Human-Robot Interface for Instructing Industrial Tasks using Kinesthetic Teaching

C. Schou[1], J.S. Damgaard[1], S. Bøgh[1], O. Madsen[1]
[1]Department of Mechanical and Manufacturing Engineering, Aalborg University, Denmark, cs@m-tech.aau.dk
(Tel: +45-20868884)

***Abstract*— Today, the manufacturing industries increasingly demand more flexible and agile production systems. This demand is also reflected onto the field of robotics, as the majority of robots in the industry today are bolted to the ground and dedicated to a specific task. An Autonomous Industrial Mobile Manipulator (AIMM) offers a higher level of hardware flexibility, but in order to benefit from this flexibility the demand for new approaches to operating and programming new tasks is inevitable. Research within this topic has proposed a task-level-programming, where robot programming is generalized into a selection of skills.**

This paper presents a human-robot interface based on task-level-programming and kinesthetic teaching, which was assessed by nine people of varying robotics experience. In evaluation of the tests several improvements to the HRI are proposed, while the underlying concept is found to simplify programming of industrial task and thus making this available to the production floor operator.

***Index Terms*— Human robot interface, industrial robotics, mobile manipulation, kinesthetic teaching.**

I. INTRODUCTION

During the past decade the production paradigm has started changing from mass production and Dedicated Manufacturing Lines towards mass customization and more flexible and agile production systems. This change is necessary in order to cope with the globalization of markets, the trade instability, and the explosion of product variety, which are stressing the time to market and the need for increased adjustment and responsiveness [1]. From this shift a demand for more flexible production equipment emerges, not least in the field of robotics.

A solution to the inflexibility of traditional robots is to mount the robot to a moveable platform and thus creating an Autonomous Industrial Mobile Manipulator (AIMM). An AIMM presents a highly flexible and automated production resource, but this technology has not made it to the industry yet as it still faces several issues. One of these issues is how the AIMM is programmed to a new task. Given the flexibility and versatility of tasks suited for an AIMM, traditional online programming is found too time consuming and offline programming requires extensive modelling of the environment, which even in the industry cannot be expected to be available for all tasks suited for an AIMM. Furthermore, given the flexible concept of AIMM the programming must be quick and available to shop floor personnel [2], [3], [4].

To meet these demands online robot programming must be brought from a robot-specific programming language to a higher level of abstraction. A paradigm that have proven feasible in research is the task-level-programming, as it is focused on object related goals and not the robot motions to achieve it.

Prior research at Aalborg University has focused on the identification, formulation and integration of skills aimed at tasks in industrial manufacturing environments [5], [6], [7]. This paper presents a Human-Robot Interface (HRI) for programming and executing tasks on AIMMs, without requiring extensive robotics training. This HRI is based on the concept of task-level-programming and has been implemented on *Little Helper*, an AIMM at Aalborg University [3]. The interface is based on a graphical Man-Machine-Interface (MMI) intended for a portable tablet and physical human-robot interaction in form of kinesthetic teaching. The purpose of this interface

In Section III t
task-level-programming is presented.
described in Section IV and Section V
assessment of the HRI conducted at AAU.
presented in Section VI and Section
concluding remarks and future work.

A. Related Research

In research, many have attended the iss
online robot programming faster, more in
more autonomous using a higher-level abs
the basic motions of the hardware cor
introducing this higher-level abstra
programming shifts from sequencing mo

hardware component to sequencing *actions* or *skills* of the robot as a whole.

