

FlexRoP - Flexible, Assistive Robots for Customized Production

Markus Ikeda^{1*}, Srinivas Maddukuri¹, Michael Hofmann¹, Andreas Pichler¹, Xiang Zhang²,
Athanasios Polydoros², Justus Piater², Klemens Winkler³, Klaus Brenner³,
Ioan Harton⁴ and Uwe Neugebauer⁴

Abstract—Flexible production assistants of the future are required to be skillful, universally applicable, safe and easy to program. State of the art robot systems that are intended to be used for human robot collaboration require in some cases unintuitive text based programming, and remain, especially in combination with peripheral hardware like external sensors or machine vision algorithms, complicated. The FlexRoP project tries to overcome current limitations by development and usage of a flexible skill-based robot programming middleware and improved user interface technologies. This paper introduces usecases, the intended system architecture, methodology for description and training of kinesthetic skills as well as first application results and intentions for future developments.

I. INTRODUCTION

Medium to small batch size production often can't be automated with robots which require costly space and need infrastructure (e.g. fences and fixtures for part allocation). Uncertainty handling (e.g. objects that are not allocated in a defined way or underlie a tolerance in type, shape or color) is far from trivial. Additional sensors and algorithms increase system complexity and require special engineering knowledge. Flexibility for industry means universal applicability and deployment to unmodified human workplaces as far as tools or processes are concerned without complex recertification procedures or questioning legal security. Ramp up of new and recommissioning of former applications is required to be done fast and by non experts.

The FlexRoP project will carry out research to make robots easier to program and thus more flexible. Project goals comprise the definition of a universal skill representation for assembly tasks, implementation of automatic and semiautomatic skill acquisition techniques based on observation learning and kinesthetic teaching, generalization techniques and implementation of skill based action synthesis algorithms.

This paper presents:

- Two selected real world usecases in the FlexRoP project.
- The system architecture for the flexible robotic assembly assistant providing workflow based programming.
- A methodology to describe and acquire kinesthetic skills from kinesthetic demonstration.

¹ Profactor GmbH, Im Stadgut A2, A-4407 Steyr-Gleink, Austria

² Universität Innsbruck, Innrain 52, A-6020 Innsbruck, Austria

³ Schmachtl GmbH, Pummererstrasse 36, A-4020 Linz, Austria

⁴ Magna Steyr Fahrzeugtechnik AG & Co KG, Liebenauer Hauptstrasse 317, A-8041 Graz, Austria

*Corresponding author: Markus Ikeda;
Markus.Ikeda@profactor.at

- Evaluation results from workflow based programming with kinesthetic parameterization and kinesthetic skill acquisition.
- Inferred intentions for future developments.

II. USECASES

Two real world production usecases from automotive pre-assembly are considered. The usecases require screwing, clip in and manipulation operations in a very broad range of applications. In so called brownfield [1] environments available (hand)tools have to be picked up by the robot rather than spanning specialized robot tools to guarantee deployability to any human workplace.

Usecase A targets the pre-assembly of a centerspeaker assembly. A speaker has to be fixed with three screws to a plastic carrier while a tweeter needs to be clipped in (see Fig. 1). Handling of the non-rigid wires is omitted. Process forces are low but the required pose precision for screwing and clipping is very high (< 1mm). The complexity of the entire process (which consists of 7 subprocesses - see Table I) is extremely high. Three different objects are presented in boxes and have to be manipulated as well as the intermediate assemblies and the power screwdriver which has to pick up, hold and manipulate the screw axially perfectly aligned during transport and process. In order to be able to guarantee product and process quality methodology for quality assessment is required. This might be natural and easy for a human but independent of the available data (acoustic, FT-signal, optical) extremely challenging for any technical system.

