USING COLLABORATIVE ROBOTS AS A TOOL FOR EASIER PROGRAMMING OF INDUSTRIAL ROBOTS

I. M. Sarivan, A. Batuev, A. E. Ciontos, Ø. Holtskog, E. N. Sivertsen and C. Schou

Materials and Production, Aalborg University,
Aalborg, 9220, Denmark
E-mail: cs@mp.aau.dk

Programming industrial robots using traditional jogging via teach pendant is a time-consuming task that requires extensive training. More intuitive and faster task programming is often possible using kinesthetic teaching. Although this feature is available on many commercial collaborative robots, it is rarely available on traditional industrial robots. In this paper we propose a framework for allowing tasks to be instructed using a collaborative robot via kinesthetic teaching, and afterwards deployed to a traditional industrial robot. The framework consists of a physical modular concept for robot exchange, and a online programming software tool called Universal Industrial Interface. To assess the framework, a feasibility study is conducted where an industrial relevant task is first programmed using a collaborative manipulator, and afterwards deployed on an industrial manipulator.

Keywords: Robot programming, industrial manipulator, collaborative robot, kinesthetic teaching, cross-vendor compatibility, flexible production.

1. Introduction

Robotic manipulators are the backbone of modern industry and one of the main tools to have in an automated industrial environment. However, programming traditional industrial robots is both time-consuming and requires special training. Nowadays due to numerous factors such as dynamic market demands, shorter product-cycles and an ever increasing demand for customization, manufacturing equipment need not only to be efficient, but also flexible and reusable. Within robotics, one answer to this need is the introduction of collaborative robots which provide more intuitive instruction methods such as kinesthetic teaching. Several have shown that this enables non-robotics experts to engage in task programming.[1–3] This is considered desirable in a dynamic environment with frequent task adjustments and new product introductions since it empowers the on-site shop floor person-

H. Montes et al. (eds.), CLAWAR 2018: 21st International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines,
Robotics Transforming the Future

nel to manage the robots with less dependency on experts and engineers. However, in most existing manufacturing industries, it would not be practical to simply exchange the majority of the installed traditional industrial robots for collaborative robots only to gain the ability to do kinesthetic teaching. Additionally, traditional robots may also hold certain features or performance metrics which cannot be met by a collaborative robot.

Since the development of collaborative robots, one of the key benefits of kinesthetic teaching is the eased instruction, therefore we propose in this paper a framework utilizing collaborative robots as an instruction tool for tasks which can later be deployed to a traditional industrial robot. Thus, a collaborative robot is installed and used to intuitively instruct a given task through a cross-vendor robot programming tool. Enabled by a modular mechanical concept, the collaborative robot is afterwards exchanged with a traditional industrial robot which performs the task execution.

The proposed framework consist of a modular mechanical concept enabling quick and precise exchange of robot manipulators, and a custom software tool called Universal Industrial Interface (UII) providing a cross-vendor graphical user interface (GUI) for robot programming.

Further on, this paper is structured as follows: Section 2 presents related work, Section 3 presents the design and user experience of UII and Section 4 describes the modular mechanical concept used for efficient exchange of robots. These are followed by a preliminary feasibility study and the conclusion in section 5 and 6 respectively.

2. Related Research

With the introduction of the *collaborative robot*,[4] robots are now moving into the dynamic environment of the human workers with more frequent task changeover, several ad-hoc tasks and less structure. A key challenge here is the need for more frequent changeovers, resulting in an increased need to reprogram the robot. In this context, faster and easier programming approaches are needed as opposed to traditional online programming using teach pendants.[5]

Biggs and MacDonald[6] present a survey on robot programming methods and find two overall approaches; manual and automatic programming of robots. In the latter, cognitive methods are used to make the robot more autonomous and thus less dependent on instruction details. It often relies on planning algorithms based on sensor inputs and a comprehensive world model. Contrary to reducing human intervention, recent research on manual programming instead focuses on making online robot programming easier

and faster.[3,7–11] This decreases the need for high-wage engineering hours and allows the shop floor operators to channel valuable process knowledge and experience into the instruction.

