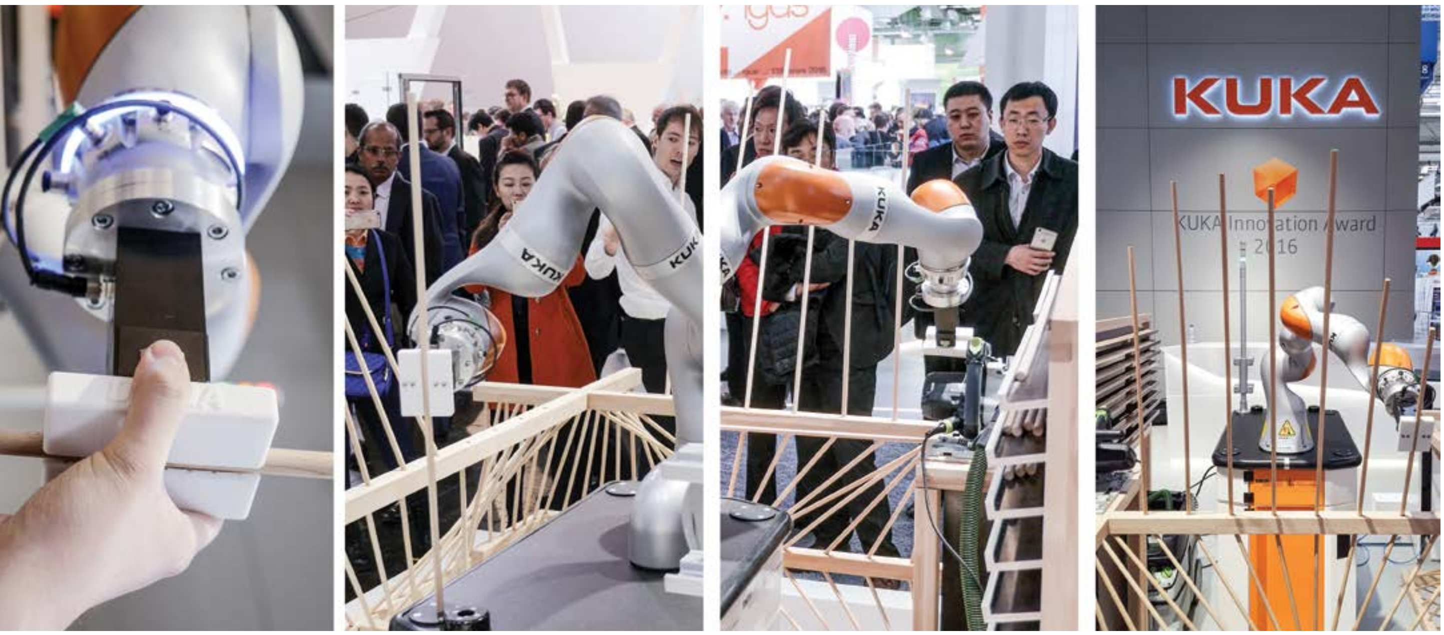
Towards New Robotic Design Tools

Using Collaborative Robots within the Creative Industry

Johannes Braumann
Robots in Architecture
UfG Linz

Sven Stumm
IP RWTH Aachen

Sigrid Brell-Cokcan
Robots in Architecture
IP RWTH Aachen



1

1 Sensitive robotic assembly informed through haptic programming. Showcased at the Hannover Fair 2016.

ABSTRACT

This research documents our initial experiences of using a new type of collaborative, industrial robot in the area of architecture, design, and construction. The KUKA LBR-iiwa differs from common robotic configurations in that it uses seven axes with integrated force-torque sensors and can be programmed in the Java programming language. Its force-sensitivity makes it safe to interact with, but also enables entirely new applications that use hand-guiding and utilize the force-sensors to compensate for high tolerances on building sites, similar to how we manually approach assembly tasks.

Especially for the creative industry, the Java programming opens up completely new applications that would have previously required complex bus systems or industrial data interfaces. We will present a series of realized projects that showcase some of the potential of this new type of collaborative, safe robot, and discuss the advantages and limitations of the robotic system.

INTRODUCTION

Ten years ago, the use of industrial robots within architectural research was considered to be high-end, often depending on robotic engineers that worked with designers to realize projects. Today, these robots have turned into well-researched tools that can be found at many larger universities, and are also starting to leave the field of research towards full-scale applications, as demonstrated in the work of Gramazio & Kohler, Achim Menges, and others (Figure 2).

Companies such as Branch Technology are now basing their innovative applications on reliable robotic platforms, thus allowing them to focus on their construction-specific tasks, rather than having to develop an entire robotic system.

Whereas robots are often used to perform tasks similar to existing CNC machines, our main research interest lies in the core of construction strategies and how we can advance robotic software technology to assist in assembling tasks (refer to the project section and Stumm et al. 2016).