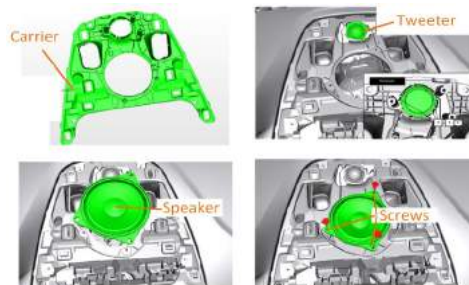


Fig. 1. Usecase A - Center-speaker assembly

Usecase B considers the joint pre-assembly of an automotive swivel-bearing assembly by human and robot. Han-

dling of components and assembly takes place close to the robot's load limits with high handling and process forces. A human carries out processes unsuitable for the robot like screw feeding and delicate ambidextrous assembly operations (e.g. mounting of brackets and brake hose - 1 in Fig. 2). Unergonomic handling of heavy objects is carried out by the robot as well as the error prone screw tightening operation for the assembly of wheel bearing to swivel bearing which needs to be carried out in a specific order (2 in Fig. 2).

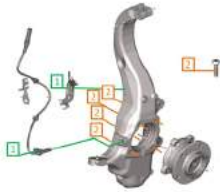


Fig. 2. Usecase B - Swivel bearing assembly

III. SYSTEM ARCHITECTURE

A. Hardware

The robot assistant (Fig. 3) consists of a passively mobile platform with retractable wheels, an electric enclosure containing robot and system controller as well as additional IO and power supply components. The platform is equipped with a KUKA LBR iiwa 14 R820 robot. The User Interface (UI) consists of a touch screen monitor on the mobile platform and the robot's touch pneumatic media flange. The robot is equipped with one universal tool for both applications. The diversity of requirements with regard to object shapes, payloads and processes result in a complex tool design (see Fig. 4) with following components:

- Force Torque (FT) sensor for measuring process wrench.
- A chassis for installation of various components.
- RGBD and 2D cameras for automatic position accuracy compensation functionality.
- Two electric grippers in order to be able to manipulate multiple objects or long objects.
- Automatic toolchanger for spanning additional process tools (ordinary hand tools articulated by pneumatic actuators).

Handling and manipulation of objects was intended with universal grippers and force closure. Tests disproved the applicability of several universal grippers for accuracy and process stability reasons so aluminium fingers with form adjusted plastic inlays are used.

B. Software

The robot assistant is required to be programmable without special training. A KUKA iiwa [2] may as a HRC-capable device offer handguidance for parameterization but needs to be programmed text based (in JAVA) as well as

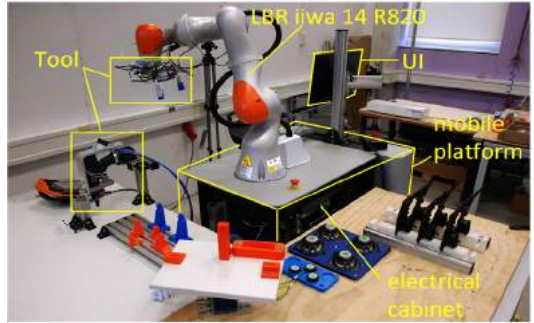


Fig. 3. System Overview

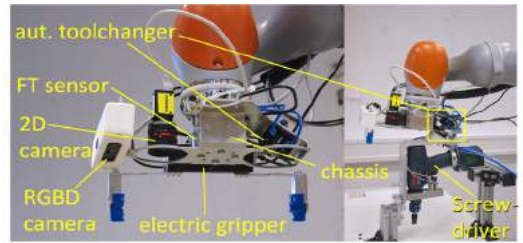


Fig. 4. Flexible tool prototype (2nd gripper not installed)

machine vision algorithms or standard PLC code. Therefore XRob™ [3] (see Fig. 6) is introduced as an abstraction layer for all hardware (cameras, sensors, robots, etc.) and software components (object pose recognition, path planning, etc.). For kinesthetic skill learning a real time interface to the robot and the FT sensor is required. Therefore ROS and the KUKA fast research interface are used. Fig. 5 describes the selected modular system architecture.