A common approach to ease online instruction of collaborative robots is to use kinesthetic teaching. Several researchers have already demonstrated the potential of using kinesthetic teaching for eased instruction of collaboration robots,[2,8,12] and the technology is also featured on several of the commercially available collaborative robots; e.g. Universal Robots, KUKA iiWA and Rethink Robotics Baxster.

Although several commercial collaborative robots offer kinesthetic teaching, they also exhibit vendor-specific user interfaces and programming languages. This increases the complexity in allowing shop floor operators to program robots in a company with multiple robot vendors. It furthermore prohibits the transfer of an already instructed robot task from one robot to another robot of a different vendor.

To overcome this issue, several has proposed third-party robot programming tools with cross-vendor capabilities.[10,11,13] Schou et al.[11] present a comprehensive tool for easy instruction of collaborative robots. The tool uses robot skills to allow robot instruction using task-level operations rather than device level commands. The tool also features cross-vendor compatibility achieved using the Robot Operating System (ROS).[14] Paxton et al.[10] also present a comprehensive tool for creating tasks for collaborative robots based on ROS. The tool combines kinesthetic teaching with explicit, object-centered instruction arranged in behaviour trees.

Compared to the tools presented above, we propose a light-weight, cross-vendor tool providing robot programming for both collaborative and traditional industrial robots. Our tool allows tasks instructed using kinesthetic teaching on a collaborative robot to be deployed on a traditional industrial robot. The programming is done on the same abstraction level as current teach pendants. Thus, the need to learn and adapt to various vendor-specific teach pendants is removed, however, the familiarity of traditional online programming is preserved. Furthermore, our tool is designed to be light-weight with the potential of being deployed to low-cost, small-scale computational units, e.g. Raspberry Pi.

3. Universal Industrial Interface

Central to the proposed framework is the software, Universal Industrial Interface (UII), developed by the authors. This tool is a cross-vendor robot operating tool with options to both program and execute industrial

robot tasks. It supports both kinesthetic teaching and traditional jogging. As such, UII supports programming of both industrial and collaborative robots. However, in this paper we focus on the exploitation of kinesthetic teaching on a collaborative robot and deploying it on an industrial robot. UII is designed to be applicable to the majority of commercial industrial robots and grippers. It is designed to provide task programming using interaction on the same level as traditional online programming. As such, the same motion commands as found on most traditional industrial robots and grippers are available; e.g., move linear, move joint, open gripper and close gripper.

3.1. *System Architecture and Implementation*

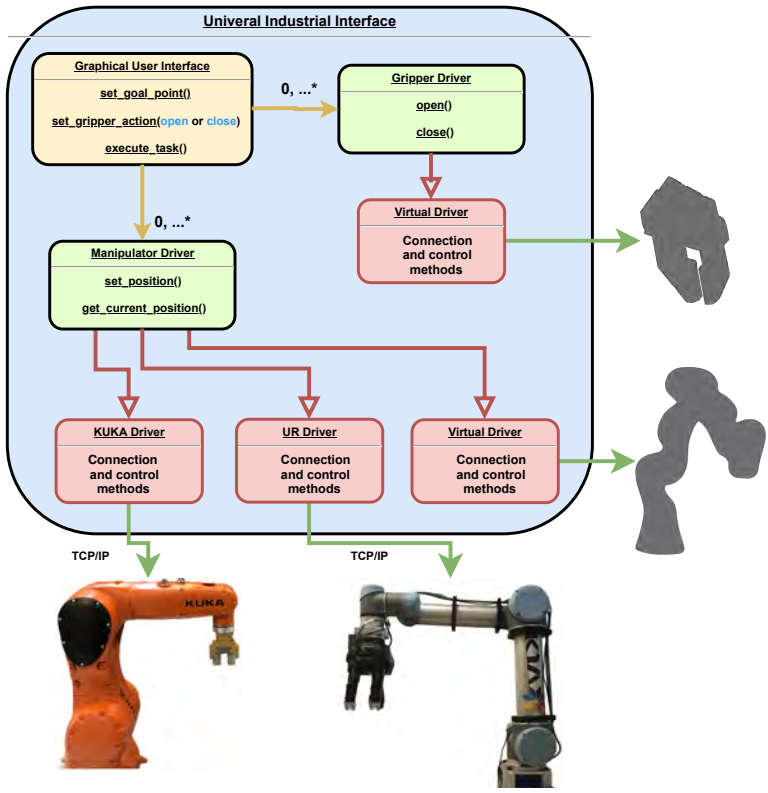


Fig. 1. Simplified UML class diagram of UII architecture.

