are strongly related with the human-robot interface, the latter regards robot motion for both lab and industrial scenarios. At first, we need to identify the objects in scene and select the order of the operations. Second, we detected the best entrance and exit position for wrapping the wire. Third, we have to make the robot deploy correctly the wire.

4.2.1. Poles selection

While the identification of the objects like coils and stators (industrial scenario) has been performed following the same approach described in Sec. 3, a different procedure has been developed for the detection of the poles selected by the operator (lab scenario). A camera network has been used to track people in the scene and compute information about the trajectory when moving the wiring tool during demonstrations (Fig. 12). No special marker or material has been placed on the person or on the tool. An automatic tool has been developed to extract useful data from videos with almost no human intervention. Starting from the trajectory extracted from the camera network system, the goal is to detect which poles the operator selected and in which order. The information has been transformed and projected on the 2D plane, so the input data is a sequence of (x, y) coordinates of the tool position (Fig. 13). The selected approach is based on the consensus algorithm. The solution of the consensus problem is the result of the agreement among a number of processes (or agents). The result we would like to achieve is the pole selected by a person while deploying the wire. Basically, the consensus problem requires agreement among a number of agents for a single data value. Similarly, our poles selection algorithm seeks at which poles have been visited and on which order. Some of the processes could be unreliable since the visual system has estimated them wrongly, therefore our selection algorithm should be able to confirm the information coming from a single point of the trajectory by compering it with the others. In the same way, consensus protocols verify candidate values, and agree on a single consensus value.

In our consensus algorithm, adapted for this particular case, we start dividing the grid in different areas belonging to the ”nearest” pole without overlapping, so that every pole is in the center of a square. The idea is to assign to each pole an afferent area homogeneously distributed. After the grid division, we perform a sort of clustering operation, where each point is substituted with the relative pole area. Once we count the number of points belonging to each pole, a threshold helps in recognizing the selected poles, without mistakenly choosing poles where the tools passes often without selecting them. It is worth to notice that each pole can be visited only once in a specific trajectory. Considering the visit order helps in correctly detect the poles in the right order. A preprocessing phase is needed for remove the still periods in which the tool is motionless in a fixed point. This case

Algorithm1 Path detection

```
←trajectory      // sequence of (x,y) coordinates
visited_poles ← 0
n ←rows(trajectory)
np ← 25      // num poles
still points remotion
for all (x,y) ∈ trajectory do
        clustering      // ∀(x,y) ∈ trajectory find pole ∈ [1,np ] : (x,y) ∈ pole belonging area
for all p ∈ [1, np ] do
        cell_duration[p ] ←  ∑_{i=1}^{n} i ⇔ trajectory[i ] == p
        count_visits[p ] ← ∑_{i=1}^{n} i ⇔ trajectory[i ] == p
        mean_time[p ] ← cell_duration[p ] / count_visits[p ]
        visited_poles ← visited_poles + 1
count ← ∑_{i=1}^{np} count_visits[i ]
threshold ← floor(count / visited_poles)
fixed_treshold ← 40
if threshold > fixed_threshold then
        threshold = fixed_threshold
for all p ∈ [1, np ] do
        if count_visits[p ] > threshold then
                ordered insertion of p in final_path based on mean_time
```

Figure 14: Pole selection algorithm.

could alter the outcome, since it would result like many consecutive samples in the same pole area. The poles detection algorithm is described accurately in Fig. 14.

4.2.2. Entrance and exit position estimation

In order to avoid breaks of the wire or tangles it is important that the robot begin and end his winding motion in the correct , both for poles (lab scenario) and coils (industrial scenario). The correct place depends mainly on the previous position of the tool. For example, if the previously visited pole is above on the left with respect to the currently selected pole, the tool should come from left. Similarly, if the tool is placed on the right of a coil, the approaching direction to start the winding will be also right. This step is crucial to understand how to practically perform the winding. In order to do so, we exploit the human knowledge and expertise. Usually, a person is capable to understand which is the best way to reach a fixed point also dealing with constraints. Accordingly, we can take advantage of the human operator knowledge and overcome the planning limits.