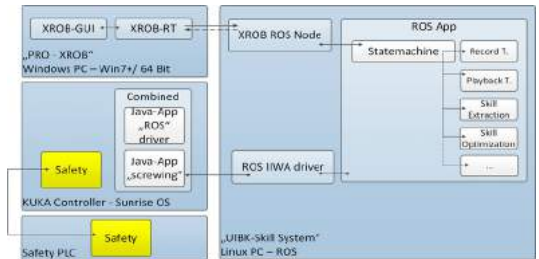


Fig. 5. Software architecture

IV. SKILL BASED WORKFLOW PROGRAMMING

Robot programming in industrial applications is done mainly in proprietary text based programming languages. Skills are treated as traditional, "unintelligent" robot motion programs (macros) that are augmented with pre and

post-conditions to add situational knowledge. Macros are supposed to work on objects in the workspace that are recognized via some kind of sensing device (e.g. optical). For example [4] presents a unifying terminology for task-level programming of highly flexible mobile manipulators in industrial environments, while [5] demonstrates the skills which are needed for industrial kitting applications.

A. Skill Based Programming Framework

Task-level programming is based on lower level entities, usually called robot skills. The description of processes can be done at different levels of granularity. Tasks can be broken into more or less complex subtasks ranging from sensory and/or motor base skills to complex aggregate subtasks. A skill is a primitive that allows the coordination, control and supervision of a specific task. The primitives can incorporate advanced task specifications, necessary control, and sensing capabilities, which allows a skill to handle uncertainties during execution. In contrast to the concept of skills, skill primitives [6], [7], [8] are rather well defined in the robotics community. This layered approach is reflected by the design of the XRob™ software framework (see Fig. 6) which can aggregate basic functionality (e.g. data acquisition, image processing, robot movements and macros, etc.) to more complex aggregate subtasks that can easily be reused. After graphical configuration of a workflow process points are parameterized by bringing the tool center point to its destination and adopting relevant data (e.g. the current position or the current camera image) electronically. That allows programming processes and movements between quasistatic intermediate process points. If more complex trajectories are required the system incorporates dynamic motion primitive based skills.

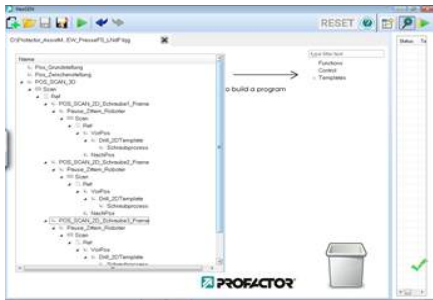


Fig. 6. XROB Graphical User Interface

B. Dynamic Motion Primitive Based Skills

In [9] and [10] is given an overview on programming by demonstration. Dynamic Motion Primitives (DMPs) have been a very popular method for learning and generalization of kinesthetically taught motions [10] with multiple extensions [11], [12], [13]. They are motivated by the need to derive a motion representation which is capable, not only to

reproduce complex trajectories but also to easily generalize them. DMPs are a combination of two terms. A simple linear dynamical system $l(\cdot)$, which is well defined and has stable behavior and a nonlinear forcing term $f(\cdot)$ which makes the reproduction and generalization of complex motions feasible

$$\dot{y} = l(g, y, \dot{y}) + f(x, g). \quad (1)$$

In the case of discrete motions the linear system is a stable attractor, usually a PID controller

$$l(g, y, \dot{y}) = \alpha_y(\beta_y(g - y) - \dot{y}), \quad (2)$$

where y is the joints' position of the robot, g is the target states, and α and β are gain terms of the PID controller which draw the manipulator to the target state. Adding a forcing term to the linear system allow to modify the trajectory:

$$\ddot{y} = \alpha_y(\beta_y(g - y) - \dot{y}) + f. \quad (3)$$

The challenge in DMPs is to appropriately define the non-linear forcing term f over time while ensuring stability of the system and generalization. This is achieved by introducing a canonical dynamical system denoted as x with simplistic dynamics:

$$\dot{x} = -\alpha_x x. \quad (4)$$

Thus the forcing term f depends on the value of the canonical system as follows:

$$f(x, g) = \frac{\sum_{i=1}^N \psi_i w_i}{\sum_{i=1}^N \psi_i} x(g - y_0). \quad (5)$$

y_0 is the starting state of the system, and $\psi_i = \exp(-h_i(x - c_i)^2)$ is a Gaussian kernel centered at c_i .

