PROUD: A Hybrid Model-Based/Learning Approach for Symbolic-Level Robot Programming by User Demonstration

Joris De Winter[1], Cristian Alejandro Vergara Perico[2], Elahe Bagheri[1], Yudha Pane[2], Erik Hostens[3], Erwin Aertbeliën[2], Bram Vanderborght[1] and Wilm Decré[2]

***Abstract*— With the industrial need shifting more towards customized products in low volume, the relative cost of reprogramming robots is increasing, mostly linked to the high cost of programming engineers. This project aims at reducing that cost by making the programming task more intuitive, replacing the engineer by cheaper factory workers, who also have knowledge about the assembly task. The architecture combines information gathered from different possible sources, in order to learn an optimal assembly action sequence with minimal iterations.**

I. INTRODUCTION

In recent years prices of industrial robots have dropped significantly. Today, not the price of the robot but (re)programming the robot to perform a task has become the biggest expense [1]. These high prices are since expert robot programming engineers are needed for programming because existing interfaces today are lacking use-ability and intuitiveness.

Industrial robots are being used for large scale production with very little adaptability and flexibility requirements. However, many enterprises (both large and small) are feeling more and more the need for agile production, shorter product life cycles in small volume, increasing the need for these intuitive robot interfaces. As a result, these interfaces can make robots available for today's factory workers, who cost less (because they require less training) and often have more knowledge about the assembly task compared to programming engineers.

Cobots, that are more often being adopted in industry, are robots that are safe to work with in close proximity or in a collaborative setting. This property opens up the possibility for humans to program these cobots by physically interacting with them, called Programming by Demonstration (PbD). PbD can be achieved on two different levels, namely the *trajectory*- and *symbolic-level* programming [2]. On the trajectory level, PbD is concerned with programming specific robot movements by mapping a demonstrated trajectory onto a robot trajectory. On the symbolic level, demonstrations are represented by a sequence of symbolic actions. Focus in this project is on both levels, combining model-based and model-free learning techniques.

All authors gratefully acknowledge the financial support of Flanders Make through project PROUD SBO

[1] Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium, Core Lab Flanders Make, www.brubotics.eu joris.de.winter@vub.be

[2]Robotics Research Group, RAM Division, KU Leuven, Belgium, Core Lab Flanders Make, www.mech.kuleuven.be/robotics

[3]Flanders Make, Belgium, www.flandersmake.be/en

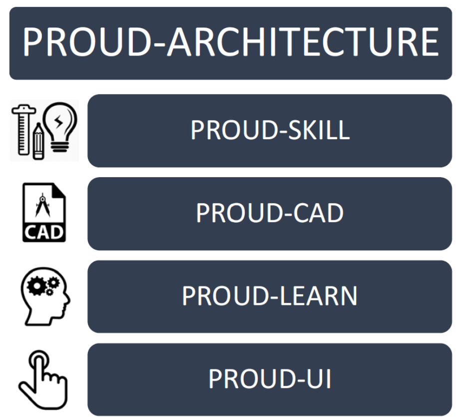

Fig. 1. The architecture of the PROUD system. It combines four different components, where each component provides a solution to a specific part of the entire programming task.

The goal of this work is to lower the bar to start programming a robot, by making abstraction of robot programming languages, and only using intuitive interaction techniques. As such, the factory worker, being skilled in the assembly process, will have the ability to "teach" a new behavior to a robot, and not only an experienced programmer. It is clear that this simplification must of course be developed in such a way that it does not come at the expense of increasing the programming time excessively.

II. PROUD ARCHITECTURE

The core idea of this work is to investigate a hybrid-framework that combines model-based techniques with interactive learning techniques. Model-based techniques require programming skills, yet are very interpretable and facilitate formal verification. Interactive learning techniques on the other hand, that here use natural human interaction as a form of communication, can make abstraction of these formal programming languages. These methods can step in when models are not available or expensive. The hybrid approach aims to combine the best of both worlds.

The improvements to the programming architecture are presented in Fig. 1, where each component provides a solution to a specific part of the entire programming task. In following paragraphs, each of these components will be further explained.

A. PROUD-SKILL

One of the main goals in robotic research is to expedite and facilitate the development of robot applications that are robust against uncertainties in dynamic environments, such as when an operator actively interacts with the robot during task execution.

