REMODEL

REMODEL - Robotic tEchnologies

for the Manipulation of cOmplex

DeformablE Linear objects

Deliverable 3.3 – Skill-based teaching by demonstration

Project acronym: REMODEL
Project title: Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects
Grant Agreement No.: 870133
ObjectsTopic: DT-FOF-12-2019
Call Identifier: H2020-NMBP-TR-IND-2018-2020
Type of Action: RIA
Project duration: 48 months
Project start date: 01/11/2019
Work Package: WP3
Lead Beneficiary: TECNALIA
Authors: TAU, UNIBO, TECNALIA
Dissemination level: Public
Contractual delivery date: 30/09/2023
Actual delivery date: 29/09/2023
Project website address: *https://remodel-project.eu/*

Table of contents

1. Task description: T3.3 Teaching by demonstration of skills for new assembly references and tasks 2
2. Introduction 3
3. Teaching-by-Demonstration Framework...........4
4. Enhanced Teaching-by-Demonstration 8
   4.1. Joystick-based Teaching 10
   4.2. EMG 12
5. Semantic Teaching 13
   5.1. Approach Overview 14
   5.2. Data processing...........16
   5.3. Experimental evaluation 17
   5.4. Robot's trajectory generation...........18
6. Conclusion 18
7. Demonstrator videos...........19
8. Publication list...........19

1. TASK DESCRIPTION: T3.3 TEACHING BY DEMONSTRATION OF SKILLS FOR NEW ASSEMBLY REFERENCES AND TASKS

Leader: **TECNALIA**; Participants: **UNIBO**, **TAU**, **TUM**, **PUT**

The robot will be programed using a skill-based system. Some skills are preprogramed in the robot and new skills can be created from the combination of already existing ones. This ability is particularly useful since in most of the cases the manufacturing tasks are not fully coded in the product design. This is the case of wiring harness manufacturing or assembly, in which even though the shape of the final product is given by the design, the sequence and the manipulation tasks to achieve the final product are managed entirely by the operator that manufactures the product itself, usually driven by its personal experience and by the requirement of the subsequent operations along the task. TECNALIA has already developed a skill-based system that will be used as a basis for this task. In order to execute the operations by the robot, and based on the developments implemented in tasks T3.1 and T3.2, the different skill-instances need to be combined and transformed into an executable program for the robot system. This will be performed through an easy programming framework in which the operator will teach by demonstration the robot by indicating some key positions and trajectories in order to do the parameterization of the different skill instances and adapt itself to each harness model. The key parameters need to be identified in each skill program and their variability from one harness model to another needs to be studied and evaluated. This approach enables a large adaption to different kinds of products in an easy and rapid manner necessitating few programming efforts.

To TRL 4: A first evaluation consisting in simulating the execution of skills (assembly of PINs, positioning of cables and key components on the manufacturing table) getting the cable assembly process key info from CAD (task 3.1) will be done. The parameters that are variable for the assembly of one harness to another are identified in the program as parameters that the operator can teach the robot; also, the parameter that could ease the assembly task. This includes information such as cable characteristics (colors, lengths…) and connector's information. Then the assembly sequence must be generated through a specific HMI of the CAD software, selecting the different components and clicking where to assemble them

To TRL 5: During this step, there is a concrete execution and implementation of the skills on an experimental platform. This implies the export of key data to the robot controller. The good execution of the sequence relies on the robot ability to locate the different cables, generate correct trajectories, and decide on end-effector tool changing if necessary. The robot should be provided in this case with a certain degree of autonomy in order to change and adapt its assembly plan when needed in order to successfully achieve the task. Safety strategies will be implemented when the robot human distance becomes too close. During this step, operators will test in a laboratory environment different teach by demonstration strategies in order to evaluate their intuitiveness.

To TRL 6: Different scenarios and tests corresponding to the selected use case will be validated and experimentally tested. The operator should be able to teach the robot new skills in a HW/SW environment that correspond to the use case requirements. The ability to automatically generate a skill network to execute a task by combining available skill instances will be evaluated.

2. INTRODUCTION

Task T3.3 aims to develop tools to assist operators in the creation of robot programs and skills using the Teaching-by-Demonstration paradigm. The main idea is to offer an intuitive way to program robots to inexperienced operators as in many cases programs need to be adapted and reconfigured, especially in complex scenarios such as the ones tackled in REMODEL. To this end, the Teaching-by-Demonstration paradigm enables the creation of new skills or program parts intuitively, guiding the robot directly to the desired points.

Therefore, a Teaching-by-Demonstration framework has been implemented in Task T3.3 based on kinesthetic teaching, where operators can guide manually the robot and create new movement sequences using a User Interface to interact with the system. Even so, during the application of this paradigm in the project's use cases, several limitations have been identified due to some particularities of the DLO manipulation and the scenarios. Accordingly, several additional developments and experiments have been carried out in an attempt to overcome the classical Teaching-by-Demonstration:

- An enhanced Teaching-by-Demonstration has been investigated, extending kinesthetic teaching with the use of a 6D joystick and an EMG (Electromyography) sensor.
- Semantic teaching is also proposed to work on a higher level of programming, overcoming the classical trajectory teaching approach.

The next sections will describe the Teaching-by-Demonstration framework developed within task T3.3, as well as the additional developments that try to overcome and enhance classical teaching.

3. Teaching-by-Demonstration Framework

Initially, the main goal was the development of a Teaching-by-Demonstration framework to teach trajectories to the robot focused on common tasks of the project such as the routing of cables. Although there are tools developed within the project (task T3.1) that make use of the CAD designs of the switchgear or wire harnesses for the trajectory generation, there are some cases where operators need to generate robot programs for some specific situations (e.g. changes in the design or extra actions on the manufacturing). Therefore, it is mandatory to offer simple and intuitive tools to generate new trajectories for non-expert operators.