Our goal is to compute the (x, y) coordinates of the start winding point and the end winding point. These coordinates are computed using a probabilistic framework. We use the start and the end winding point coordinates recorded from many winding tests performed by many different operators. The human expertise and knowledge give us the best way to overcome this critical issue in a probabilistic way. Furthermore, using many experiments performed by different subjects brings generality into the system, since every person could think to a different, although correct solution. The use of several executions allows the achievement of a generalized solution, which takes into account the intrinsic variability in the tasks. We obtained such results by using a Robot Learning by Demonstration paradigm, able to build a robust model of the coordinates starting from a limited number of demonstrations. Another vantage of this solution is that it could be used also by unskilled operators, since no further information or training phases are needed. The interesting coordinates are selected from tool trajectory tracked during the operator motion. GMM [10] Sec. 2 is used as probabilistic framework to predict the (x, y) coordinates.

In order to build the probabilistic model we introduces two fictitious positions in the system: *pos -1* and *pos* $+\infty$. The first one represents the robots starting point, while the second one represents the final goal, both far from the object to be winded. Considering data collected from S subjects, each of them completed the task T times and for each task he perfomed P different winding operations. The total number of data sample is $N = S * T * (P + 1)$. A single data in input at the framework ζ_j, $1 \leqslant j \leqslant N$ is described in Eq. 18.

$$\zeta_j = \{\alpha_w, \alpha_h, \beta_w, \beta_h, \gamma_x, \gamma_y, \lambda_x, \lambda_y\} \in \mathbb{R}^8 \tag{18}$$

with α_w, α_h respectively width and height of the previous position of the tool; β_w, β_h respectively width and height of the object to be winded up; γ_x, γ_y respectively x and y coordinates of the exit position; λ_x, λ_y respectively x and y coordinates of the entrance position. The resulting probability density function is computed as in Eq. 6.

The GMR provides smooth and generalized exit and entering points for the considered poles starting from the GMM. Every exit and entering points and their covariance are estimated from the known visited poles using Eq. 19

and Eq. 20

$$\{\hat{\gamma}_x, \hat{\gamma}_y, \hat{\lambda}_x, \hat{\lambda}_y\} = E\left[\{\gamma_x, \gamma_y, \lambda_x, \lambda_y\} | \{\alpha_w, \alpha_h, \beta_w, \beta_h\}\right] = \sum_{k=1}^{K} \eta_k \{\hat{\gamma}_x, \hat{\gamma}_y, \hat{\lambda}_x, \hat{\lambda}_y\} \tag{19}$$

$$\hat{\Sigma}_s = Cov\left[\{\gamma_x, \gamma_y, \lambda_x, \lambda_y\} | \{\alpha_w, \alpha_h, \beta_w, \beta_h\}\right] = \sum_{k=1}^{K} \eta_k \hat{\Sigma}_{\{\hat{\gamma}_x, \hat{\gamma}_y, \hat{\lambda}_x, \hat{\lambda}_y\},k} \tag{20}$$

with η_k, the weight of the kth Gaussian component through the mixture; $\{\hat{\gamma}_x, \hat{\gamma}_y, \hat{\lambda}_x, \hat{\lambda}_y\}$, the conditional expectation of $\{\gamma_x, \gamma_y, \lambda_x, \lambda_y\}$ given $\{\alpha_w, \alpha_h, \beta_w, \beta_h\}$; $\hat{\Sigma}_{\{\hat{\gamma}_x, \hat{\gamma}_y, \hat{\lambda}_x, \hat{\lambda}_y\},k}$, the conditional covariance of $\{\gamma_x, \gamma_y, \lambda_x, \lambda_y\}$ given $\{\alpha_w, \alpha_h, \beta_w, \beta_h\}$. The generalized form of the motions required only weights, means and covariances of the Gaussian components calculated through the EM algorithm.

4.2.3. Robot movement

Once we have obtained the entrance and exit coordinates to wind up the selected objects, we have the complete information needed in order to make the robot repeat the task by using inverse kinematics. We use Trac-IK [35] as inverse kinematic motor. The planner is RTT connect [36] from the OMPL library [37]. The planner includes an obstacle avoidance algorithm, in order to avoid obstacles already modeled in the ssytem. We use MoveIt [38] as interface for planning and visualization in a virtual environment. With our solution, once completed the preprocessing phase everything is handled autonomously from the robot inverse kinematics or Kinesthetic Teaching.

4.3. Results

Since the project is structured as a challenge, we needed to obtain the best results in the shortest time in order to gain a good score.