In the literature, one method for more intuitive online robot programming is *programming by demonstration*, where the robot "learns" from the human operator. One approach is for the robot to observe the human either offline or online [8], [9], [10], [11]. Another approach seen in research is the *kinesthetic teaching* in which the human user pilots the robot manipulator through the intended trajectory [12], [13], [14], [15]. Kormushev et al. (2011) [13] presented a kinesthetic teaching approach where both position but also force profiles could be taught to the robot. Akgun et al. (2012) [14] used kinesthetic teaching on the humanoid *Simon*. In their work the user could both teach trajectories and *keyframes*, hence single positions. They argue that for some tasks the operator might not want to explicitly define the trajectory, but simply the target position. S. Wrede et al. (2013) [15] presents a framework for kinesthetic teaching of redundant manipulators in constraints workspaces by first teaching the allowable kinematic configuration and afterwards teaching the task-related trajectory; while obeying the previous kinematic configuration. Their approach is also levelled for and tested on laypersons.

The contribution of this paper is a method for instructing industrial tasks on an AIMM intuitively without the need for robotics expertise. This method is demonstrated with a HRI, which not only implements kinesthetic teaching, but also includes a surrounding graphical framework for intuitive instantiating and sequencing skills. An assessment of the HRI demonstrates the feasibility of this approach to robot programming and the availability to non-robotics experts.

II. Little Helper

In 2007 the first AIMM at Aalborg University was designed and assembled. The fundamental vision was to create a "little helper" to assist the shop floor operators in industrial manufacturing by attending some of the simple repetitive tasks in the production. From this vision the mobile robot and the project itself got the name "Little Helper" [3]. The latest version, Little Helper 3 from 2012, is shown in Fig. 1.

Fig. 1. Little Helper 3, an AIMM at Aalborg University designed for attending complex assembly tasks.

Little Helper 3 is primarily assembled from commercial-of-the-self (COTS) components, but a custom aluminium frame structure to interface the different components had to be designed. The key components are:

- Neobotix MP-655 non-holonomic platform
- KUKA Light Weight Robot 4
- Schunk WSG-50 electric parallel gripper
- Two onboard PCs, one is an accessible laptop
- Wireless emergency stop

The software and thus control of Little Helper uses a distributed architecture where the low-level and real-time control of each device is carried out on the device itself. The Robot Operating System (ROS) [16] is used to link the distributed nodes. A central node utilizes all the distributed device nodes and thus creates the higher level of control. This node also incorporates the user interface including task-level-programming, which is described in the next section.

III. Task-Level-Programming

If the AIMM is to be programmed by an operator with limited robotics knowledge at the shop floor during production runtime, the programming interface must be easier and faster to use, than conventional robot programming interfaces. In order to do this it is chosen to bring the robot programming from the level of simple device specific commands to a task focused level.
The approach of task-level-programming is divided into three layers consisting of device primitives, skills, and tasks; inspired by [17]. Fig. 2 shows the setting and the interaction between the individual layers. Tasks, skills, and device primitives are described in the following sections.

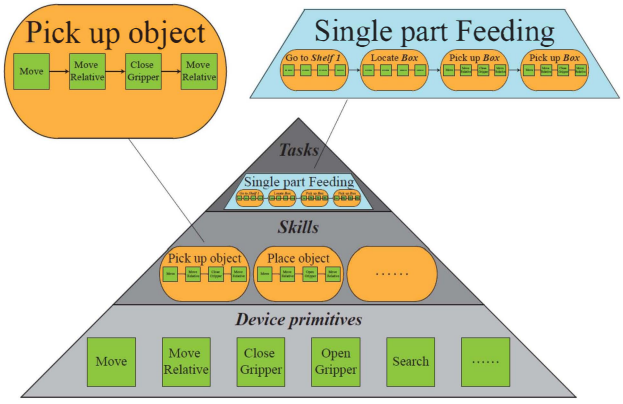

Fig. 2. This figure shows the three layers of abstraction used on Little Helper. The lowest level is the device primitive layer, which is basic functionalities and motions of the different devices; hence this layer is the closest to real-time. The middle layer is the skill layer, which are object-centered capabilities of the AIMM as a whole. Sequencing and parameterizing skills creates a task and thus the layer above skills. This architecture is inspired by [17].