After analyzing the current Teaching-by-Demonstration tools offered by robot manufacturers, the majority rely on a simple approach where operators need to guide the robot to the main task points (e.g. part grasping and release points) and record these positions. Even so, the cable routing process requires the recording of a huge number of points to reproduce precisely the path to carry out a suitable routing. Therefore, to offer a proper tool for the scenarios posed on the REMODEL project, the Teaching-by-Demonstration framework focuses on the following aspects:

- **Continuous data acquisition** for a precise replication of the routing movements
- **Intuitive User Interface** to interact with the system
- Capability to **record and reproduce trajectories**
- Capability to **adapt to changes in robot setup** such as the robot's tool TCP

Based on these specifications, a ROS-based framework has been designed and implemented based on the kinesthetic teaching paradigm. Initially, for the data acquisition phase, Figure 1 represents the schema of the framework, which includes the following modules:

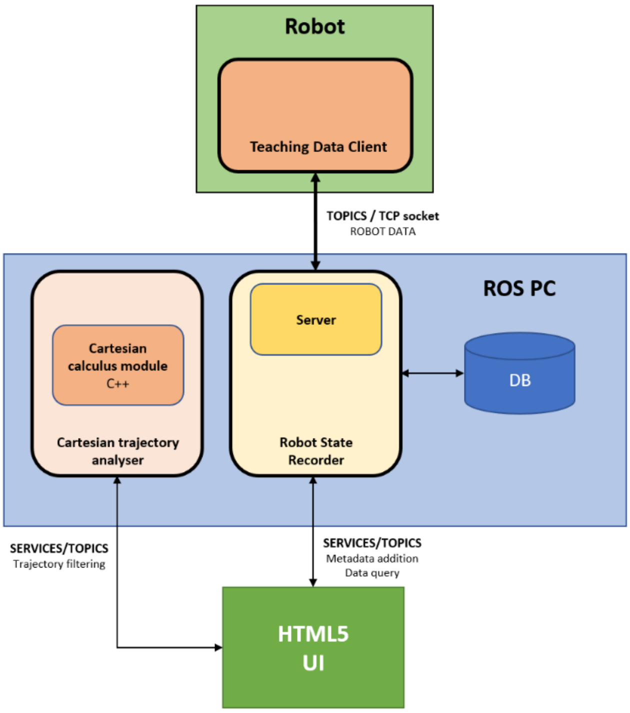
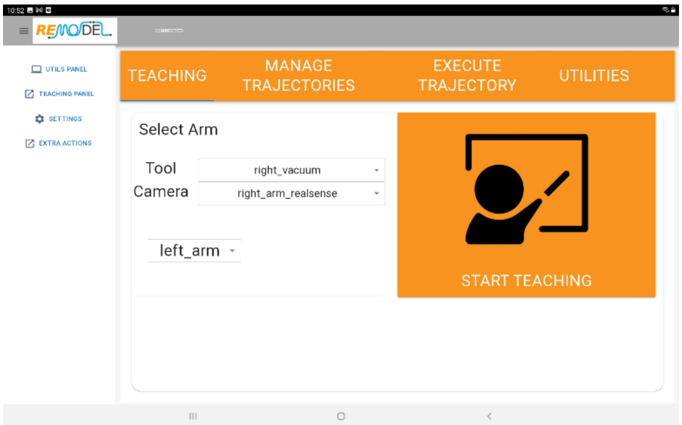


Figure 1 - Data acquisition schema

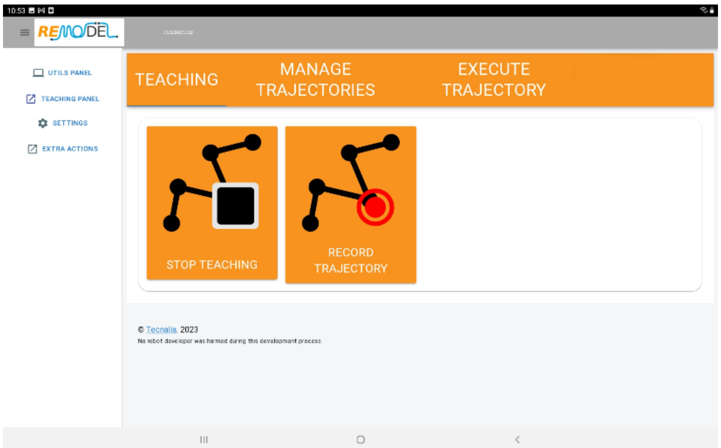
- **Robot State Recorder:** Central module in charge of recording the robot data. Based on the input commands received from the User Interface, the module can activate the gravity compensation through the robot driver and store the robot information on the database. Additionally, the module can manage additional information such as trajectory data or tool information.
- **Database:** This module implements an SQL database storing information about robot points (joint and Cartesian space), trajectories, and tools. Additionally, the module offers different ROS services to access the stored data.
- **Teaching Data Client:** The robot includes a driver with a data client that waits for connections from external devices (the ROS PC in this case). Once the connection is established, the module can activate/deactivate the gravity compensation mode and stream robot data with a frequency of up to 50Hz (although the standard data acquisition frequency is set to 10Hz).
- **Cartesian Trajectory Analyser:** This module offers different functionalities to manage and modify the Cartesian trajectories stored in the database. Specifically, it includes queries to downsample the trajectories (reduce the point volume) as well as filters to modify the trajectories' initial and final points.
- **User Interface:** The user interface, implemented in HTML5, offers a web interface to activate and use the different functionalities of the presented tool. The interface can be displayed and used in any web browser, allowing its use on PC as well as in portable devices such as a tablet.