4.3.1. Lab scenario

In the lab scenario, a person shows 5 arbitrary paths previously selected from an external subject by moving the end effector in a 25 peg grid for a maximum time of 1 minute. A set of metrics has been selected to compute the performance of the system. *Metric I* measures the mean time needed to compute the information provided by the camera network system after the demonstration stops. *Metric II* measures the mean time needed to update

Metric	Description	Achievement	Worst case
Metric I	Time needed to extract the demonstrated trajectory	56.24s	58.21s
Metric II	Time needed to update the robot model	2.20s	2.21s
Metric III	Number of additional demonstrations	0	0

Table 2: Final results for lab scenario.

the model. Learning frameworks are usually based on probabilistic models built from a series of previous demonstrations called training set. An initial training set of 40 examples has been used as a basis to compute the robot trajectory. Anyway, it could not cover all possible paths, in those cases the model needs to be updated. Moreover, the operator should be able to check the validity of a novel demonstration as soon as possible. *Metric III* measures the mean number of examples needed to learn a selected path in addition to the initial demonstration. Targets for each metric have been selected by looking at expectation of the industrial partner and taking into account the state of the art in the field.

The time needed to compute the data recorded by the camera network was in mean 56.24s. We are able to provide an updated model starting only by the initial model and the data acquired during last demonstration in 2.20s. Nevertheless, the initial model has been always sufficient in order to compute the correct path during all the 5 different paths, resulting in 0 additional demonstrations. The system guarantees high success rate, high responsiveness and low effort for humans. In fact, even in the worst cases, we over performed the targets by obtaining 58.21s, 2.21s, and 0 additional demonstrations respectively for Metric I, II, and III. The results are summarized in Tab. 2.

4.3.2. Industrial scenario

The field-test was meant to prove the robot capability to work with different sizes and characteristics of the coils, and the correctness of both pick and place tasks. Tests concentrated on the winding and pick and place of the coils Fig. 15. A monitor shows wire tension during winding operations, in order to identify in real time picks that could break the wire. The performance have been tested in terms of productivity, repeatability, reduced manufacturing costs, flexibility, and setup time. At the end of each winding process, the coil has been compared with standards coming from actual industrial manufacturer by checking copper fill factor, inductances, resistances,

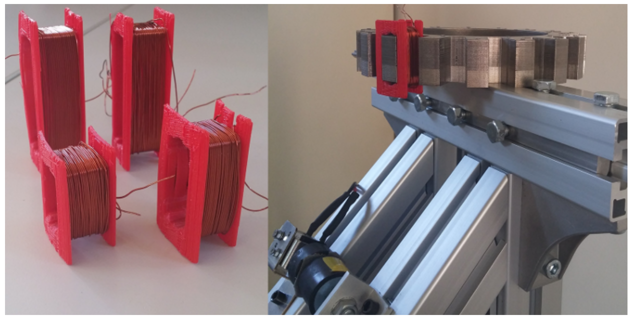

Figure 15: A set of wound coils ready for testing and a coil mounted on the stator.

Table 3: Results obtained during the testing in the industrial scenario.

Metric	Manufacturer	Our system
Number of stator wound (8 hours)	7	15
Correct wound coils	90%	50%
Mean Copper fill factor	0.2	0.5
High voltage	Pass	Pass
Resistance	Pass	Pass
Repeatability	20%	100%

and conductance at high voltage. In particular, the copper fill factor is the ratio of the copper conductors area over the total slot area. For a section of coil, the copper fill factor is computed as $CFF = \frac{nS}{bh}$, where n is the number of copper turns (conductors); S is the part of cross section composed by copper conductor, b is the base of the cross section of a coil slot, h is the height of the cross section of a coil slot. During the tests in Stuttgart, we were able to reach a Copper Fill Factor of 0.5. For electric motors used in standard applications, the copper fill factor is usually around 0.2. Moreover we achieved a valuable advantage in terms of productivity and repeatability. At the current state the production system in the manufacturer facility is able to wind 7 stators in 8 hours. In ours case the robotic arm provides 15 completed stators every 8 hours. Finally, the repeatability has also increased in a very significant manner due to the robotized approach. On the other hand, we faced a major problem with an increased number of faulty products. The results can be certainly improved with a more accurate tuning of the overall system, but some parts of the framework should be revised in order to avoid failures. A simple example regards the material used for the coils.