A. Device Primitives

Device primitives are described as basic motions and functionalities of the different devices e.g. *Grasp*, *Move to XYZ* or *Search in x-direction*. The level below device primitives is described as the device specific driver.

B Skills

One of the key elements in bringing robot instruction from a robot-specific programming language to a higher abstraction level is skills. On one hand the skills represent the foundation of the task, and thus the building blocks used by the operator. On the other hand skills are defined as a higher-level abstraction of functionalities and motions of the individual devices of the robot; that is a higher level abstraction of device primitives. A skill utilizes these device primitives as motions and combining them with sensor input, advanced mathematics and advanced robotics the skill manipulates an object in order to achieve a production-related goal. That is, a skill is an object-oriented capability of the AIMM as a whole, i.e. "pick", "place", "rotate" etc.

It is in the skill layer the robotic experts can bring advanced robotics software into the system, yet keeping an object-oriented, and thus task-oriented, level of abstraction for the operator, compared to a level of simple device specific commands.

Based on the skill definition a general skill model is established in Fig. 3.

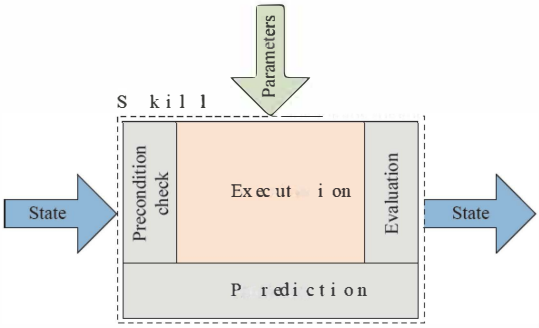

Fig. 3. Skill model. A skill is composed of an execution routine, preceded by a precondition check to verify that the initial state is feasible to this skill. Subsequent to the execution routine a postcondition check verifies the outcome of the skill and compares it to a predicted outcome. [5]

A skill relies on a defined motion sequence, but the motions themselves are adapted to a specific task by a set of parameters, thus making the skills generic within a certain scope. The parameters are established during the instructing of the robot and stored in a task file along with the skill sequence.

During execution of the skill these parameters are given to the system together with a set of state variables. A precondition check is conducted by examination of the given input parameters compared to the measured state. If the precondition check is performed successfully the execution part is conducted. In other words, the precondition check serves as a safety net. The postcondition check is conducted based on a prediction of the desired goal and an evaluation of the outcome. If the comparison between the evaluation and the predicted outcome is consistent within a given range, the postcondition check is successful. As a result of the accomplished skill the system state variables are changed and updated.

Currently, 13 different skills are implemented into a library and these have proven sufficient for programming a variety of different industrial tasks, both machine tending, assembly and logistic tasks.

C Tasks

A task is described as a sequence of skills and contains an overall goal e.g. *pick up the rotor at station A*. In this way a task is established on the basis of a library of skills. A task is defined as a sequence of skills each parameterized to the specific task and thus the task itself is merely a file containing the sequence of skills and the parameters for each skill. The task has a set of measurable state variables that are changed continuously during the skill sequence.

IV. Human-Robot Interface

A HRI structured around the concept of task-level-programming has been developed and implemented on Little Helper. This HRI provides functionality for both executing and programming new tasks. This is met on the skill level by implementing both a programming routine and an execution routine in each skill. The correlation between task programming and execution is illustrated in Fig. 4.

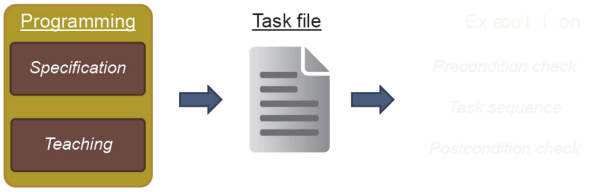

Fig. 4. Illustration of the correlation between the programming part and the execution part of a task. During the programming part the sequence of skills is established and a set of parameters is obtained through a programming routine. This information is stored in a task file, from which the task can later be executed. The programming part is divided into a specification phase and a teaching phase.