The presented framework allows the recording of robot trajectories with a standard frequency of 10Hz, which has been found as sufficient for the posed application. To facilitate the recording process, the user interface offers an intuitive teaching process following the next sequence:

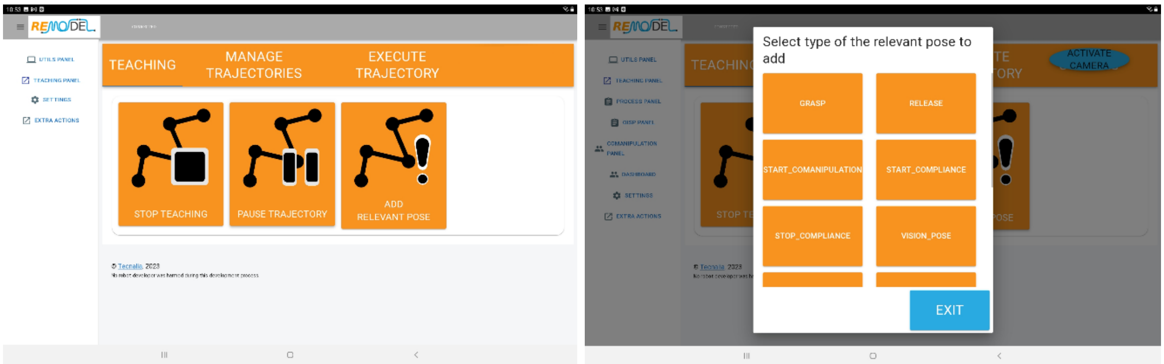
1. Start the teaching process, defining the specific tool attached to the robot and the specific robot arm (in dual-arm configurations). Additionally, the system also allows to define the camera attached to the robot although it is not used in the current version of the development. When the *"start teaching"* button is pushed, the gravity compensation mode of the selected robot arm is activated.



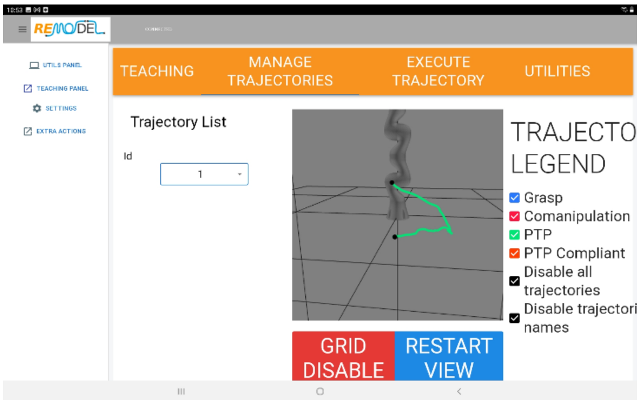
2. Once the robot is in gravity compensation, the operator can guide the robot to the initial trajectory pose and start the data recording.



3. During the trajectory recording, operators can insert a relevant pose at any moment, indicating if there is any specific action to carry out (e.g. gripper activation/deactivation). The system allows adding extra "relevant pose tags", opening the doors to new applications in the future with new requirements and functionalities.



4. Finally, the operator can visualize the taught trajectory to verify the acquired data.



Once the trajectories are recorded, the execution process makes use of a similar architecture although it includes several new modules, as shown in Figure 2. Specifically, the execution framework includes the next modules:

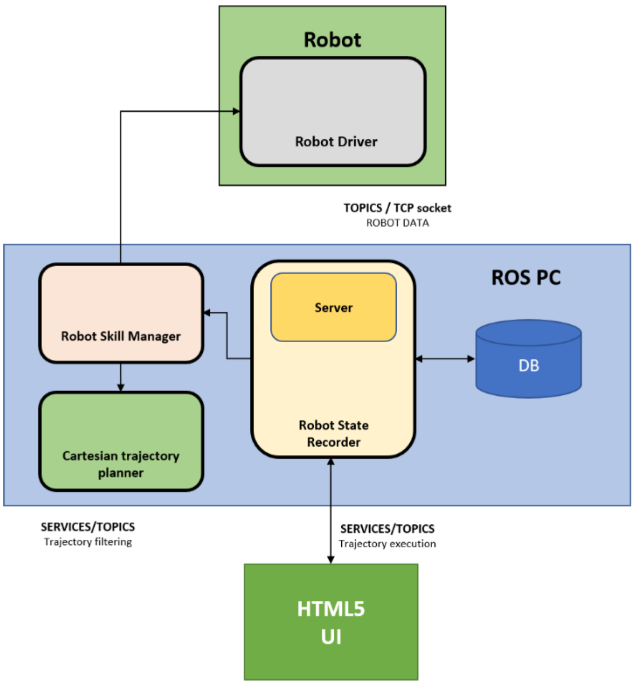
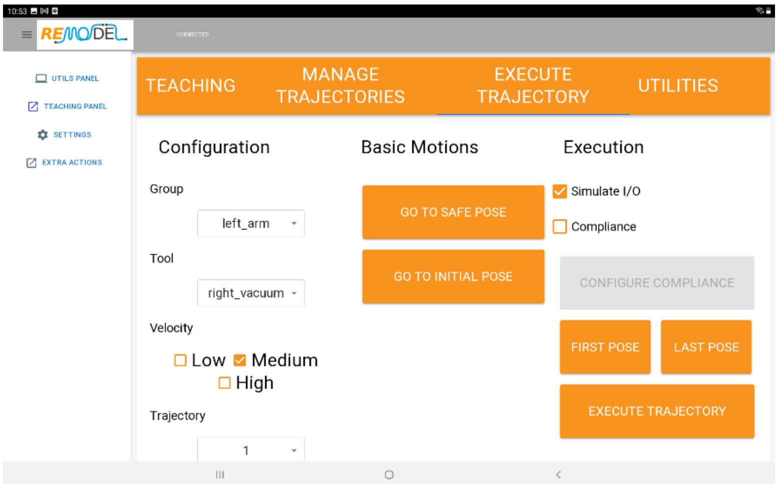