It was too fragile for the robotized procedure and sometimes it broke while winding the copper wire. The breaking problem could be avoided by 3D printing with a different material in order to find a solid structure. Anyway, it is worth to notice that the copper material (the most expensive one) used for the spare parts can be recycled in the very same process, and it had not been wasted.

Another very important aspect of our system is the high flexibility provided with respect to industrial winding machines available in the market. The capability to switch between different types of motors with minimum cost for additional tooling is essential. Commercial winding machines usually do provide very limited flexibility with expensive additional tools requested to wind different stators types. Moreover, the time needed for switching from one tool to another is quite long taking from some hours to a day. With our system, a complete change of the entire production from a motor type to a different one require more or less 15 minutes, reducing drastically the minimum number of pieces for a sustainable production, and opening the market to small-medium enterprises with a low margin of investment.

5. Conclusions

In this chapter, we presented a Robot Learning framework able to acquire information from the outer world and to generalize and understand it, obtaining in response a coherent robot motion. A robust preprocessing phase has been developed in order to clean and smooth the signals from variability due to noise or outliars. By removing artifacts, we have been able to highlight the peculiar common characteristics of the specific motion along several different trials. The refined signals have been used to train offline a probabilistic model, namely a Gaussian Mixture Model (GMM), able to represent the signal as a weighted sum of Gaussian components. A regression technique, namely Gaussian Mixture Regression (GMR), has been used to continuously estimate the joint bending angles to control a robotic device. We tested the proposed framework in two relevant industrial case studies: (i) the automatic assembly of a car door with its module, and (ii) robotize the manufacturing process of electric machines, in particular winding of coils on stator or rotor cores. In both cases, the proposed system can keep the costs low, while improving the production and the flexibility as part of the Industry 4.0 paradigm.

References

[1] M. Hägele, K. Nilsson, J. N. Pires, Industrial robotics, in: Springer handbook of robotics, Springer, 2008, pp. 963–986.

[2] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Robot programming by demonstration, in: Springer handbook of robotics, Springer, 2008, pp. 1371–1394.

[3] S. Calinon, A. Billard, Active Teaching in Robot Programming by Demonstration, Robot & Human Interactive Communication (2007) 702–707.

[4] R. Dillmann, Teaching and learning of robot tasks via observation of human performance, Robotics and Autonomous Systems 47 (2004) 109–116.

[5] D. R. Myers, M. J. Pritchard, M. D. Brown, Automated programming of an industrial robot through teach-by showing, in: Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, volume 4, IEEE, pp. 4078–4083.

[6] C. Baroglio, A. Giordana, R. Piola, M. Kaiser, M. Nuttin, Learning controllers for industrial robots, Machine learning 23 (1996) 221–249.

[7] P. Neto, J. N. Pires, A. P. Moreira, Highlevel programming and control for industrial robotics: using a handheld accelerometerbased input device for gesture and posture recognition, Industrial Robot: An International Journal 37 (2010) 137–147.

[8] W. Knight, This factory robot learns a new job overnight, `https://www.technologyreview.com/s/601045/this-factory-robot-learns-a-new-job-overnight/`, 2016. Accessed: 2019-07-13.

[9] T. Matsubara, J. Morimoto, Bilinear modeling of emg signals to extract user-independent features for multiuser myoelectric interface, Biomedical Engineering, IEEE Transactions on 60 (2013) 2205–2213.

[10] S. Calinon, F. Guenter, A. Billard, On learning, representing, and generalizing a task in a humanoid robot, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 37 (2007) 286–298.

[11] S. Michieletto, N. Chessa, E. Menegatti, Learning how to approach industrial robot tasks from natural demonstrations, in: Advanced Robotics and its Social Impacts (ARSO), 2013 IEEE Workshop on, IEEE, pp. 255–260.

[12] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, M. Hoffmann, Industry 4.0, Business & Information Systems Engineering 6 (2014) 239–242.

[13] B. Siciliano, F. Caccavale, E. Zwicker, M. Achtelik, N. Mansard, C. Borst, M. Achtelik, N. O. Jepsen, R. Awad, R. Bischoff, Euroc-the challenge initiative for european robotics, in: ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of, VDE, pp. 1–7.

[14] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society. Series B (Methodological) (1977) 1–38.