A. Programming a Task

As shown on Fig. 4 the programming part is divided into a specification phase and a teaching phase. During the specification phase the sequence of skills is chosen and partly parameterized, and during the subsequent teaching phase the locational parameters are obtained through kinesthetic teaching.

1) Specification Phase

The specification phase is conducted in a graphical application on a PC or on a tablet. During the specification phase the skill sequence is chosen from the library of skills and several parameters are obtained through user input in the MMI. To simplify the interface several parameters are preselected to standardized values.

Fig. 5 shows two screenshots from the graphical MMI; a screenshot of an established sequence and a screenshot of the parameters selected for a skill.

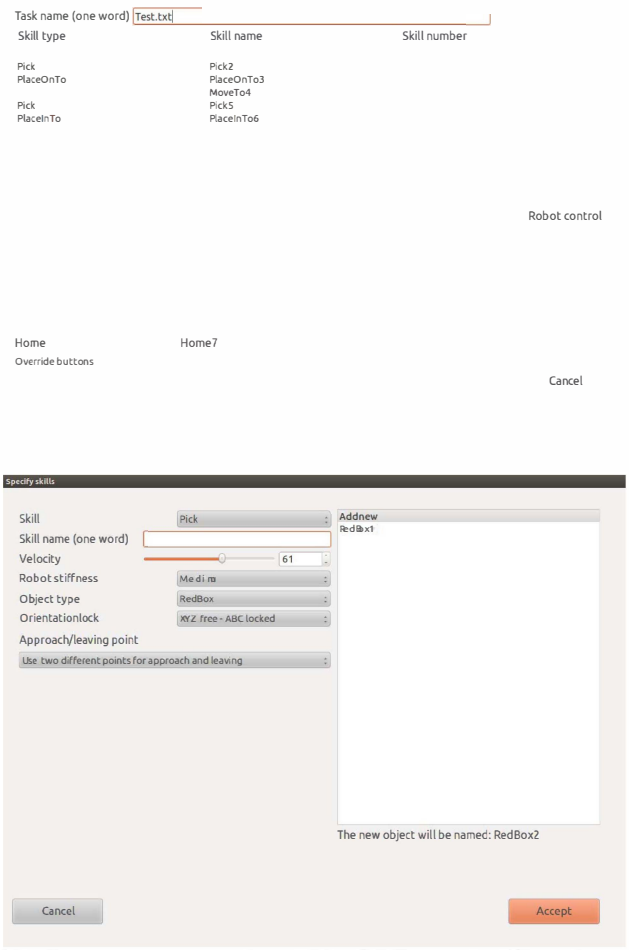

Fig. 5. Two screenshots of the graphical MMI. The upper figure shows the window for creating a new task where a skill sequence has been established. The lower figure shows the window for selecting a skill along with parameters for it.

During the specification several parameters are given for each skill, where some of these correlate to the execution of the skill, hence correlates to the task, and other parameters configure how the following teaching phase is conducted.

2) Teaching Phase

Once the skill sequence has been configured the teaching phase is commenced. This phase relies on kinesthetic teaching during which the operator directly interacts with the manipulator and pilots it to the specified target. During the teaching phase each skill in the skill sequence is taught sequentially. In this way the teaching sequence correspond directly to the execution sequence and thus creates a clear overview of the progress and outcome. The teaching of the skills follows the teaching routines for each skill, during which the operator is guided through written instructions in the graphical MMI and sound outputs.

The impedance control mode of the KUKA LWR [18] makes the manipulator actively compliant, which facilitates the kinesthetic teaching. In this mode the manipulator is set to only compensate for its own weight (including tool) by using integrated torque sensors in each joint for the active compliance. Furthermore these force-sensors facilitate the measuring force applied to the end-effector, which is used as a convenient method of obtaining user input. Thus during the teaching phase the user does not need to intervene with the graphical application besides reading the instructions.