Currently, significant development in the area of software can be observed in our field, enabling tablet-like interaction with the control panel and new networked devices to act as a robot controller through software options like KUKA mxAutomation (Munz et al. 2016, Braumann and Brell-Cokan 2015). Within the context of architecture and design, the community of creative robot users itself has become the main enabling factor to greatly lower the entry bar for new users of robotics by exchanging ideas on the same level and providing accessible but powerful tools such as KUKA|prc, HAL, and others.

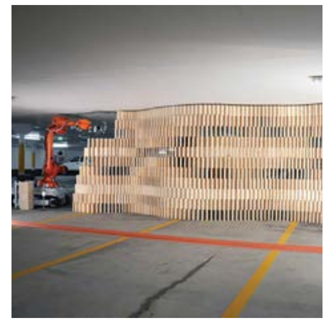
Recently, we can observe changes happening within the robot industry as well, as companies are moving their focus away from traditional robot installations within safety fences towards safe human-robot collaboration, while also investigating new applications beyond the automotive sector.

INTELLIGENT WORK ASSISTANT

For this purpose, a number of robotics companies have developed a new generation of lightweight robotic arms such as KUKA's LBR iiwa (intelligent industrial work assistant), with similar "safe" machines being available from ABB (YuMi), UR, and a range of smaller robotic startups, building upon research such as by the DLR (Albu-Schäffer et al, 2007). The iiwa (Figure 3) differs from the common robotic-arm template. Immediately visible is its design, which is optimized to reduce the potential of harming the user by minimizing the amount of sharp edges and reducing crushing hazards, thus making it possible to safely interact with the robot in a haptic way.

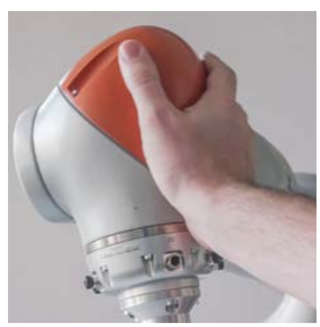
2a

2b



2c

3a

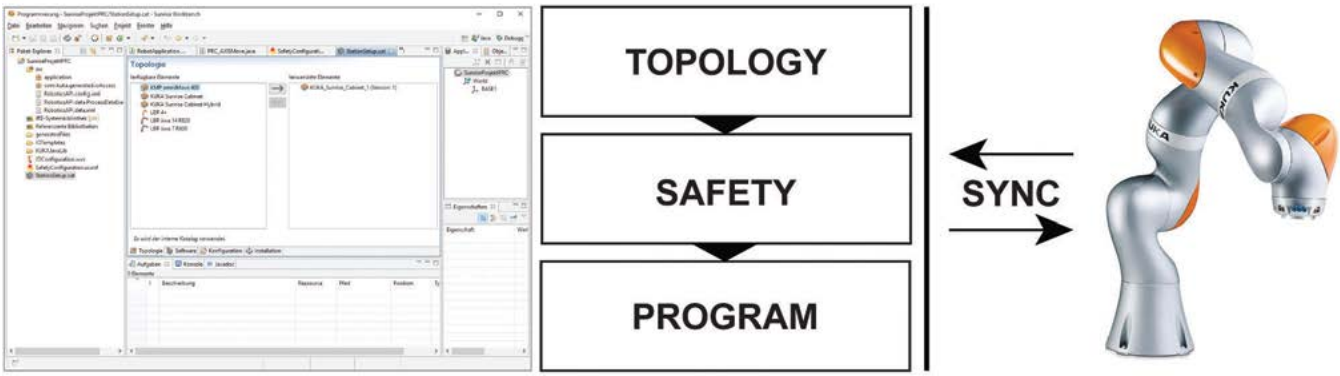


3b

2 Architectural robotic projects: ICD/ITKE Research Pavilion 2015–16 (a), Branch Technologies (b), Echord ETH Zurich (c).

3 Save interaction with a collaborative robot: Compliant mode (a), collision detection (b).

Furthermore, the LBR iiwa employs seven axes, enabling the robot to move the kinematic chain behind its tool, without moving the tool itself. The additional redundant axis greatly increases the robot's kinematic flexibility, offering mathematically infinite possibilities to move to a defined point in space, which can be used to avoid obstacles, and to offset the comparably limited axis range. The force-torque sensors within every axis constitute another step-change in robotic technologies, enabling the robot to feel and react to contact and pressure. Previously,



4 SunriseWorkbench: Defining robot topology, safety and program within the IDE and synchronizing it directly to the LBR iiwa robot.

force-torque sensors for industrial robots were mostly mounted directly between the tool and the flange, or the drive currents for individual axes were used to provide only a rather coarse feedback. Adding these sensors to every axis does not just enable force-sensitive applications, but makes the robot potentially safe to work with, as it can now feel collisions at every joint, rather than just at the tool tip.