Figure 2 - Trajectory execution schema

- **Robot State Recorder:** Central module offering trajectory execution services. Initially, the trajectory points and tool data are retrieved from the database and are handled by the Robot Skill Manager for its execution.
- **Database:** The module offers different ROS services to access the stored trajecotry and tool data.
- **Robot Skill Manager:** This module receives the raw trajectory data, as well as the tool information to execute the trajectory in the selected robot. Initially, the module performs the required geometric transformations to ensure that the selected tool follows the recorded path. Afterwards, the calculated path is handled by the Cartesian Trajectory Planner to check if there is any valid robot trajectory in joint space. Finally, the plan is sent to the robot to be executed.
- **Cartesian Trajectory Planner:** This module generates a trajectory plan in joint space based on the provided Cartesian path. It makes use of the Descartes capability offered by MoveIt! package.
- **User Interface:** The user interface, implemented in HTML5, offers a web interface to execute the stored trajectories. This interface is intended for testing and validation purposes, allowing to define the trajecotry number, robot arm, tool, speed and additional control parameters.



These previous paragraphs summarize the Teaching-by-Demonstration framework defined for REMODEL, including both the trajectory recording and execution phases.

4. Enhanced Teaching-by-Demonstration

After the initial implementation of the Teaching-by-Demonstration framework, the application was tested for the cable routing task, a common job within the REMODEL project (Figure 3). During its use, three main drawbacks have been identified:

Figure 3 - Kinesthetic teaching in wire harness manufacturing scenario

- Kinesthetic teaching offers an intuitive way to guide and program robots, as operators only need to grasp the robot's tool to move it around the workbench. Even so, as the gravity compensation works so fluidly, it is complex to guide the robot describing simple and straight lines. These linear movements, in some cases, would be the optimal path to reach the target pose, as well as the most understandable and intuitive robot movements. In many cases, operators pointed out after the teaching phase that the obtained robot trajectory was a bit erratic and that they would like to repeat the teaching to obtain a cleaner trajectory. Therefore, it seems necessary to introduce ways to teach these clean and understandable trajectories.
- Additionally, in various scenarios posed in REMODEL there is little space in the robot setup to carry out proper kinesthetic teaching. For example, in the switchgear assembly setup, the dual arm setup with both robots around the device makes it too difficult to grasp the tool and guide it along the routing paths due to the lack of space. In the case of wire harness manufacturing, with a workbench of 1x2m, operators can not reach some parts of the work area, forcing them to climb on the table to complete the teaching in some areas. It would be necessary to add some mechanism to guide the robot remotely for some scenarios posed in REMODEL.
- During kinesthetic teaching, the operator needs to switch between the robot grasping (for the robot guidance) and the tablet placed in the work area. These changes caused a loss of focus, making it difficult to define the desired trajectory, especially in complex routes. Therefore, the investigation of new ways to introduce commands on the system would be celebrated by operators.

Therefore, it was decided to investigate two new items to enhance the Teaching-by-Demonstration framework:

- The use of a **6D joystick** to remotely control the robot during the teaching

- The inclusion of **EMG sensors** to introduce commands seamlessly in the teaching application

The next lines provide further information about both developments.

4.1. Joystick-based Teaching

To enhance traditional kinesthetic teaching, the initial idea is to introduce a mechanism to guide the robot remotely to allow teaching in narrow and uncomfortable setups. To this end, a 6D joystick has been included in the framework; specifically a portable 6D joystick (see Figure 4) that operators can place at any part of the cell. Additionally, one of the premises is that both kinesthetic teaching and joystick-based teaching must coexist, switching between both modes easily.

Figure 4 - 6D joystick for robot guidance

Initially, a complete control framework was implemented to allow an easy integration of the joystick, framework depicted in Figure 5. The implementation includes the next modules:

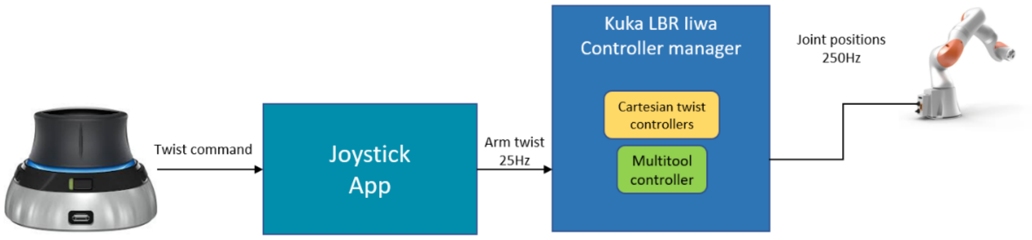
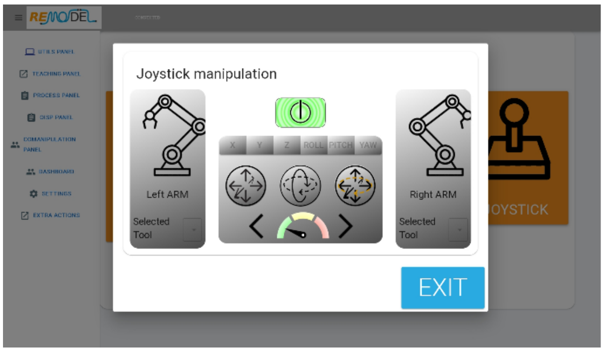
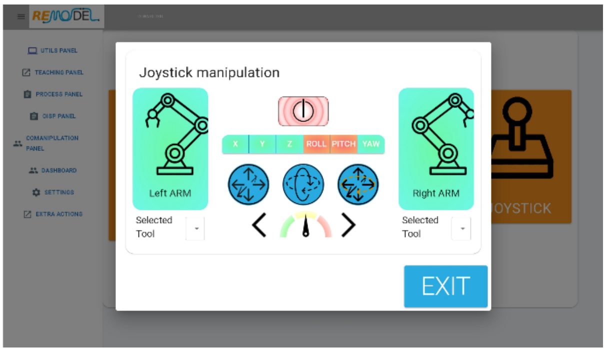


Figure 5 - Control framework for the 6D joystick

- At the lowest level, a **Cartesian twist controller** has been implemented following the ROS Control paradigm. Specifically, the **Multitool Controller** receives Cartesian twist commands and generates the next joint position command using the Jacobian. To enhance its possibilities, the controller offers topics to change the robot's TCP and reference frame in real-time. This controller is inserted in the robot's Controller Manager node, which sends position commands to the robot at a frequency of 250Hz.