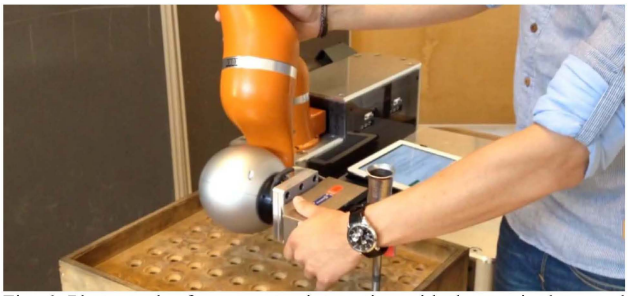

Fig. 6. Photograph of an operator interacting with the manipulator and piloting it to a coordinate.

Teaching a 3D coordinate is central in many skills. This starts by the operator applying force to the end-effector to start the teaching process. This puts the manipulator into gravity compensation mode and the operator "drags" the tool to the location, see Fig. 6. The location is stored when the tool centre point (TCP) is held steady for three consecutive seconds.

B Executing a Task

When implementing AIMMs on a larger scale in an industrial production environment determining which tasks it should carry out during the day should be handled by a centralized mission planner with access to the Enterprise Resource Planning (ERP) system. In this way the AIMM is automatically assigned to the tasks, where the capacity needs are the highest and thus the full benefit of the flexibility of the AIMM is achieved.

In several scenarios the operator might need to execute a task or assign a small mission to the AIMM, i.e. to test a newly programmed task or if the AIMM is used to assist the operator for shorter periods of time.

The graphical MMI incorporates both functionalities. It has an automatic mode, where it responds to missions from a mission planner, and it has a simple execution mode, where the operator selects one or several tasks to execute. In both cases the system will open the selected task file, interpret it and execute the given skills with the parameters from the task file, while following the skill model in Fig. 3.

V. HRI Assessment

To assess the presented HRI a series of user tests have been conducted at Aalborg University. The main purpose was to obtain feedback from various users and to assess how well they comprehended and operated the system.

The test consisted of two separate tasks; a simple pick and place task and a more advanced peg-in-hole task. The users performed the test individually and were each given a 15 minutes introduction, including the hardware devices of Little Helper, the skill library, the kinesthetic teaching, the graphical MMI and the tasks. In the programming

phase the users worked independently, but an expert stood by to provide help when requested by the operator.

A. Task 1 - Simple

The first task was a simple pick and place, where an object was to be picked from a fixture and afterwards placed on a surface within a given area. From an industrial point of view this task could substitute a single part feeding task, where the robot is to unload single components from its platform to a table or a machine. The object and fixture used was from the Cranfield benchmark and the task was conducted on the platform of Little Helper, see Fig. 7.

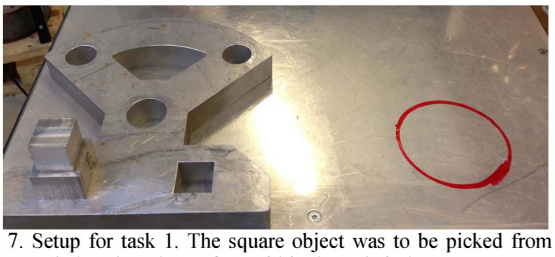

Fig. 7. Setup for task 1. The square object was to be picked from the fixture and placed on the surface within the red circle.

B Task 2 - Advanced

The second task was a more advanced pick and place task, as it required more object manipulation, and more foresight and accuracy in the teaching phase. The task was to pick one of the similar components from the left fixture, see Fig. 8, turn it up-side down, and place it into the fixture on the right, as shown on Fig. 8.

Fig. 8. Setup for task 2. One of the objects from the left fixture was to be placed up-side down into the fixture on the right.