Training of DMPs is achieved by optimizing its hyper-parameters (w) with a given trajectory. While the desired motion is demonstrated, the sensors' values are recorded and they are used to derive the hyper-parameters based on Eqn. (3) which is written as:

$$\ddot{y} - \alpha_y(\beta_y(g - y) - \dot{y}) = f(x). \quad (6)$$

Thus, the forcing term is optimized to compensate the error of the linear dynamical system – which are the training targets of the learning rule – at each state of the canonical system x which is the training input. This corresponds to a regression problem which can be solved with a variety of methods such as Locally Weighted Regression [14] or Locally Weighted Projection Regression [15].

C. Motion Assessment Primitive

The motion assessment primitive is responsible for providing an evaluation of the performed motion, thus it evaluates the DMP's performance. This is achieved by a two-tier process which exploits the trajectory recorded through kinesthetic teaching. Those recordings include both the joints' states and the exerted force/torques on the end-effector. This makes the derivation of the motion's contact dynamics

model through machine learning techniques feasible which maps joint states to exerted forces/torques. Thus the system "learns" which forces and torques to expect at specific joint states. Therefore, a ground-truth model is created from the end-user demonstration and a comparison model is created from the recording of the autonomous DMP's movement reproduction. The difference of those two models is measured and fed to the second tier which classifies the motion as successful or not.

Gaussian processes (GPs) are employed for learning the wrench model of the executed task. GPs are a powerful non-parametric machine learning approach. Contrary to other methods that infer a set of function parameters, GP infers the function f directly and therefore can be anticipated as probability distribution over functions. A GP is defined by a mean $m(\mathbf{x})$ and a kernel (covariance function) $\mathbf{K}(\mathbf{x}, \mathbf{x})$ as illustrated in Eqn. (7). Typical choices are a squared exponential kernel and a zero mean

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), \mathbf{K}(\mathbf{x}, \mathbf{x})). \quad (7)$$

GP, employ the Bayes rule for the derivation of the posterior distribution over functions -see Eqn. (8), where \mathbf{t} is the vector of target values, the force/torques in this case. In regression problems the latent function f is continuous and therefore an appropriate likelihood is the normal distribution $\mathcal{N}(f|\mathbf{m}(\mathbf{x}), \mathbf{K}(\mathbf{x}, \mathbf{x}))$ and the GP prior is also a Gaussian process $p(f|\mathbf{X}) \sim \mathcal{GP}(0, \mathbf{K}(\mathbf{x}, \mathbf{x}))$.

The posterior distribution which represents the learned wrench model given the recorded data .

$$p(f|\mathbf{X}, \mathbf{t}) = \frac{\mathcal{N}(f|\mathbf{m}(\mathbf{x}), \mathbf{K}(\mathbf{x}, \mathbf{x}))p(f|\mathbf{X})}{p(\mathbf{t}|\mathbf{x})}. \quad (8)$$

The term of interest in the case of motion assessment primitive is the marginal likelihood $p(\mathbf{t}|\mathbf{x})$ because the optimal parameters of the kernel are derived by optimizing it. Thus, the contact dynamics model derives by minimizing the logarithm in Eqn. (9) which can be achieved by using any gradient-based optimization method such as gradient descent

$$\log(\mathbf{t}|\mathbf{X}) = -\frac{1}{2}\mathbf{t}^T \mathbf{K}^{-1}\mathbf{t} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log 2\pi. \quad (9)$$