- At a higher level, the **Joystick App** manages all the aspects of the joystick. To facilitate the use, and taking into account the dual-arm setup and the needs of the REMODEL tasks, the application offers the next options:
  - Activate the joystick, which internally changes the robots' controllers to the Multitool Controller.
  - Select the active arm, choosing between the left arm, the right arm, or both simultaneously.
  - Select the speed between low, medium, and high speed.
  - Select the active movement axis. As some of the teaching tasks require moving or rotating in some specific axis of the workspace, the application offers the possibility to "mute" the linear and rotational movements at any time.
  - An additional User Interface was developed to manage the aforementioned joystick options.





Once the Joystick Application was developed, it was necessary to update the Teaching-by-Demonstration framework to include the new joystick option. As the Joystick App can act as a standalone application to manage the robot arms through the 6D joystick, it was necessary only to modify the initial teaching interface (Figure 6) to mark if the operator is performing the standard kinesthetic teaching or joystick-based teaching.

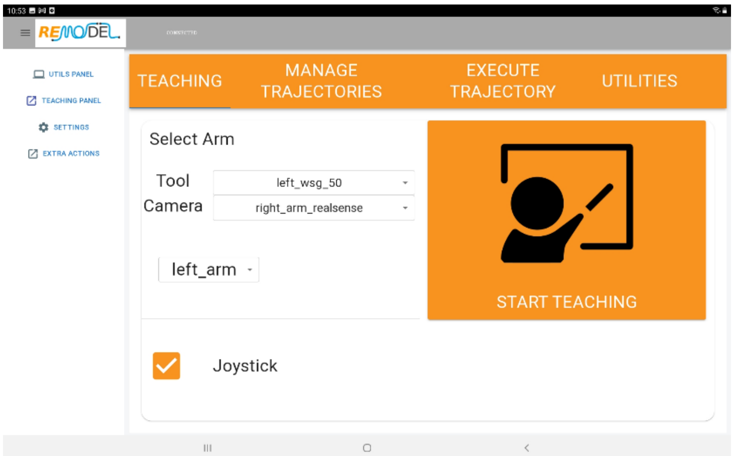


Figure 6 - Teaching-by-Demonstration UI with joystick activation

The next images show an operator teaching the robot to grasp and route cables using both the kinesthetic and joystick-based approaches.

Figure 7 - Operator using kinesthetic and joystick-based teaching

4.2. EMG

Existing literature on PbD mainly focuses on single modalities for robot trajectory demonstrations: observational and kinesthetic PbD. Real-world applications, especially in industry, often require piece-wise trajectory definitions from different modalities. For example, kinesthetic PbD is preferred for its simplicity and operator intuitiveness, while joystick-driven teleoperated PbD could be used for inaccessible segments due to workspace limitations. We therefore provide additional programming degrees of freedom for collaborative functionalities while demonstrating robot trajectories using a composition of both teleoperated and kinesthetic modalities without compromising any PbD capability.

To also address the challenge of providing users with additional programming degrees during PbD without requiring extra upper limb movements, we propose using wearable human-robot interfaces based on biological signals and cutaneous stimulations. Brain-Machine Interfaces (BMIs) have potential but lack sufficient reliability, while surface Electromyography (sEMG)-based neuromuscular interfaces offer stability and effective human motor intention estimation. Our approach leverages sEMG measurements of forearm's co-contraction level (CC-level), modulated by the user as an additional programming input via hand stiffening level changes. To assist users in this modulation, a vibrotactile feedback through a wearable coin motor is provided, regulating vibration intensity in real-time. Fig. 8 illustrates the proposed programming concept. Note that, in literature, various examples of using user impedance to teach robots have been explored, but, unlike these previous works, our framework uses sEMG to estimate the user's overall hand stiffness from forearm muscles, instead of an estimation of the arm end-point impedance. This allows providing an additional programming input for the user that can be freely modulated.