From an industrial point of view this task could simulate a step in an assembly task. Components and fixtures from a real industrial setup were used. The task was conducted on a small trolley located next to Little Helper.

C. Users

A total of nine people have participated in the test and conducted the two tasks. The test persons were chosen from three different experience levels and thus sorted into three groups. The purpose of having people of various experiences test the HRI was to assess the implications of robotics knowledge on the users' performances.

Group 1 is people with no robotics experience and no experience with the presented programming interface. This group consists of employees from the machine shop at Aalborg University and a PhD student.

Group 2 is people with robotics expertise, but without any prior hand-on experience with the presented interface.

Group 3 is people with both robotics expertise and experience with the presented interface.

D. Measures

For each user the two tasks were treated individually. During the tests the user worked independently, but an expert was standing by to provide help when requested. The time spent on the specification phase and the teaching phase was recorded along with the selected sequence of skills. Additionally, the number of errors made by the user and the number of times help was requested from the expert was logged. This included a short description of the error made or the help provided. Some user created errors let to failure of the teaching phase in the sense, that it could not be completed. In this case the error was recorded and the user started over.

After programming either task the obtained task-file was executed to validate the outcome.

VI. Results

Table I presents the mean programming times for each user group from both tasks. The times are in seconds and the upper number in each square is the combined specification and teaching time, hence the total programming time. The two numbers in the parenthesis is the specification and teaching time respectively.

TABLE I
MEAN PROGRAMMING TIME FOR EACH USER GROUP

Time [s]	Group 1	Group 2	Group 3
Simple	320 (193/126)	280 (150/130)	157 (87/70)
Advanced	749 (288/461)	581 (185/396)	280 (83/196)

Table II shows the mean for each group of the recorded number of errors and help requests by the user. The errors and help requests have been combined, as without the help provided the user would most likely have induced an error.

TABLE II
MEAN ERRORS/HELP REQUESTS FOR EACH USER GROUP

	Group 1	Group 2	Group 3
Simple	2.3	0.3	0.0
Advanced	4.7	1.3	0 .3

Of all the errors and help requests recorded, approximately 47% related to foreseeing the progress of the task, hence predicting each step in the task. This was relevant in both the specification and the teaching phase. Approximately 53% of the errors and help requests were related to the written instructions provided to the user during the teaching phase.

VII. Conclusion and Future Work

In this paper a HRI for programming industrial tasks has been presented. The foundation for the interface is a three-layered architecture with skills as the building

blocks for the operator to compose a task from. The programming is divided into two phases. In the specification phase, the skill sequence and initial parameters are selected through an intuitive graphical application on either a PC or a tablet. In the teaching phase kinesthetic teaching is used for obtaining locational bound parameters for each skill. The HRI was tested by nine people with varying robotics experience on two industrial relevant tasks of different complexity.

With the presented system and assessment it is shown, that an intuitive method for robot programming making the technology available to non-robotics experts is feasible using an abstraction in terms of skills, a graphical interface and kinesthetic teaching. Compared to related research this does not assess or expand the kinesthetic teaching itself, but introduces the necessary surrounding interface, including the preceding instantiation phase referred to as specification, for an inexperienced user to operate it. The presented HRI is considered both applicable for programming AIMMs and traditional robots, though the need for a robot with a compliant control mode, or creation of such using an external force sensor, is inevitable.

The tests revealed that the interface still requires better instructions to support the operator during the teaching phase and basic issues from the robotic domain still needs to be addressed, such as avoiding joint limits and choosing collision free paths. However, as a concept the presented interface shows promising as a platform for making technology available to the shop floor operators. As a result they can transfer their production insight and knowledge to the robot-sequence.

In future work the instructions given during the teaching phase will be replaced by short animations or voice based instructions. A motion planner will be integrated to automatically generate collision free paths and avoid joint limits, which will significantly aid the operator in the teaching phase. To address the issue of foreseeing the progress of the entire task, the specification and teaching phases might be carried out consecutively for each skill instead of consecutively for the entire task.

