Human-Machine Interaction for Intuitive Programming of Assembly Tasks in Construction

Sven Stumm\(^a\),*, Johannes Braumann\(^b\),c, Sigrid Brell-Cokcan\(^a\)\(^c\)

\(^a\)Individualized Production, RWTH Aachen University, 52062 Aachen, Germany
\(^b\)Usg Linz, 4040 Linz, Austria
\(^c\)Association for Robots in Architecture, 1060 Vienna, Austria

* Corresponding author. Tel.: +49-241-80-95013; fax: +49-241-80-92374. E-mail address: stumm@ip.rwth-aachen.de

Abstract

A variety of robot programming techniques exists ranging from constraint based over skill based programming to learning by demonstration. In order to extend their applicability propose to join some of these approaches. We therefore combine visual CAD based programming with skill based programming through demonstration. This constitutes the basis of the outlined strategy. We then employ human feedback through hand gestures for incremental parameter modification. We propose this approach in order to potentially lower times to production for new products and allow efficient use of robotics in low lot-sizes especially in the context of assembly for construction.

Keywords: Robot programming, Skill based programming, Visual programming, CAD based programming, Gesture recognition, Construction

1. Introduction

Classic industrial robot programming is a static process: In order to achieve a certain motion, a set of predefined positions in Cartesian coordinates or axis angles are used for motion planning. This process creates a predictability within robot positioning but does not offer the flexibility that is required to deal with high tolerances. To compensate for measurement differences and product variations sensors can be used for dynamic adaptation of positions. This approach is especially feasible in conjunction with high product quantities. However, in order to achieve lower changeover times and to utilize the flexibility of robotics for mass customization and the production of individualized products new methods need to be integrated. A number of approaches were created within mobile and service robotics which have to be able to work within unknown, dynamically changing and seldom predictably environments. Though a wide variety of approaches exist, the common denominator is a process of autonomous decision making. The results of this process can either be unpredictable, as is the case with autonomous behavior, or hard to define and program, as is the case with state machine based approaches.

The complexity of working with uncertainty and tolerances requires a deeper knowledge in multiple areas of robotics. In order to make robotics accessible to a wider range of workers this complexity needs to be mediated. While constraint based robot programming [1] made accessible implementation for a variety of tasks possible it still requires detailed knowledge of control theories and controller strategies. Constraints need to be defined and the resulting motion is not intuitive for people with a less technical background. The application of robotics in construction constitutes an ideal and difficult testing environment. Construction environments are inherently unstructured. Due to fluctuations within the work force a number of untrained personal needs to be able to control and manipulate robotic tasks. In most cases robots also need to perform their tasks without separating safety measures in direct collaboration with human workers. The Association for Robots in Architecture works on making robotics accessible to the creative community. An approach that had significant success was the combination of graphical programming of parametric geometries with robotics, allowing for parametric robot control and mass customization.

While being easy to use, this user-friendly approach focuses on CAD based robot programming and currently does not include sensor feedback. New industrial controller systems allow a more direct access to robot actuators and a more extensive use of sensors. At the same time industrial robot manufactures want to extend the use of robotics through mobile platform development.
Both developments need to be considered for easy-to-program interfaces for new industrial robots. We present our first concepts and results for a new combined approach towards transportable robot programming. The approach combines interactive, skill based [2], constraint based [1] and parametric robot programming [3]. Within this paper we present the considered methodology as illustrated in figure 1 that extends the parametric robot programming. The central building block will be described in more detail in section 6.

Fig. 1. The programming methodology.

2. Robots in the Construction Industry

The construction industry has always aspired to reach a similar degree of automation as the automotive industry. However, where the production lines of the automotive industry are fabricating a single product with only slight variances over the entire product’s lifetime, especially the high-end construction industry deals with small lot sizes where in an extreme case a façade structure may consist only of unique panel geometries. Currently, many building components are therefore manually fitted on-site by workers, especially in low-wage countries where construction tolerances are generally accepted to be higher. Strategies such as BIM (Building Information Modeling) [4] attempt to control construction processes in a more effective fashion by working with intelligent objects that can be assigned properties and parameters. Rather than drawing lines and applying hatchings to declare materials, objects such as walls are directly placed in a 3D-environment and given parameters that set their construction phase, material, and even costs. This results in a model theoretically capable of organizing the assembly of highly complex building structures. However, in order to communicate with the workers on-site, this complex model has again to be broken down into printed, physical plan drawings, which are then followed by workers with a certain amount of autonomy. In practice, even the most well-planned construction site is therefore an unstructured environment where the current state of the site only loosely correlates with the building information model’s construction phase. Thus, the main challenges of automation in the construction industry are the small lot sizes and the unstructured environments, coupled with the generally high complexity of modern buildings. In order to establish robots in construction they need to be able to mediate the complexity and should therefore be easily programmable for a wide variety of tasks.