UII is designed to be light weight and cross-platform. It is written in Python and will run on both Windows and Linux computers. It supports connection to robot manipulators and grippers using TCP/IP connections through the local area network (LAN). Figure 1 shows the overall architecture of UII.

3.1.1. *UII GUI*

The GUI provides a high level, intuitive interaction between the operator and the manipulators. The GUI enables the operator to set a sequence of goal points and gripper actions which can later be executed on a collaborative or industrial manipulator. The GUI is directly integrated inside UII without the need of a special API, as this component will stay unchanged regardless of the device connected to UII.

3.1.2. *UII Core*

This element contains the core functions of UII. This includes engines for task programming and task execution, task file management and system state monitoring. The UII core is implemented as a number of classes.

3.1.3. *UII Drivers*

The driver layer of UII provides connectivity to the devices utilized by UII. Thus, any specific driver provides the means to control and communicate with a specific robot manipulator or gripper. This is done by using the devices own protocol of sending and receiving data using a TCP/IP connection. To ensure the possibility of adding more manipulators and grippers in UII, an application programming interface (API) was designed which has to be implemented when building a driver. This API is implemented as a generic driver for a robot manipulator and a generic driver for a gripper. As seen on figure 1, the specific drivers (red boxes) all inherit from either a generic robot manipulator driver or gripper driver (green boxes). Figure 1 contains a simplified UML diagram of UII. The generic drivers provide a set of generic device functions for manipulators and grippers in a generic syntax. The generic device functions are used by the programming and execution engines in the UII core. When a generic device function is invoked by either the programming or execution engine in UII core, the specific device driver instantiated will translate the generic device function into the specific syntax and format of a particular device. This translation includes

converting between different units, angle representations and parameter sequences.

3.2. *Task programming and execution*

The UII GUI is designed to be as simple and intuitive as possible while still providing the essential features to program and execute tasks. It provides the necessary means for the operator to program a task by using both kinesthetic teaching and jogging. During programming, the operator can quickly guide the robot kinesthetically to any desired position. Afterwards, that position can be fine tuned using the jogging buttons of the GUI. When the desired position is reached, the set goal point button stores the point. Gripper actions can be added in between the robot motions by pressing the assigned button on the GUI. The UII GUI is shown in Figure 2.