The sensors are not limited to a Boolean yes/no collision, but provide fine-grained data that can be used for many different applications—e.g., those that require information about contact state or process force. Additionally, the robot can be moved around manually in compliant mode, which does not simply release the brakes of the motors, but compensates gravity and measures the force applied at each joint to support any user-given movement. Even the weight of workpieces can be compensated in a similar fashion, so objects mounted onto the robot can be moved as if they weighed nearly nothing.

Within the scope of architecture and design, we see a special potential for interactive design parametrization and force-guided assembly, which is further discussed in the project section.

Robot as an App

While similar applications have been possible before, the iiwa's new programming provides accessible libraries that allows the creative user to utilize these features at a high level without needing external data processing or specialized software.

Unlike previous LBR iterations such as the LBR4, which relied on the regular KUKA Robot Language (KRL), the iiwa introduced SunriseOS, which uses Java as the robot's programming language. This allows us to use state-of-the-art programming strategies, such as object-oriented programming, and most importantly, to implement external libraries that range from geometric functions to image processing and cloud networking. As an integrated development environment (IDE), KUKA provides the Sunrise Workbench, a modified version of the open-source software Eclipse. In addition to making complex tasks easier to program, it also greatly opens up the scope of robot programming. Rather than being limited to certain interfaces and tech packages, we can define custom ways of communication and interaction, e.g., using high-performance technologies developed by the game industry for Android, which also uses Java for its apps.

An exemplary programming workflow via Sunrise Workbench (Figure 4) starts with the user creating a new Java project and writing the logic using the provided KUKA-specific libraries for movements, force conditions, etc. Tools and coordinate systems are also created within the Sunrise Workbench and can be intelligently nested, so that, for example,multiple tool-tip coordinate systems are assigned to a single tool. Finally, safety-specific properties are set in the Safety Configuration and the IO configuration is loaded from WorkVisual, as with regular KRC4 robots. The project can now be synced with the robot, replacing the previous program. On the robot, the previously set-up tools and coordinate system now show up and can be calibrated, thus matching the digital environment with the actual physical space. The next synchronization between robot and IDE then pushes these changes back into the Workbench. A simulation in advance is not possible, only syntax errors are checked and highlighted.

Initial Experiences

As the iiwa robot has only recently been introduced and is priced comparatively high, we could not build upon the knowledge of many other, creative robot users. For example, applications in the creative industry are robotic scale model testing at the BRG/ETH Zurich (2016) and camera motion control for movies at CMOCOS (Shepherd and Buchstab 2014). The findings below represent our initial conclusions and subjective evaluation after slightly less than a year of using iiwa robots.

- **Acceptance:** While negative connotations for robotic arms are common, the design of the iiwa seemed to greatly increase public acceptance as it distanced itself from regular industrial robots, while also not too directly emulating human arms. People were quick to interact with the iiwa without hesitations or safety concerns, while students were curious to work with a new machine.

- **Mechanics:** Immediately noticeable is the iiwa's low weight of less than 30 kg, which makes it significantly easier to move and set up than, for example, 50 kg Agilus robots. At the same time, the larger iiwa can manipulate up to 14 kg, while the Agilus series currently tops out at 10 kg, resulting in a very favorable weight to payload ratio. The relatively low axis-speed of 57–144°/sec can be limiting for some applications, but most collaborative setups require significantly lower speeds for safety reasons. The iiwa's main drawback compared to other robots is the very limited range of its axes (e.g. +-140 at A1 and +-152.5 at A7), which is most likely the result of the effort to reduce crushing hazards, while also leading internal wiring and tubing up to the flange. Even with presumably easy movements, it has been very common for us to hit the axis limits of the machine, usually requiring fine-tuning of the posture as well as the redundant, seventh axis.

- **General Programming:** The idea of adapting an established IDE for robot programming definitely shows merit, as does the choice of Java as a well-documented language that many students have been exposed to as early as in secondary school. The KUKA libraries are well integrated and come with a certain amount of documentation, making it possible to quickly create an initial program. A significant challenge arises from the decision to only allow a single active project on the robot (though it can contain multiple sub-programs). This leads to particular challenges when multiple users are collaborating on a project, as the current system lacks any kind of version control, which gives users the choice between synchronizing the project to their local machines (thus overwriting the local project) or moving data the other way around. Therefore, any versioning has to happen before the project is uploaded to the robot.

- **Motion Programming:** While Java offers many advantages for the efficient programming of tasks, thus allowing us to work with parallel threads and asynchronous tasks, the physical robot itself can only do a single operation at once. This leads to problems with blended movements, as the robot has to know the next position in order to create a smooth trajectory without stopping at every programmed position. The user can either put multiple commands into a MotionBatch, or execute motiontasks asynchronously, both of which caused problems with very large, complex toolpaths. Troubleshooting is further complicated by the fact that the only feedback provided on the robot control panel are console messages as well as exceptions. While the older KRC4 controller presents the user a pointer at the currently active line, SunriseOS only allows the user to step through a program if a debugging process is started through the Sunrise Workbench from a remote PC. Any kind of "block selection" has to be programmed in Java, e.g., through a custom debugging interface.