The assessment primitive creates six ground-truth models, one for each wrench degree of freedom and other six models from the autonomous execution of the DMP. Those probabilistic models are then compared using Hellinger distance - Eqn. (10) - which yields a similarity measurement h for each k wrench component. Those measurements are fed to the second tier of the primitive, a Naive Bayes classifier which classifies the similarity measures as success or failure

$$h_k(\mathcal{GP}_k^{\text{demo}}, \mathcal{GP}_k^{\text{exe}}) = \sqrt{1 - \frac{\sqrt{|\mathbf{K}_k^{\text{demo}}| |\mathbf{K}_k^{\text{exe}}|}}{\sqrt{\frac{1}{2}|\mathbf{K}_k^{\text{demo}} + \mathbf{K}_k^{\text{exe}}|}}}. \quad (10)$$

In the second stage of the primitive the set of similarity measurements \mathbf{h} are fed to a Naive Bayes classifier which

applies the Bayes rule - Eqn. (11) - for the derivation of $p(C_j|\mathbf{s}^*)$, where $p(C_j) = N_j/N$ is the prior probability of the class j , $p(\mathbf{h}^*|C_j)$ is the likelihood that the sample \mathbf{h}^* belongs to the class j and $p(\mathbf{h}^*)$ is a scaling term independent from the class and therefore can be omitted

$$p(C_j|\mathbf{s}^*) = \frac{p(C_j)p(\mathbf{s}^*|C_j)}{p(\mathbf{s}^*)}. \quad (11)$$

The likelihood derives based on the assumption that the similarity measurements are independent and identically distributed and is calculated as:

$$p(\mathbf{h}^*|C_j) = \prod_{d=1}^D p(s_d^*|C_j), \quad (12)$$

where K is the dimensionality of the similarity measurements. It is assumed that their values are distributed according to a Gaussian distribution $\mathcal{N}(\mu_k^j, \sigma_k^j)$ with mean μ_k^j , the mean value of similarity measurement d which belongs to class j and its corresponding variance σ_k^j . Thus Eqn. (11) can be written as:

$$p(C_j|\mathbf{s}^*) \propto p(C_j) \prod_{d=1}^D \mathcal{N}(s_d^*|\mu_d^j, \sigma_d^j), \quad (13)$$

where the parameters of the Gaussian distribution derive by maximum likelihood estimation.

D. Intermediate Results

1) *Skill Based Workflow Programming*: FlexRoP identified macros for screwing operations as well as the clip-in operation that can be considered as robotic skills themselves and serve as baseline for performance comparison with the kinesthetic skills developed by the project.

The screwing macro considers the basic parameters: start pose, screw length and process force. The clipping macro considers in a similar way start pose, end pose and process force. Together with parameterizeable macros for other operations (robotic movements, etc.) screwing and clipping are accessible through XRobTM.

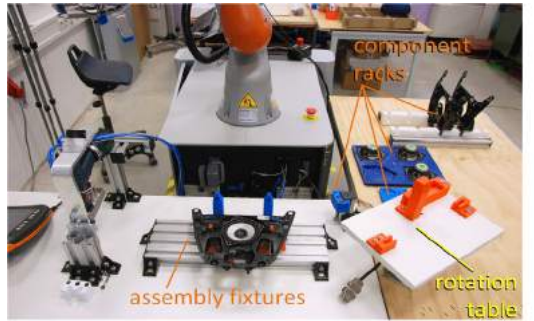


Fig. 7. Overview - Usecase A

Usecase A was split into several suboperations and program templates were created accordingly. Parameterization