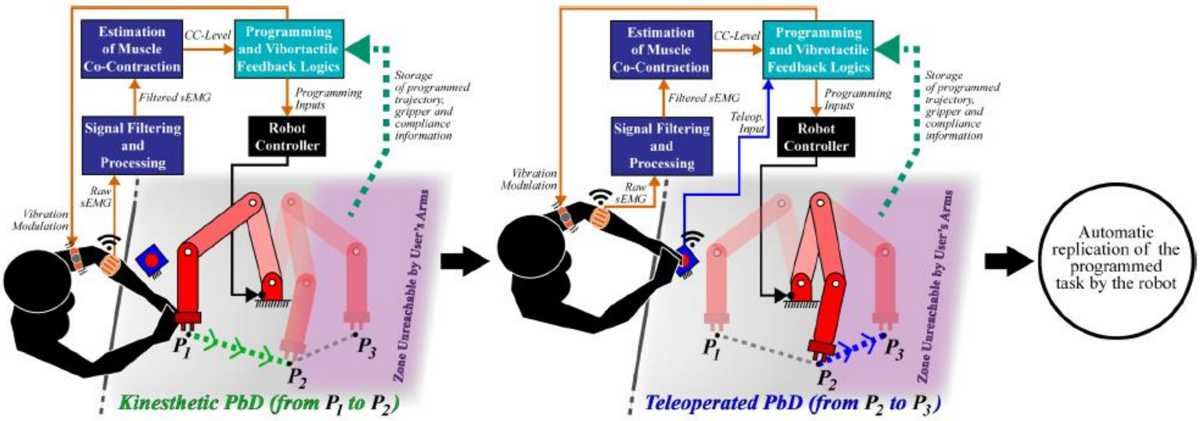


Figure 8 - PbD enhanced by sEMG

The performed experimentations (Figure 8) demonstrate that the framework enables the programming of two robot functionalities during PbD: manipulator compliance and gripper grasping. Participants programmed a robot trajectory via PbD, utilizing kinesthetic PbD for the first part and teleoperated PbD for the second part, the latter being inaccessible due to work table encumbrance/limitations. Meanwhile, they used the wearable interface to modulate the CC-level for grasping wires and adjusting robot compliance during the task.

Figure 9 - Experiments with the sEMG-enhanced PbD approach

Online automatic execution of the robotic wiring task achieved a 100% success rate, showcasing the approach practical viability. This approach paves also the way for even more versatile and user-friendly programming of collaborative robots, allowing them to perform complex tasks in various scenarios, ranging from industrial applications to service contexts with close human-robot interaction.

5. SEMANTIC TEACHING

All the content within this section has been adapted from [5]. For a more scientific and detailed explanation of the approach, please refer to this article.

5.1. Approach Overview

Programming by Demonstration (PbD) approaches can be divided in two main categories: PbD for low-level motion, which focuses on learning trajectories from the user's demonstration, and PbD for high-level task, which prioritizes extracting the semantic meaning of human actions to understand the goal of the demonstration. In REMODEL project these two lines have been investigated, developing approaches for both of them. While Sections **¡Error! No se encuentra el origen de la referencia.** and **¡Error! No se encuentra el origen de la referencia.** focus on the PbD for low-level motion systems, this section presents an approach for PbD for high-level task.

The aim of this approach is the extraction of the high-level plan for the demonstrated task, containing the sequence of skills to be executed, while the robot trajectories within these skills are automatically optimized by utilizing the information provided about the working environment layout and the characteristics of the manipulated object, which is provided by the CAD Platform (T3.1) (see Figure 10). Thanks to this, the programming responsibilities can be separated between the robot programmer (who develops the robot skills), and the task specialist (who demonstrates how to perform the task, or in other words, who demonstrates how to combine the existing skills to achieve the goals of the task).

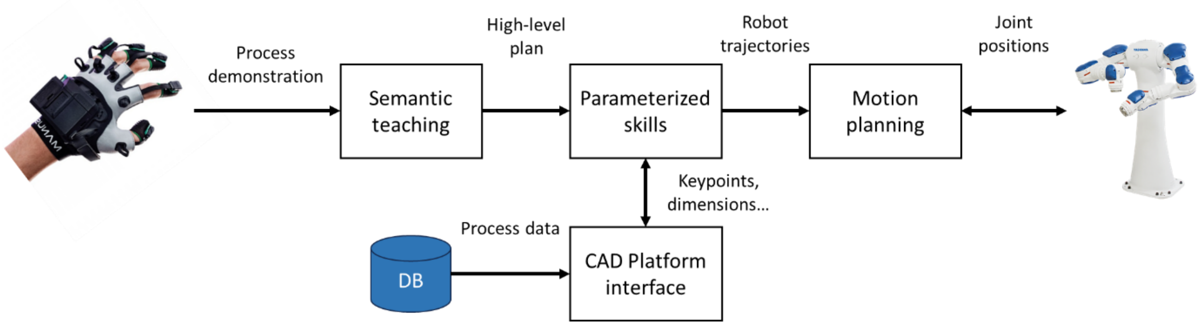


Figure 10. Integration of semantic teaching within the REMODEL system.

To improve the clarity and understanding of the developed approach, we have created an accompanying video[1], which complements the text offering a visual guide to all the steps and techniques involved. The video also illustrates the practical application of the approach, showcasing some of the performed experiments. We highly encourage the reader to watch it before continuing reading. Additionally, as stated before, a more exhaustive and technical analysis of the approach can be found in [5].

As it was previously introduced, the goal of this approach is understanding and digitizing single-arm manipulation processes performed by humans, based on the data captured by different kinds of sensors. The approach is sensor-agnostic, as long as these provide

[1] https://www.youtube.com/watch?v=nPZHHYW00rE

information about the user's movements and/or interactions with the manipulated objects. As the data captured from these sensors is simple and independent it can't be compared directly with a manipulation process. Therefore, the data collected from all of them has to be processed, and integrated, to increase its complexity level. Thus, the proposed methodology differentiates four manipulation levels according to the data complexity:

- **Sensor's data:** Raw data captured by the sensors, which depends on the sensor's type and resolution. For instance, the cartesian coordinates of the hand, or the bending angle of a finger.
- **Primitive:** Discretized sensor's data with a semantic meaning. They can be further categorized into different primitive's variables depending on the type of information they provide. Some examples of primitive's variables are: the hand rotations (e.g., +Yaw rotation, -Roll rotation…), calculated from the sequence of hand orientations, or the hand gestures, obtained by anlyzing the bending angles of all the fingers.
- **Operation:** Simplest manipulation action that can be performed with an object. It can be modeled as the combination of multiple primitive variables sequences. For instance, an operation could be grasp, screw, or insert.
- **Process:** Combination of operations performed with or on an object, in sequence, to achieve common goal. An example of a process could be a switchgear wiring connection, defined a sequence of indidual operations: grasping the cable, inserting it in a terminal block, screwing the connection, routing the cable through a wire collector, and so on.