- **Expandability:** Through additional libraries, it was quickly possible to integrate additional features such as a custom GUI based on the Java.Swing toolkit, as well as geometric operations that allow a similar interaction with geometry as in CAD software. See the project section for more details.

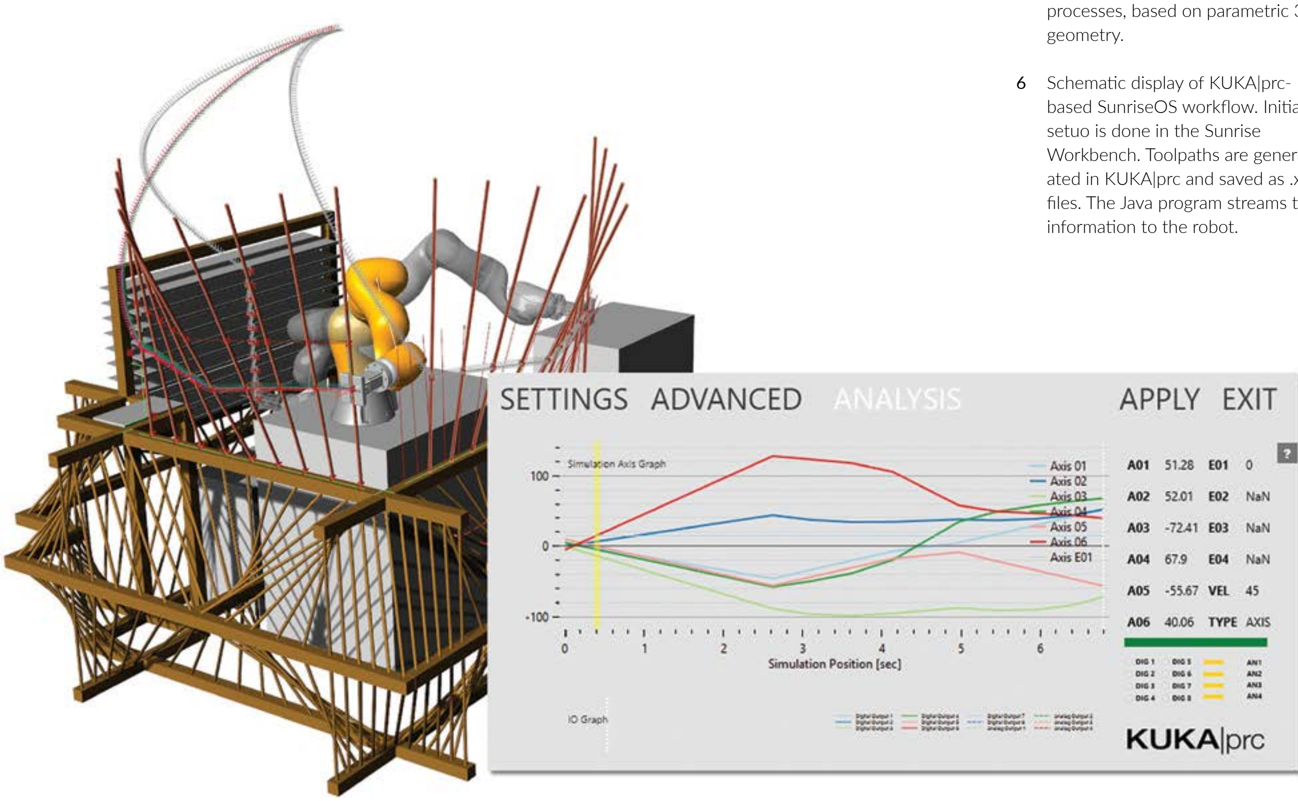
- **Safety:** For applications that involve direct interaction between the user and robot, such as collaborative assembly, but also for the initial programming of robotic tasks, especially in education, the integrated safety features of the iiwa offer significant advantages over conventional robots. With the relevant safety package, it is possible to set a global collision stop that will be applied irrespective of the particular program that is being used, and to protect that setting with a password. While these features do not make any application automatically safe—e.g., when sharp tools are involved—many applications do not require any further, external safety equipment.