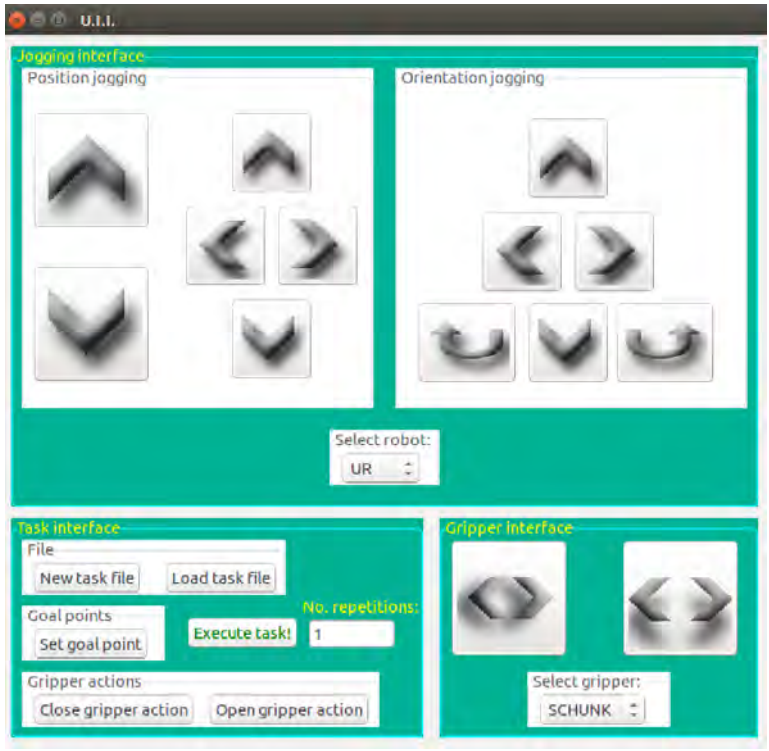


Fig. 2. UII Graphical User Interface

Once the programming is completed, the obtained robot program is stored in a UII-specific file format. It can now be executed again by loading the task and pressing the execute task button on the GUI. Given the cross-vendor capability of UII, the obtained task can be executed on any connected hardware. Thus, the option to use collaborative robot for task programming and a traditional industrial robot for execution arise. However, it only makes sense to execute the task on a different setup if the robot is installed in the same work-space in which the task was programmed. This also requires accurate positioning of the robot to ensure correct task execution.

4. Modular Mechanical Interface

To allow a task programmed using a collaborative robot to be executed on an industrial robot, it is essential that the two robots can be exchanged efficiently and accurately. To achieve this, our proposed framework adopts a modular hardware concept proposed by Schou and Madsen.[15] This concept is build around a modular table-top with standardized pallets for mounting robots, fixtures, feeders and other equipment, see Figure 3.

With this concept, each robot manipulator is mounted to a pallet providing it with the interface defined on the table-top concept. A similar "adapter" flange is used between the tool and the robot manipulator, see Figure 4. As a result, only four M8 bolts are to be removed in order to exchange a robot manipulator with another. It should be noted, that only manipulators with a weight and size suitable for table-top applications are applicable in this hardware concept.

5. Preliminary Feasibility Study

A feasibility study is conducted in order to examine the presented framework and demonstrate the procedure of programming a task using a collaborative robot and afterwards deploying the task on a traditional industrial robot. Thus, the purpose of the feasibility study is to prove that kinesthetic teaching can be used as an alternative programming tool for traditional industrial manipulators.

The task used in the feasibility study is inspired from a real industrial task described by Madsen et al.,[16] where parts exiting a processing machine on a conveyor must be picked up and placed into a container. In our simulated setup, the parts (cups) are to be picked from a mock-up conveyor and placed into a cardboard box, see Figure 5.

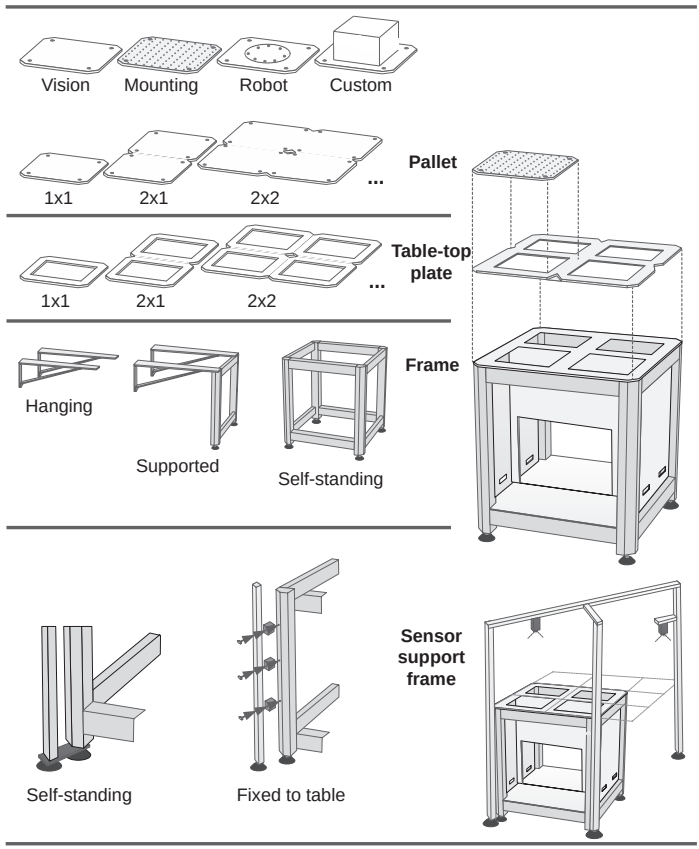
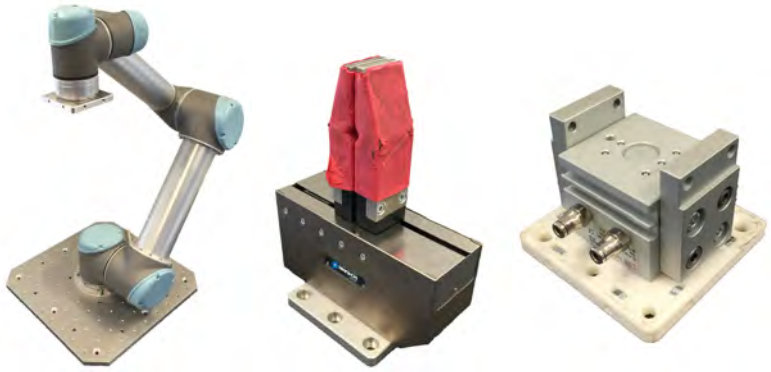