To tackle this problem, as in [3], we combine learned and modeled information using the *expression graph based task specification language* (eTaSL) [4] to enable the specification of a set of reactive behaviors that take into account sensor inputs, such as trajectory generation, robot motion or collision avoidance. In this view, *PROUD-SKILL* will provide a library of skills defined as function of the proposed robot behaviors. These skills can be intuitively composed while maintaining the necessary robustness of the system to solve a large variety of robot task. Some examples of these kind of skills are grasping, moving, placing, pushing, following, inserting, drilling, among others.

As a result, robot applications containing free motions and motions in which the robot is in contact with an environment can be deployed while allowing reactivity, hence, increasing the system robustness against uncertainties of the environment.

B. PROUD-CAD

In industrial applications, CAD models are widely used for storing relevant workpieces, workcell and tasks information. This work investigates the exploitation of such information to enable more automated robot programming. Concretely, the PROUD-CAD component connects to the PROUD-SKILL by using CAD-derived information to aid the programmer in selecting, composing and parameterizing suitable skills for the given task. While CAD data is not sufficient to completely generate the appropriate skills, this module also facilitates manual annotation of CAD models with task-specific information. An example of this annotation is the intended assembly relation and the geometric features that this relation should be imposed on. The design and implementation of this part will continue our work in [5], where the use cases are limited to assembly tasks. In addition to these, PROUD-CAD will also take into account the possibility of programming reactive trajectory following tasks, which are relevant for industrial cases such as painting and polishing.

C. PROUD-UI

When we speak about user interfaces, we often mean a one-directional interaction interface. This project investigates not only what the robot can learn from the human, but also what the human can learn from the robot.

First, a multi-modal communication channel is being investigated, where the human uses speech in natural language and physical demonstrations to transfer their knowledge to the robot. These two channels were chosen because since the beginning of humanity we have used these to pass our knowledge on to the next generation. We therefore assume that this is an intuitive interface that we all can use.

On the other side, the robot can also assist the human in optimising their teaching. For this we are investigating an application of explainable AI, where the robot, based on the knowledge it has, can explain to the user why it is behaving in a particular way. This is especially useful for really complex assembly tasks where the user can use help in identifying the information that leads to a specific action, or where a new user needs an explanation to improve an existing robot program.

D. PROUD-LEARN

Lastly, PROUD-LEARN uses a combination of model-based and model-free learning techniques in order to learn an optimised behavior. The model-based aspects focus on skills. In each skill, we can differentiate between *given parameters* (e.g., coming from CAD or specified explicitly by the operator) and *learnable parameters* (e.g. parameters that can be deduced from a physical demonstration). Challenges lie not only in developing the appropriate learning algorithms, but also in monitoring the learning process, including safety constraints, selecting appropriate parameter bounds, and representing the results in a transparent way that facilitates explaining them to the user.

Model-free techniques will be used for learning robot behavior from human advice in natural language. Natural language is expressive and can be used by everyone in industry. The challenge lies in interpreting sentences into meaningful information and the ability to ground these sentences against the knowledge we have about the environment. Interactive Reinforcement Learning [6] is used to learn a correct robot action sequence based on the given advice. Learning is performed in an online fashion, meaning that the user can give additional advice while the robot is already executing the assembly task.

References

[1] B. C. Group, "Economic justification for industrial robotic systems," *International Federation of Robotics*, 2007.

[2] A. Skoglund, "Programming by demonstration of robot manipulators," Ph.D. dissertation, 01 2009.

[3] C. A. Vergara, J. De Schutter, and E. Aertbeliën, "Combining imitation learning with constraint-based task specification and control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1892–1899, April 2019.

[4] E. Aertbeliën and J. De Schutter, "ETaSL/eTC: A constraint-based task specification language and robot controller using expression graphs," in *IEEE International Conference on Intelligent Robots and Systems*, 2014, pp. 1540–1546.

[5] Y. Pane, M. H. Arbo, E. Aertbeliën, and W. Decré, "A system architecture for cad-based robotic assembly with sensor-based skills," *IEEE Transactions on Automation Science and Engineering*, 2020.

[6] J. De Winter, A. De Beir, I. El Makrini, G. Van De Perre, A. Nowe, and B. Vanderborght, "Accelerating interactive reinforcement learning byhuman advice for an assembly task by a cobot," *Robotics*, vol. 8, no. 4, 12 2019.