NEW PROGRAMMING INTERFACES FOR COLLABORATIVE ROBOTS

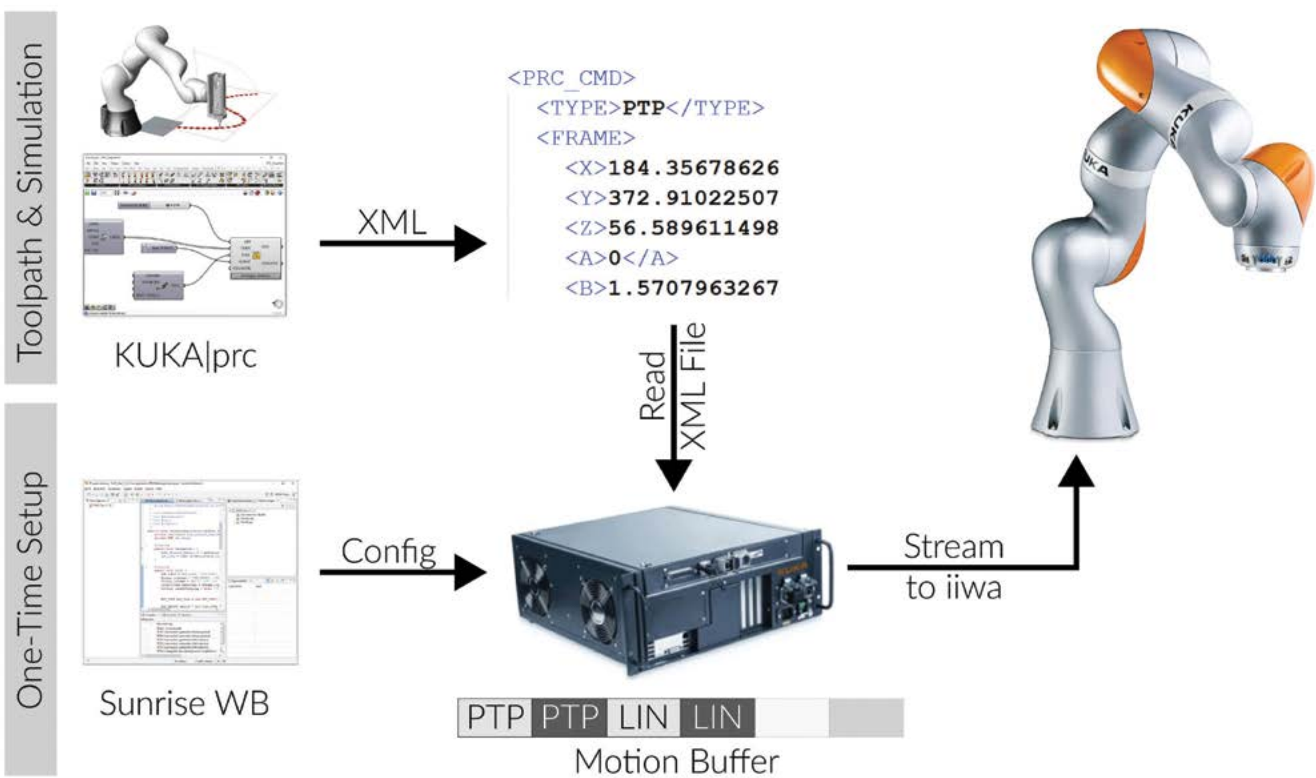
In previous research we have developed KUKA|prc (parametric robot control – Braumann and Brell-Cokcan 2011), allowing creative users to quickly and intuitively program and simulate robotic arms within a visual programming environment. For many users, this environment proved essential towards rapidly prototyping new fabrication strategies, but also for quickly engaging students. In our own teaching we have experienced it to be highly beneficial to provide students with an accessible interface that allows them to explore the capabilities and constraints of robotic arms in both a virtual environment, immediately followed by actual, physical experiments. Depending on their skills and interests, students can then either work very "deep" within the programming of the robot, or at a higher level through the visual programming environment—and of course in any combination of these two.

5 KUKA LBR iiwa simulation within KUKA|prc—in parallel optimizing the reachability of several robotic processes, based on parametric 3D geometry.

6 Schematic display of KUKA|prc-based SunriseOS workflow. Initial setuo is done in the Sunrise Workbench. Toolpaths are generated in KUKA|prc and saved as .xml files. The Java program streams the information to the robot.



5



6

We were unable to find any suitable offline programming environment capable of simulating and controlling an iiwa arm;the KUKA SimPro software was only capable of simulating the robot, but unable to output code, and a direct link for ROS would require very deep changes to the controller, as well as software that is not common outside of the core-robotics sector. Thus, we decided to integrate the iiwa robot into the KUKA|prc environment (Figure 5).

The first challenge was the definition of a new kinematic model that supports a single kinematic chain of seven rotary axes, as opposed, for example, to a six-axis robot with an external turntable. This step was taken in preparation of the iiwa's arrival, building upon our experience with the iiwa's predecessor. As literature did not provide any fully formulated solution, we experimented with using an evolutionary solver towards creating series of axis movements with the same toolframe but different position of the redundant axis. Using this empirical data, we were able to formulate and test geometric hypotheses within Grasshopper that finally resulted in a fast and reliable inverse kinematic model of the iiwa robot.