Fig. 3. Modular table-top concept used for quick exchange of robots. Concept proposed by by Schou and Madsen.[15]

The task was first programmed on a Universal Robots UR5 collaborative robot equipped with a Robotiq 3-finger electric gripper. Once programmed and verified, the UR5 was dismounted from the table-top, and replaced with a KUKA Agilus KR6 R700 sixx industrial robot. The generated program was then without further refinement or system adaptation executed on a KUKA Agilus KR6 R700 sixx. It is important to note, that the UII software does not need reconfiguration or rebooting.

As part of the examination of the framework, the programming time, the exchange time and the execution time was recorded, see Table 1.

It should be noted, that the exchange time is measured for exchanging the Universal Robots UR5 with the KUKA Agilus KR6 R700 sixx. However,



(a) Universal Robot UR5 (b) Schunk WSG50 (c) MetalWork P4-16

Fig. 4. Examples of robots and grippers adapted to fit the modular hardware concept.

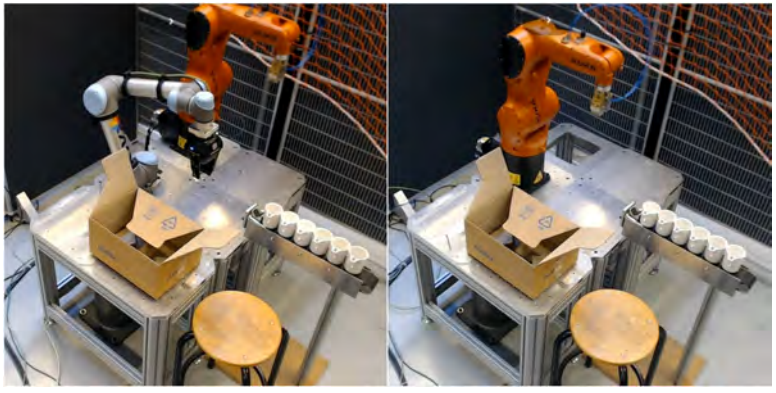


Fig. 5. The test setup before and after the exchange of robots.

Table 1. Test parameters used for evaluating the feasibility of the system.

Timing of programming:	1min 20sec
Timing of exchange:	2min 13sec
Timing of execution on Universal Robots:	12sec
Timing of execution on KUKA:	8sec

in cases where an existing robot cell has to be reprogrammed, an exchange of the traditional robot with the collaborative one prior to programming must be considered as well. Here, the exchange time is estimated to be the

same. For timing of execution, only one repetition was measured for both robots. As expected, the industrial manipulator is faster.

6. Conclusion

This paper presents a framework for using a collaborative robot as a programming tool for industrial robot tasks with the purpose of reducing programming time and effort.

The framework features a cross-vendor robot operating tool called UII. This software provides an intuitive, simple GUI for programming tasks of both collaborative and industrial robots. In the case of collaborative robots, it support kinesthetic teaching, which several has concluded to be reduce the required skill-level in robot programming. The framework also includes a modular hardware concept enabling quick and accurate exchange of robots and grippers. The concept is designed as a table-top robot cell, and thus only support robotic manipulators of a certain size and weight. However, the majority of collaborative robots are also found in this "small" segment of robots.