The system is composed of five modules. Three of these modules are in charge of increasing the complexity level of the manipulation data. First, the Discretizing module converts the data captured by the sensors to the primitive level. Then, the Understanding module converts the primitives into operations utilizing Markov Models, and finally, the Sequencing module converts the segmented sequence of operations into a process. The last two modules are Training, which trains the Markov Models of the operations; and Interface and Calibration, which is used to guide de user and calibrate the sensors. An overview of the entire system can be seen in Figure 11.

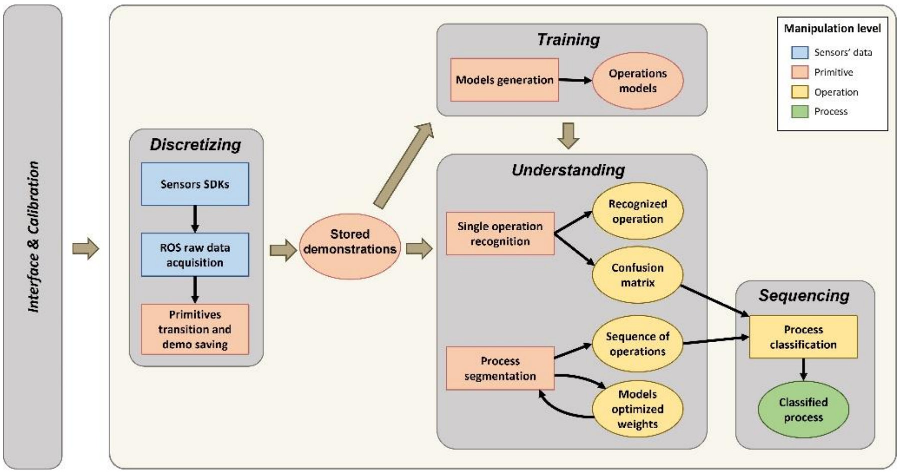


Figure 11. Semantic teaching system overview. Taken from [5].

5.2. Data processing

The biggest challenge in this approach is how to convert the sequences of primitives from different variables into operations. As stated previously, this is handled by the Understanding module, which exploits the concept of Markov Models. However, due to the complexity of the manipulation data, this problem couldn't be solved by using conventional Markov Models, and a novel approach named as Optimized Multiorder Multivariate Markov Model was implemented. These models combine Markov Models from different primitive variables, and from multiple orders, optimizing the weights of each of them for the operations of every object to maximize its recognition accuracy. This way, if the information of a variable is not relevant for performing a certain operation, its weight would be reduced, and just the other variable models would be taken into account. On the contrary, if the recorded sequence of primitives was not divided into different variables, if any of them were not relevant, these would affect the transitions of the entire demonstration and it wouldn't be possible to cancel their effect. For instance, to identify a bottle shaking operation, it would be interesting to pay more attention to the transition of hand motion primitives and less to the transition of finger movements primitives. The optimization of these weights is performed automatically by an iterative algorithm. Regarding the order of the models, the similarity between the recorded operation and its model typically decreases as it increases. This is because higher-order models are more sensitive to deviations from the ideal model. However, this increased sensitivity also leads to an increased similarity ratio between the most similar operation and the other operations, which can help to distinguish between similar operations. Therefore, the choice of the order of the model depends on the specific variable being modeled.

Considering all this, for an isolated operation recognition, the likelihood of the observed sequences of primitives (obs_j) is computed for each possible operation using its corresponding model, and the operation with the highest probability is then selected (op). Each operation (i) is characterized by a Multiorder Multivariate Markov Model (λ_i), which consists of multiple models ($\lambda_{i,j}$), one for each primitive's variable (j). Each one of these models has an associated weight (W_j). The order of the model for each variable is determined empirically. Thus, the following formula is used for the isolated operation recognition:

$$op = \underset{i}{argmax}\left(\sum_{j=0}^{len(files)} W_j \cdot P\left(obs_j \mid \lambda_{i,j} \right) \right)$$

The complexity of the problem increases when the operations are not isolated, but they are part of a process. In this case, as there is no information about the start and end point of each operation, the models can’t be applied directly and the use of a segmentation algorithm is required. The method followed is based on the sliding window technique, evaluating the probability of every possible operation along the demonstration timeline in windows of four primitive transitions. The first step is to identify the maximums in every operation's initial state probability timeline. Then, for every detected maximum, the average transition probability of every operation is calculated for the subsequent evaluation window (i.e., the average transition probability during the next four primitive transitions). If the average of the highest values in the window corresponds to the operation with the highest initial state probability and surpasses a threshold of 30%, then the operation is added to the segmented sequence. Additionally, when a new operation is detected, the hand coordinates during that evaluation window are analyzed to identify the location where it was executed. A more exahustive analysis of the implemented segmentation approach can be found in [5].

After this step, a high-level plan for the robot is extracted, which contains the list of operations to perform and their respective locations. However, not all these operations may be necessary to achieve the ultimate manipulation goal. This can be due to multiple reasons, such as segmentation errors, sensor inaccuracies, data limitations, or inefficient execution. To address this issue, the system includes a final Sequencing module. This module identifies the executed process based on the segmented list of operations. It does this by comparing all potential combinations of the segmented operation sequence (without altering their order) with the operation sequences of all the potential processes related to the handled object. The method also accounts for the probability of incorrect operation detection, utilizing the operations confusion matrix generated from the training data. Ultimately, the process with the highest similarity is selected.

5.3. Experimental evaluation

The system was tested for recognizing distinct operations and processes performed with five different objects: bottle, pliers, screwdriver, taping gun, and hammer. The sensing devices used to capture the user’s movements were a dataglove, which captures the hand orientation, fingers bending, and pressure in the fingertips, and a motion tracker, which captures the cartesian position of the hand. The system was trained with 40 demonstrations of each operation, performed by 4 different users. Figure 12 shows a user of the system recording a demonstration.