Next, we had to consider on how to transfer data from Grasshopper to the visual programming environment. We first evaluated "hacking" the synchronization process of the Sunrise Workbench to directly sync projects to the robot, but ultimately saw too many practical as well as safety issues. Therefore, we chose to use a "Firmata" by creating a standardized, base program that is capable of communicating with outside sources through defined means. A similar approach has been chosen by Andy Payne and Jason Kelly Johnson for their Firefly (2016) environment, where a firmata is uplloaded onto an Arduino controller which can then communicate with Grasshopper.

For the transfer of data between the offline programming system and the iiwa robot, we decided to utilize a human-readable format to allow quick and easy changes to the file, finally choosing XML due to the availability of powerful libraries both on the .NET side as well as for Java.

As discussed above, the blending of complex toolpaths is comparatively complicated for the iiwa, as it has to put the movement commands either into a MotionBatch, or process them asynchronously, which can lead the program to terminate early, despite several asynchronous movements commands still awaiting processing. Similar to our work with mxAutomation, which deals with several similar limitations, we therefore implemented a custom buffering system that would not process all commands in advance, but selectively group them into MotionBatches when needed and watch over the execution of asynchronous movements (Figure 6).

Rather than uploading a new project for each element, users are now simply confronted with a filebrowser that allows them to select XML files from all sources accessible to the Windows site of the controller, from network shares to USB drives. This allows us to very quickly deploy new projects to the iiwa robot, as well as effectively share a single robot between multiple groups of students.

INITIAL PROJECTS

In the past years, we have used the KUKA LBR iiwa collaborative robot in a series of research and teaching activities. The following projects were chosen to showcase some of the iiwa/Sunrise specific features that set the robotic system apart from other industrial robots.

Robotic Calligraphy was developed as a showcase installation for KUKA Robotics and the Ars Electronica Center (AEC), a digital media museum in Europe. It builds upon the concept of generating different greyscale values by rotating an asymmetric calligraphy pen that produces pure black when its wide side is normal to the toolpath, and a lighter shade the more it deviates from the normal. In previous projects and workshops (Brell-Cokcan and Braumann 2014), we have implemented the strategy within the Grasshopper environment and were therefore able to immediately test it with the iiwa by changing the code generation from KRL to Sunrise/XML. Building upon this proof of concept, it was our goal to create a completely integrated program that runs entirely on the Sunrise controller, with its own graphical user interface and without requiring an external PC (Figure 7).

The first step was to create a simple geometric library—based on the default javax.vecmath library—that allows us to work with geometric objects such as planes in a similar way as within the RhinoCommon framework. It was then very easy to layout the toolpaths and apply the rotation based on the brightness of a raster image. The comparison with the CAD output enabled an immediate error checking of the results.

Once the pathplanning was finished, we looked into the capturing of photos through a camera. Instead of depending on an industrial camera, we were able to simply attach a Logitech C920 consumer webcam to the Sunrise controller and install the accompanying driver software. We then captured an image through OpenCV, used Haar feature-based cascade classifiers (Viola and Jones 2001) to recognize faces, and cropped the image to ideally fit the face within the aspect ratio of the paper. Finally, we implemented a graphical user interface based on javax.swing that allowed the user to fine-tune the brightness and contrast of the captured image.

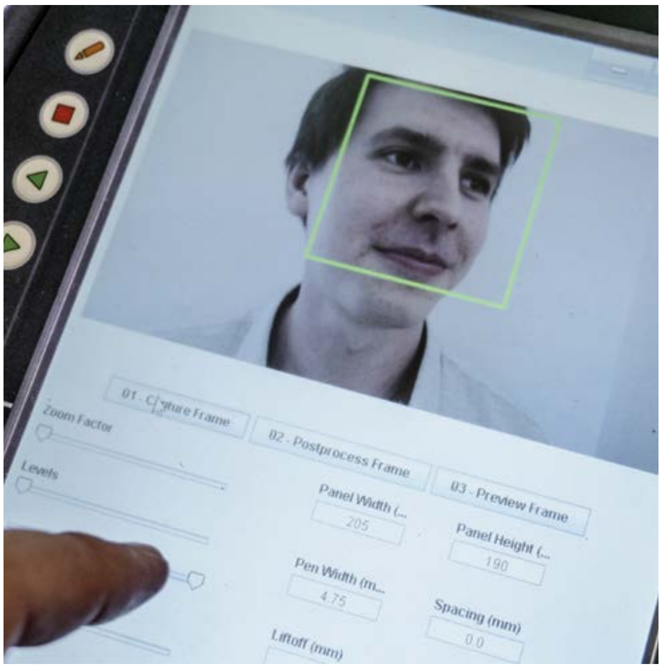
Both calligraphy pen as well as the webcam were then mounted

7a

7c

7 Image capturing with a regular webcam (a), image processing through a custom made GUI running on the smartPAD for easy interaction (b), calligraphy process (c).