To examine the proposed framework, a preliminary feasibility study was conducted. A simple pick and place task was instructed using kinesthetic teaching and UII, and afterwards executed using an industrial robot. The feasibility study demonstrates how the proposed framework successfully allows a collaborative robots to be used as programming tool. In conclusion, we find the framework feasible, but further assessment and development is needed.

In future work, we will conduct a series of user studies in order to assess the skill-level required to use the framework. This will include users with different background and robotic experience. Furthermore, we will also verify the framework on more complex and diverse tasks. This includes the integration of additional devices, as well as improving cross platform compatibility. The integration of vision could allow for object-recognition and possibly improved grasping techniques. A throughout safety study would also be beneficial for a system such as UII.

UII is currently running of a PC, either laptop or desktop. However, deploying UII on a portable device (tablet, phone, etc.) could ease the use within a robot cell with little free space. We intend to develop a UII into a portable user interface, and a back-end running of a micro-computer, e.g. a Raspberry Pi.

References

1. C. Schou, J. Damgaard, S. Bøgh and O. Madsen, Human-robot interface for instructing industrial tasks using kinesthetic teaching, in *44th International Symposium on Robotics (ISR)*, (IEEE, 2013).
2. S. Wrede, C. Emmerich, R. Grünberg, A. Nordmann, A. Swadzba and J. Steil, *Journal of Human-Robot Interaction* **2**, 56 (2013).
3. K. R. Guerin, C. Lea, C. Paxton and G. D. Hager, A framework for end-user instruction of a robot assistant for manufacturing, in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
4. J. E. Colgate, W. Wannasuphoprasit and M. A. Peshkin, Cobots: Robots for collaboration with human operators, in *Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition - Atlanta, GA, USA*, 1996.
5. T. Owen, *Assembly with robots* (Prentice Hall, Inc.,Old Tappan, NJ, Jan 1985).
6. G. Biggs and B. MacDonald, A survey of robot programming systems, in *Proceedings of the Australasian conference on robotics and automation*, 2003.
7. Y. Maeda, T. Ushioda and S. Makita, Easy robot programming for industrial manipulators by manual volume sweeping, in *2008 IEEE International Conference on Robotics and Automation*, 2008.
8. A. Muxfeldt, J.-H. Kluth and D. Kubus, Kinesthetic teaching in assembly operations – a user study, in *Simulation, Modeling, and Programming for Autonomous Robots*, eds. D. Brugali, J. F. Broenink, T. Kroeger and B. A. MacDonald (Springer International Publishing, 2014).
9. M. Stenmark and J. Malec, *Robotics and Computer-Integrated Manufacturing* **33**, 56 (2015), Special Issue on Knowledge Driven Robotics and Manufacturing.
10. C. Paxton, A. Hundt, F. Jonathan, K. Guerin and G. D. Hager, Costar: Instructing collaborative robots with behavior trees and vision, in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
11. C. Schou, R. S. Andersen, D. Chrysostomou, S. Bgh and O. Madsen, *Robotics and Computer-Integrated Manufacturing* **53**, 72 (2018).
12. R. Caccavale, M. Saveriano, A. Finzi and D. Lee, *Autonomous Robots* (2018).
13. R. H. Andersen, L. Dalgaard, A. B. Beck and J. Hallam, An architecture for efficient reuse in flexible production scenarios, in *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*, 2015.
14. M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng, Ros: an open-source robot operating system, in *ICRA workshop on open source software*, (3.2)2009.
15. C. Schou and O. Madsen, Towards shop floor hardware reconfiguration for industrial collaborative robots, in *In proceedings of the 19th International Conference on Climbing and Walking Robots and Support Technologies for Mobile Machines (CLAWAR 2016), London, United Kingdom*, 2016.
16. O. Madsen, S. Bøgh, C. Schou, R. S. Andersen, J. S. Damgaard, M. R. Pedersen and V. Krüger, *Industrial Robot: An International Journal* **42**, 11 (2015).