The assessment presented in this paper will be proceeded in fall 2013 by a test in a running production environment with industrial components and fixtures, using test persons from the production floor.

ACKNOWLEDGMENT

This work has been supported by the FP7 project 260026 TAPAS - Robotics-enabled logistics and assistive services for the transformable factory of the future.

The authors would like to thank all participants.

REFERENCES

[1] F. Jovane, Y. Koren, and C. Boër, "Present and future of flexible automation: Towards new paradigms," *Cirp Annals-manufacturing Technology*, vol. 52, no. 2, pp. 543–560, 2003.

[2] European Strategic Robotics Platform (EUROP), "The strategic research agenda for robotics in Europe - robotics vision for 2020 and beyond," 2009.

[3] M. Hvilshøj and S. Bøgh, "Little Helper - An autonomous industrial mobile manipulator concept," *International Journal of Advanced Robotic Systems*, vol. 8, no. 2, 2011.

[4] "Robotics-enabled logistics and assistive services for the transformable factory of the future (TAPAS)." Project under the European Community's Seventh Framework Programme. Webpage: http://www.tapas-project.eu/.

[5] S. Bøgh, O. Nielsen, M. Pedersen, V. Krüger, and O. Madsen, "Does your robot have skills?," in *The 43nd ISR (International Symposium of Robotics)*, 2012.

[6] S. Bøgh, M. Hvilshøj, M. Kristiansen, and O. Madsen, "Identifying and evaluating suitable tasks for autonomous industrial mobile manipulators (AIMM)," *The International Journal of Advanced Manufacturing Technology*, vol. 61, no. 5-8, pp. 713–726, 2012.

[7] M. Hvilshøj, S. Bøgh, O. Nielsen, and O. Madsen, "Multiple part feeding – real-world application for mobile manipulators," *Assembly Automation*, vol. 32, no. 1, pp. 62–71, 2012.

[8] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," *Springer Handbook of Robotics*, chapter 59, Springer, 2008.

[9] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zollner and M. Bordegoni, "Learning robot behaviour and skills based on human demonstration and advice: The machine learning paradigm," in *Proceedings of the International Symposium on Robotics Research*, pp. 229–238, 1999.

[10] J. Aleotti and S. Caselli, "Interactive teaching of task-oriented robot grasps," in *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 539-550, 2010.

[11] D. Kulic, C. Ott, D. Lee, J. Ishikawa and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," in *International Journal of Robotics Research*, vol. 31, no. 3, pp. 330–345, 2011.

[12] H.B. Amor, E. Berger, D. Vogt, and B. Jun, "Kinesthetic bootstrapping: Teaching motor skills to humanoid robots through physical interaction," in *Proceedings of the 32nd annual German conference on Advances in artificial intelligence (KI'09)*, pp. 492–499, 2009.

[13] P. Kormushev, S. Calinon, and D.G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input", in *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[14] B. Akgun, M. Cakmak, J.W. Yoo, and A.L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: a human-robot interaction perspective," In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction (HRI '12)*, pp. 391-398, 2012.

[15] S. Wrede, C. Emmerich, R. Grünberg, A. Nordmann, A. Swadzba, and J.J. Steil, "A User Study on Kinesthetic Teaching of Redundant Robots in Task and Configuration Space," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 56 – 81, 2013.

[16] ROS Wiki, URL: http://www.ros.org.

[17] E. Gat, "On three-layer architectures," in Artificial Intelligence and Mobile Robots, MIT Press, 1998.

[18] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer and A. Beyer, "The KUKA-DLR Lightweight Robot arm – a new reference platform for robotics research and manufacturing," in *Proceeding of Joint 41st International Symposium on Robotics and 6th German Conference on Robotics*, pp. 741–748, 2010.