8 LBR iiwa "cell" at the Ars Electronica Museum, without requiring additional safety (a). Calligraphy result (b).



7b

8a

8b

on the iiwa through a 3D-printed endeffector. The process was laid out in a way that someone could manually position the iiwa through hand-guiding, have their "selfie" taken by the robot, which the operator could then fine-tune on the control panel before starting the drawing process.

The developed system was active for more than a month at the Ars Electronica Center, operated exclusively by non-expert staff without any previous robot experience (Figure 8).

DIANA: In previous research, we have experimented with the fabrication of ruled surfaces through wooden rods, with the goal of using a small-scale robot to create a large-scale installation. Until recently, a KUKA Agilus robot only performed the cutting and multi-axis drilling of the support structure, while the rods would have to be cut and placed manually (Figure 9). The reason for that can be found in the fact that wooden rods are a natural material with very high tolerances. These tolerances can be due to improper storage and humidity, or simply because of the used wood type, and may amount to more than 10 mm on a 1000 mm rod. DIANA, the dynamic interactive assistant for novel applications, is a robot installation developed by teams from RWTH Aachen University (Chair for Individualized Production in Architecture and the Cybernetic-Cluster IMA/ZLW & IfU) and Robots in Architecture as part of the KUKA Innovation Award for the Hannover Fair 2016 that showcases the challenges of using robots in the construction industry, demonstrating concepts such as mass customization and strategies towards dealing with environments with high tolerances.

For this project, we digitally designed and built a rod-structure out of 45 x 45 mm wooden slabs and 12mm diameter birch rods. The robot setup provided by KUKA consisted of a iiwa 14 R820 robot mounted onto a flexFELLOW platform—a robot base that integrates the controller and can be quickly relocated. An important design tool from the very beginning was our iiwa simulation through KUKA|prc that allowed us to optimize the geometry in regards to reachability and collisions. While we

9a

9 Robotically fabricating the supporting structure with a regular, non-complaint robot (a). Previous, manual assembly of rods (b). New, senstitive assembly informed through haptic programming presented at the Hannover Fair 2016 (c).

9b

9c

were able to incorporate the position and orientation of each mounting point into the Java program, these values assume an ideal, digital environment and take neither production tolerances, nor an imprecise placement of the robot itself into account.

First of all, the user takes the iiwa robot and manually guides it from one side of the base structure to the other side, a process we refer to as "haptic programming" (Stumm et al. 2016). By capturing the movement, a curve is parametrized that informs the fabrication process, making every generated structure unique. Based on this data, the robot generates a list of tasks for that fabrication process: First of all, the robot takes a rod out of the supply station, which groups rods within a range of 50 mm. In order to calculate the exact length of the rod, the robot moves each tip of the rod onto the flexFELLOW platform until a contact is established. This contact is established solely through the force-torque sensors, without requiring additional sensory equipment. The robot records the distance between the gripper fingers and the measuring position, thus being able to calculate the length in each direction. Using a common circular saw, the rod is then cut according to the previously generated curve. For the actual assembly, the robot moves the rod into the approximate position of each mounting position as known from the design. Due to the high material tolerances, it then searches for the exact position via the force sensors. Once it feels a significant drop in the force values, it concludes that the rod has slipped into mounting position and continues the assembly process by retrieving the next rod (Figure 10).

In the past, such systems have mostly relied on external measuring equipment that calculates the offset between the ideal and the actual toolframe. However, such systems are complicated to use and not ideally suited to the rough, changing environments of a construction site. The force-sensitivity of the iiwa provides a well-integrated way towards implementing such functionality into an assembly process, without requiring external equipment or special software.

CONCLUSION AND OUTLOOK

The KUKA LBR iiwa represents a significant step change not only for the robotics industry, but also for the construction industry, as it enables completely new ways of interacting with robots;both in regards to physical and digital interaction. Even more than previous machines, the robot becomes a universal platform that can be adapted through software for specific tasks and implemented into existing systems, from the cloud to Building Information Modeling.

Using the prepared "firmata" allows users to quickly prototype processes and check reachability through the KUKA|prc and the visual programming environment, where toolpath layouts can be more easily constrained to geometries. Within Sunrise Workbench, the user can continue working with the automatically generated XML file, adding, for example,additional safety and interaction features around it.

At the moment, the iiwa's working range limits larger-scale applications within a construction context. However, the transfer of

10

the robotic system onto a mobile platform (KMR iiwa, Figure 11), will allow us to navigate autonomously within a workspace that is limited only by the capacity of its batteries.