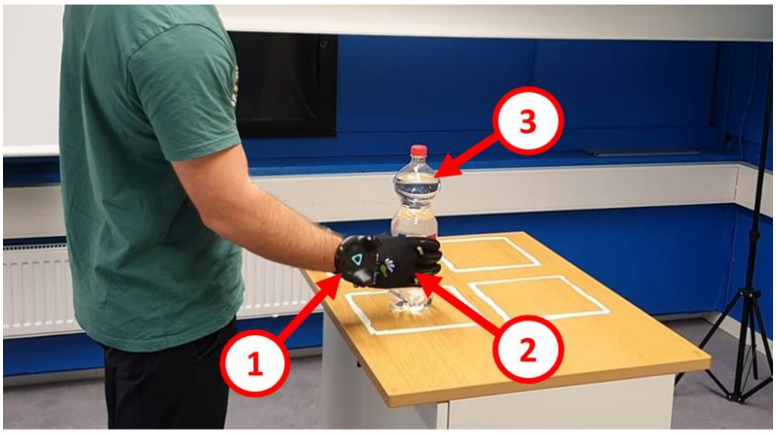


Figure 12. User recording a manipulation demo. 1: hand tracker, 2: dataglove, 3: manipulated object. Adapted from [5].

These experiments showed good results both for recognizing isolated operations and operations within a process. In the case of isolated operations, the average operation identification accuracy was higher than 70% for four of the five tested objects. Regarding the analysis of processes, the percentage of perfectly segmented processes was around 35%. However, these results improved significantly after analyzing the segmented sequences with the Sequencing module, reaching a process identification accuracy of almost 70%. This increases to more than 80% when five demonstrations of the same process are provided. For a more detailed discussion about the results, please refer to [5].

5.4. Robot's trajectory generation

The system described in this section extracts the semantics of a manipulation process, digitizing it into a high-level plan, which contains the sequence of operations to perform and the locations where to perform them. However, this still needs to be translated into commands and trajectories for the robot. An approach for this was presented in [6], where all the possible operations were predefined as parameterized functions, named skills. This way, the segmented sequence of operations would determine which skills to call, and the operation locations would be the parameters used to configure them, optimizing the robot's trajectory in every situation. In [6], these skills were programmed to optimize the robot trajectories based on the input parameters and information about the process and the geometry and distribution of the working environment provided by the CAD Platform Interface (T3.1). However, an alternative would be to combine this approach with the Enhanced Teaching-by-Demonstration presented in Section **¡Error! No se encuentra el origen de la referencia.**. This way, the sEMG-enhanced kinesthetic teaching would be used to define the trajectories of the individual skills, while the semantic teaching would be used to define the sequence of skills to be executed.

6. Conclusion

This document summarizes the work carried out in task T3.3. Initially, the developed Teaching-by-Demonstration framework has been presented. The application can record the robot's position with a frequency of 10Hz, offering a web User Interface to interact with the system and insert relevant points (e.g. grasping/release points) in the trajectory. For the execution phase, the application can retrieve all the recorded data, filter it and generate robot trajectories based on the user's parametrization, proposing a higher level of data management than in the classic approaches where only the final points of the trajectories are used in the execution.

Even so, during the tests carried out in the different use cases, several limitations were found in the application of the kinesthetic teaching paradigm. Therefore, some experiments have been carried out to investigate and explore new ways to enhance the classical Teaching-by-Demonstration. On the one hand, an enhanced Teaching-by-Demonstration has been explored making use of an external 6D joystick and an EMG sensor, which include additional ways to interact with the teaching framework. On the other hand, high-level semantic teaching has also been considered, in an attempt to overcome the classical point-to-point teaching.

These developments offered a functional tool for the teaching of applications and program parts for DLO manipulation, as well as new ways to face and enhance classical teaching solutions.

7. Demonstrator videos

Enhanced Teaching-by-Demonstration [LINK]

Teaching-by-Demonstration & EMG [LINK]

Semantic teaching: [LINK]

8. Publication list

[1] Meattini, R., Amerì, A., Bernardini, A., Gonzalez-Huarte, J., Ibarguren, A., Melchiorri, C, & Palli, G. (2023). Neuromuscular Interfacing For Advancing Kinesthetic And Teleoperated Programming By Demonstration Of Collaborative Robots. IEEE Transactions on Haptics. **(Under Review.)**

[2] Biagiotti, L., Meattini, R., Chiaravalli, D., Palli, G., & Melchiorri, C. (2023). Robot Programming by Demonstration: Trajectory Learning Enhanced by sEMG-Based User Hand Stiffness Estimation. IEEE Transactions on Robotics.

[3] Meattini, R., Chiaravalli, D., Galassi, K., Palli, G., & Melchiorri, C. (2022). Experimental Evaluation Of Intuitive Programming Of Robot Interaction Behaviour During Kinesthetic Teaching Using sEMG And Cutaneous Feedback. IFAC-PapersOnLine, 55(38), 1-6.

[4] Meattini, R., Chiaravalli, D., Biagiotti, L., Palli, G., & Melchiorri, C. (2021). Combining unsupervised muscle co-contraction estimation with bio-feedback allows augmented kinesthetic teaching. IEEE Robotics and Automation Letters, 6(4), 6180-6187.

[5] Malvido Fresnillo, P., Vasudevan, S., Mohammed, W. M., Martinez Lastra, J. L., & Perez Garcia, J. A. (2023). A Method for Understanding and Digitizing Manipulation Activities

using Programming by Demonstration in Robotic Applications. Robotics and Autonomous Systems. **(Under Review.)**

[6] Malvido Fresnillo, P., Vasudevan, S., & Mohammed, W. M. (2022). An approach for the bimanual manipulation of a deformable linear object using a dual-arm industrial robot: cable routing use case. IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS), Conventry, United Kingdom, 2022.