[15] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 6 (1978) 461–464.

[16] H. Akaike, Information theory and an extension of the maximum likelihood principle, in: 2nd International Symposium on Information Theory, pp. 267–281.

[17] A. Barron, J. Rissanen, B. Yu, The minimum description length principle in coding and modeling, IEEE Transactions on Information Theory 44 (1998) 2743–2760.

[18] C. S. Wallace, D. L. Dowe, Minimum message length and kolmogorov complexity, The Computer Journal 42 (1999) 270–283.

[19] S. Calinon, A. Billard, Incremental learning of gestures by imitation in a humanoid robot, in: Proceedings of the ACM/IEEE international conference on Human-robot interaction, ACM, pp. 255–262.

[20] F. Orabona, C. Castellini, B. Caputo, A. E. Fiorilla, G. Sandini, Model adaptation with least-squares svm for adaptive hand prosthetics, in: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, pp. 2897–2903.

[21] T. Tommasi, F. Orabona, C. Castellini, B. Caputo, Improving control of dexterous hand prostheses using adaptive learning, Robotics, IEEE Transactions on 29 (2013) 207–219.

[22] H. Chen, Y. Liu, Robotic assembly automation using robust compliant control, Robotics and Computer-Integrated Manufacturing 29 (2013) 293–300.

[23] M. Goyal, S. Netessine, T. Randall, Deployment of manufacturing flexibility: An empirical analysis of the north american automotive industry, Available at SSRN 2077659 (2012).

[24] P. Tavares, A. Sousa, Flexible pick and place architecture using ros framework, in: Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on, pp. 1–6.

[25] A. Vakanski, F. Janabi-Sharifi, I. Mantegh, A. Irish, Trajectory learning based on conditional random fields for robot programming by demonstration, in: Proceedings of the IASTED International Conference on Robotics and Applications, pp. 401–480.

[26] E. Rosten, T. Drummond, Fusing points and lines for high performance tracking., in: IEEE International Conference on Computer Vision, volume 2, pp. 1508–1511.

[27] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: European Conference on Computer Vision, volume 1, pp. 430–443.

[28] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 2564–2571.

[29] E. Marchand, F. Chaumette, Virtual visual servoing: a framework for real-time augmented reality, in: EUROGRAPHICS 02 Conf. Proceeding of Computer Graphics Forum, Saarebrucken, Germany, volume 21, pp. 289–298.

[30] S. Pal, Europe electric motor market by output power (integral horsepower (ihp), fractional horsepower (fhp) output), by application (industrial machinery, motor vehicles, heating ventilating and cooling (hvac) equipment, aerospace & transportation, household appliances, commercial applications) - opportunity analysis and industry forecast, 2014 2022, https://www.alliedmarketresearch.com/europe-electric-motor-market, 2016. Accessed: 2019-07-13.

[31] K. Feldmann, A. Dobroschke, S. Junker, Advanced processes and systems for the automated assembly of magnetic components for electrical machines, in: 1st CIRP-International Seminar on Assembly Systems, Stuttgart, pp. 83–88.

[32] E. Tosello, S. Michieletto, A. Bisson, E. Pagello, E. Menegatti, A learning from demonstration framework for manipulation tasks, in: ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of, VDE, pp. 1–7.

[33] F. Stival, S. Michieletto, E. Pagello, How to deploy a wire with a robotic platform: Learning from human visual demonstrations, Procedia Manufacturing 11 (2017) 224–232.

[34] F. Castelli, S. Michieletto, S. Ghidoni, E. Pagello, A machine learning-based visual servoing approach for fast robot control in industrial setting, International Journal of Advanced Robotic Systems 14 (2017) 1729881417738884.

[35] P. Beeson, B. Ames, Trac-ik: An open-source library for improved solving of generic inverse kinematics, in: Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on, IEEE, pp. 928–935.

[36] J. J. Kuffner, S. M. LaValle, Rrt-connect: An efficient approach to single-query path planning, in: Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, volume 2, IEEE, pp. 995–1001.

[37] I. A. Sucan, M. Moll, L. E. Kavraki, The open motion planning library, IEEE Robotics & Automation Magazine 19 (2012) 72–82.

[38] S. Chitta, I. Sucan, S. Cousins, Moveit![ros topics], IEEE Robotics & Automation Magazine 19 (2012) 18–19.