of the templates was done by moving the robot to a specific process point and recording relevant data (e.g. cartesian positions, reference images, gripper opening,...). For usability reasons movement of the robot was planned to be done by hand guidance. Tool parameters (inertia, mass) are tuned and the robot flange is intended to be used in zero gravity mode. Total parameterization time of existing workflows sums up to 285min (see Table I), which is high (compared to a target time of 30min which is derived from a productivity calculation) and was caused by high accuracy demands to be able to achieve process stability. Tight clearances of carrier plate and fixtures as well as required positioning accuracy for screws ins screw-holes and components to be assembled require precise teach in which cannot be achieved in gravity compensation mode. For perfect vertical tool alignment and fine positioning of the tool it was, due to not available interfaces required to use the robot teach pendant which required several stop and start operations of the XRob™ driver on the robot controller as well as operation mode changes from automatic to hand mode and vice versa in order to be able to use the robot teach pendants integrated positioning utilities. A detailed analysis of subprocess 1 (see Fig. 8) reveals that operation mode changes as well as interaction with the GUI of the robot (which is required to select correct coordinate frames to travel in for fine positioning or selection of speed) in addition with finerepositioning itself is accountable for almost two thirds of the reparameterization time. Interaction with the XRob™(HTML-)GUI and adjustment of the finger positions in comparison requires less time.

TABLE I
AVERAGE PARAMETERIZATION TIME - 3 TRIALS

subprocess	description	parameterization time
1	Move carrier from rack to assembly fixture	30min
2	Move speaker from rack to assembly fixture (via rotation table)	60min
3	Pick-up of power tool from pod	15min
4	Screw-pick-up & screwing operation (three target positions)	45min
5	Deposit of power tool to pod	15min
6	Reorientation of assembly	60min
7	Move tweeter from rack to clip in position and clipping operation	60min

2) *Dynamic Motion Primitive Based Skills*: In order to evaluate the performance of both the motion and assessment primitives a mock-up which imitates the project’s clip-in process was designed (see Fig. 9). For the evaluation a KUKA iiwa equipped with an ATI force/torque sensor and a simplistic suction cup was used. The primitives were trained on recorded data from one single kinesthetic demonstration and their generalization ability is tested by varying the start pose of the manipulator. The motion primitive managed successfully to execute 17 out of 44 trials resulting in a 39% success rate. An illustration of a successful snap

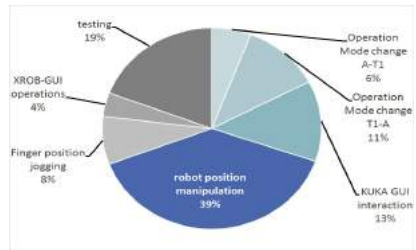


Fig. 8. Subprocess 1 - teach in time breakdown



Fig. 9. The snap-fit used for performing evaluation of the motion and assessment primitives.

is presented in Fig. 10. The motion assessment primitive was evaluated off-line on datasets collected from 31 motions using a cross validation method for training and evaluating the Naive Bayes classifier. In this evaluation method, the data is partitioned in training and testing datasets. The former are used for optimizing the hyper-parameters of the Naive Bayes classifier while the later for evaluating its performance. In detail, a leave-one-out cross validation is performed where the classifier is trained with all the datasets except one which is used for testing. This iterative procedure finishes when all the datasets have been used for testing.

V. CONCLUSIONS & FUTURE WORK

Two immediate directions for further improvements were identified.

A. Skill Based Workflow Programming

Experiments Showed that workflow based programming is still complicated for untrained users. Programming in the worker’s domain without kinesthetic manipulation of the robot itself remains desirable. A novel instrumented power tool as UI for teach in operations is planned. A worker will not have to specify numeric values on a GUI in order to parameterize process workflows. The instrumented power tool will record trajectories in 6DOF as well as time series of process forces and torques as well as the actuation of the tool. Startposition, process forces and screw lengths will



Fig. 10. KUKA iiwa performs a successful snap-fit using DMPs

be derived from the analysis of the data. In comparison to kinesthetic teach in the so called embodiment problem has to be solved since the robot has different reach and multiple kinematic configurations that can be used to position a tool.



Fig. 11. Instrumented tool concept

B. Dynamic Motion Primitive Based Skills

The future work on DMPs will be focused on the issue of the low success rate. A reason for the low performance could be that DMPs create a single model for each degree of freedom. Valuable information regarding the correlations which exists between the joints' states and the exerted forces/torques may be lost. This can be dealt with using multi-modal motion representations which couple the joint state with the exerted forces/torques and thus create a single model using all the sensory inputs.