ACKNOWLEDGEMENTS

This research was performed as part of the "Robotic Woodcraft" research project (FWF-PEEK AR238) and the KUKA Innovation Award. The DIANA team consisted of Sven Stumm, Martin von Hilchen, Elisa Lublasser and Prof. Sigrid Brell-Cokcan (IP-RWTH Aachen), Philipp Ennen and Prof. Sabina Jeschke (IMA/ZLW & IfU RWTH Aachen) and Johannes Braumann (Robots in Architecture).

REFERENCES

Albu-Schäffer Albin, Sami Haddadin, Christian Ott, Andreas Stemmer, Thomas Wimböck, and Gerd Hirzinger. 2007. "The DLR Lightweight Robot: Design and Control Concepts for Robots in Human Environments." *Industrial Robot: An International Journal* 34 (5): 376–385.

Braumann, Johannes and Sigrid Brell-Cokcan. 2011. "Parametric Robot Control: Integrated CAD/CAM for Architectural Design." In *Proceedings of the 31st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA).* Banff, Alberta, Canada: ACADIA. 242–251.

Braumann, Johannes, and Sigrid Brell-Cokcan. 2015. "Adaptive Robot Control - New Parametric Workflows Directly from Design to KUKA Robots." In *Real Time: Proceedings of the 33rd eCAADe Conference*, vol. 2, edited by B. Martens, G. Wurzer, T. Grasl, W. E. Lorenz, and R. Schaffranek. Vienna, Austria: eCAADe. 243–250.

Brell-Cokcan, Sigrid and Johannes Braumann. 2014. "Robotic Production Immanent Design: Creative Toolpath Design in Micro and Macro Scale." In *ACADIA 14: Design Agency—Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture*, edited by David Gerber, Alvin Huang, and Jose Sanchez. Los Angeles: ACADIA. 579–588.

ETH Zurich. "Robotic Scale-Model Testing". Accessed May 5th 2016. http://www.block.arch.ethz.ch/brg/research/robotic-scale-model-testing

Munz, Heinrich, Johannes Braumann, and Sigrid Brell-Cokcan. 2016. "Direct Robot Control with mxAutomation." In *Rob | Arch 2016: Robotic Fabrication in Architecture, Art and Design 2016*, edited by Dagmar Reinhardt, Rob Saunders, and Jane Burry. Vienna: Springer. 440–447.

Payne, Andy and Jason Kelly Johnson. "Firefly." Accessed May 5th 2016. http://www.fireflyexperiments.com

Shepherd, Stuart and Alois Buchstab. 2014. "KUKA Robots On-Site." In *Rob | Arch 2014: Robotic Fabrication in Architecture, Art and Design*, edited by Wes McGee and Monica Ponce de Leon. Vienna: Springer. 373–380.

Stumm, Sven, Johannes Braumann, Martin von Hilchen, and Sigrid Brell-Cokcan. 2016. "On-Site Robotic Construction Assistance for Assembly Using A-Priori Knowledge and Human-Robot Collaboration." In *Proceedings of the International Conference on Robotics in Alpe-Adria-Danube Region*. Belgrade, Serbia: RAAD.

11

10 DIANA: Capturing geometry through manual guiding (1), measuring exact length (2), cutting (3), force-sensitive assembly (4).

11 KMR iiwa platform at the Chair for Individualized Production in Architecture at RWTH Aachen.

Viola, Paul, and Michael Jones. 2001. “Rapid Object Detection using a Boosted Cascade of Simple Features.” In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. Kauai, HI: CVPR. 551–560.

IMAGE CREDITS

Figure 2a: © University of Stuttgart
Figure 2b: © Branch Technologies
Figure 2c: © ETH Zurich
All other figures: © Robots in Architecture and RWTH Aachen University

Sigrid Brell-Cokcan and Johannes Braumann founded the Association for Robots in Architecture in 2010 with the goal of making industrial robots accessible to the creative industry. Towards that goal, the Association is developing innovative software tools such as KUKA|prc (parametric robot control) and initialized the Rob|Arch conference series on robotic fabrication in architecture, art, and design which–following Vienna in 2012, Ann Arbor in 2014, and Sydney in 2016 – will be held 2018 in Zurich. Robots in Architecture is a KUKA System Partner and has been validated as a research institutions by national and international research agencies such as the European Union’s FP7 program. Recently, Sigrid founded the new chair for Individualized Production in Architecture at RWTH Aachen University. Johannes is heading the robotics lab at UfG Linz and leading the development of KUKA|prc. Their work has been widely published in peer reviewed scientific journals, international proceedings, and books, as well as being featured in formats such as Wired, Gizmodo, FAZ, and RBR.

Sven Stumm is a computer scientist at the Chair of Individualized Production in Architecture (IP) at RWTH Aachen University with a focus on intelligent systems and backgrounds in humanoid, mobile and stationary industrial robotics. His core competences are in robot controller development, sensor data processing, probabilistic modelling as well as software development for programming and simulation of robotic applications. At IP he researches assembly and production processes within construction, and novel interactive interfaces for robot programming.