Furthermore, motion assessment is currently performed after the completion of the motion. A possible expansion is to assess the motion during runtime. This would significantly decrease the chance of damage for both the robot and the manipulated object. A minor issue is the high computational complexity of GPs which affects the time needed for assessment, especially on long motions. Therefore, it is planned to investigate the applicability of other, more computationally efficient models.

Finally, the main focus of the future work will be given on the development of a motion optimization primitive. This would optimize the hyper-parameters of the DMPs in such a way that the probability of a successful motion is maximized and thus will close the loop between motion and assessment primitives. The machine learning approach which will be used belongs to the class of reinforcement learning. In detail, the contact dynamics model could be exploited and so the DMPs will be optimized based on simulations of the learned model instead of the real system. Such an approach belongs to the class of model-based reinforcement [16] learning which has advantages such as minimal optimization time and also minimal risk of damage for both the robot and the manipulated objects which makes it appropriate for manufacturing tasks.

ACKNOWLEDGMENT

FleXRoP is funded by the Austrian Federal Ministry of

Transport, Innovation and Technology (BMVIT) under the program "ICT of the Future" between [2016-09 and 2018-08]. More information <https://iktderzukunft.at/>.

This project is supported by grant funding "FTI Struktur Land Obersterreich".

This project is supported by grant funding "IWB/EFRE-Programm Österreich 2014-2020".

REFERENCES

- [1] R. Hopkins and K. Jenkins, *Eating the IT Elephant: Moving from Greenfield Development to Brownfield*, ser. IBM Press. Pearson Education, 2008.
- [2] (2018) KUKA IIWA. [Online]. Available: http://www.kuka-robotics.com/en/products/industrial_robots/sensitiv/lbr_iiwa_7_r800/start.html
- [3] S. C. Akkaladevi, M. Plasch, and A. Pichler, "Skill-based learning of an assembly process," *e & i Elektrotechnik und Informationstechnik*, vol. 134, no. 6, pp. 312–315, Sep 2017.
- [4] F. Rovida, M. Crosby, D. Holz, A. S. Polydoros, B. Großmann, R. P. A. Petrick, and V. Krüger, *SkiROS—A Skill-Based Robot Control Platform on Top of ROS*. Cham: Springer International Publishing, 2017, pp. 121–160.
- [5] A. S. Polydoros, B. Großmann, F. Rovida, L. Nalpanitidis, and V. Krüger, "Accurate and versatile automation of industrial kitting operations with skiros," in *Towards Autonomous Robotic Systems*, L. Alboul, D. Damian, and J. M. Aitken, Eds. Cham: Springer International Publishing, 2016.
- [6] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233 – 242, 1999.
- [7] A. Björkelund, L. Edström, M. Haage, J. Malec, K. Nilsson, P. Noguees, S. G. Robertz, D. Störkle, A. Blomdell, R. Johansson, *et al.*, "On the integration of skilled robot motions for productivity in manufacturing," in *Assembly and Manufacturing (ISAM), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 1–9.
- [8] N. Krüger, J. Piater, F. Wörgötter, C. Geib, R. Petrick, M. Steedman, A. Ude, T. Asfour, D. Kraft, D. Omrcen, *et al.*, "A formal definition of object-action complexes and examples at different levels of the processing hierarchy," 2009.
- [9] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.
- [10] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469 – 483, 2009.
- [11] M. Deniša, A. Gams, A. Ude, and T. Petrič, "Learning compliant movement primitives through demonstration and statistical generalization," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2581–2594, 2016.
- [12] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in neural information processing systems*, 2013, pp. 2616–2624.
- [13] D. H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, 2008, pp. 91–98.
- [14] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," *Artificial Intelligence Review*, vol. 11, no. 1, pp. 75–113, Feb 1997.
- [15] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Scalable techniques from nonparametric statistics for real time robot learning," *Applied Intelligence*, vol. 17, no. 1, pp. 49–60, Jul 2002.
- [16] A. S. Polydoros and L. Nalpanitidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, May 2017.