3. Parametric Robot Control

Commercial software for the programming and simulation of robotic arms such as KUKA SimPro are modeled after the teaching by demonstration process of robotic arms. Rather than guiding the physical robot, a digital model is placed in 3D-space and the robot’s positions are recorded. These movement component can then be coupled with logic components to handle subprograms, loops, conditionals, etc. For many industrial processes, such a work flow is ideal as it allows an intuitive and safe interaction with the robot. While it requires a relatively large time-investment into the programming of more complex cycles, the generated revenue by the mass-production of the resulting element easily offsets the programming costs. Challenges only arise with smaller lot sizes, where the programming makes up an increasingly large part of the costs of the final product. This is exacerbated in the construction industry, where complex, free-formed buildings can consist of countless individual parts. In order to cope with this complexity, we can utilize the fact that these individual parts are often based on a single, global topology that is then adjusted to fit the local conditions, e.g. a façade that uses a standardized aluminum profile system to create simple n-gons, which can vary in size, angle, and filling material (transparent/opaque). Rather than manually drawing hundreds of individual panels, parametric design software today allows architects and designers to create a single, parametric model that can be dynamically fitted to the local conditions. In previous research we have developed software tools that allow us to directly integrate the fabrication logic into the same parametric model that controls the geometric shape of the model [5]. Thus, whenever a parametric object is adapted to new local conditions, not only its geometry but also its construction logic is updated, enabling the immediate robotic fabrication of the part as is illustrated in Figure 2.

Fig. 2. Parametric robot control through KUKA|prc: Visual programming components (below), robot simulation with axis graph (above).
This process was implemented in Grasshopper, a visual programming environment built upon the CAD software Rhinoceros that is rapidly gaining popularity in the AEC fields (Architecture, Engineering and Construction). The main advantage of Grasshopper is its accessibility, modularity and visual representation [6]: Rather than typing code, the user simply picks components from a large library that comes with the software. Each component performs a mathematical or geometrical operation, taking data from the inputs on its left side, and outputting the results on the right side. Parametric relationships are defined by linking components with each other, thus creating an increasingly complex directed, acyclic graph. Our software expands the capabilities of the visual programming environment beyond parametric design by integrating a series of components that allow the programming and simulation of robotic arms in an environment that suits itself particularly well for such applications, as e.g. new tools can simply be plugged and unplugged, similar to the physical robot.

As a natively integrated component our software can therefore be directly linked to the parametric design components that define the local form of a global construction component, as illustrated in figure 3. Therefore, when any changes are performed to the parametric geometry, they propagate through the graph and immediately affect the fabrication strategy and the robotic path planning. We refer to this process as production-immanent design [7] as it allows the designer or engineer to optimize structures in a very intuitive way as he is permanently provided with fabrication feedback. At the same time, we can use the strategy for mass customization, automatically writing the robot control data files for all components deriving of the same parametric model.

This process is especially useful for applications that are inherently repetitive, such as complex multi-step assembly processes that combine simple manipulation of objects with e.g. subtractive methods. Through the visual programming environment, the fabrication-steps can be tested and evaluated individually, and finally assembled into a single program - without incurring a significant overhead compared to common off-line programming methods.

4. Adapting to Unstructured Environments

While new parametric programming strategies allow us to fabricate self-similar building components with a minimum of programming effort per element, the challenges of working within an unstructured environment remain. In the past, attempts have been made to structure construction sites more tightly, as e.g. in the Japanese drive towards construction robotics in the 1980ies [8], which resulted in the financial collapse of several innovative companies. Thus, we prioritize the adaption of the robot to the construction site, over the adaption of the construction site to the robot. In order to accommodate this adaption, we see the need for bringing parametric fabrication strategies from the office space directly to the construction site: At the moment, the parametric robot control described in Section 3 is done mostly offline within the simulation environment. The resulting files are then sent to a fabricator (or in-house fabrication shop) for manufacturing and then transported to the construction site.

Instead, we want to integrate fabrication strategies directly into advanced robot controllers such as the KUKA iiwa’s SunriseOS so that the parameters that define the local object can be gathered on-site, e.g. by manually guiding the robot along the outline of the building component. Furthermore integrating external vision sensors allows us to create a reconstruct model of the environment. We employ simultaneous localization and mapping techniques [9] for an approximation of the workspace and use object reconstruction to model possible work pieces as illustrated in Figure 4. By fitting a parametric model to the object the system is able to identify the corresponding manufacturing strategy. Furthermore features of the reconstructed object can be identified and used as reference points within the assembly planning.

5. Adapting to Assembly in Unstructured Environments

The process of joining, handling and position adjustment are central assembly tasks. Parametric robot programming was developed with geometric form in mind within the context of parametric CAD. While this is important for individual assembly parts, common assembly strategies consist of a sequence of assembly operations and therefore require a more iterative
procedural planning. Within skill based robot programming elementary behaviors are defined as a skill set, which is in turn used for state transitions within the working environment. This approach seems feasible for assembly tasks due to parallels in structure of assembly plans and skill sequences. Using a priori knowledge from assembly planning can also lead to a more complete model of the assembly process [10]. Pederson et. Al. [2] described significant success in planning robotic production on a skill basis, specifically lowering time to production. By using skills created by experts the programming effort could significantly be reduced and untrained workers were enabled to use robotics. The skill however were created by experts in robotics, while considering a number of constraints. The skills themselves were classified based on industrial standard operating procedures (SOP), and required a description which allowed the handling of a variety of forms. This however hampers the extension of the skill catalog. Furthermore skill based robot programming requires a high amount of preplanning which can also lead to planning errors, if the environmental conditions change, as is often the case in dynamic and unstructured environments. Assembly operations focus on relative positions between assembly parts. To define these position consistently for variations of assembly parts, they also need to be defined relative to recognizable part features. Constraint based robot programming allows for exactly that, by viewing the resulting robot path as an optimization problem that can be determined using feature based constraints, which makes it ideal for a sensor based robot path planning. With the main disadvantage being, that solving optimization problems can take a considerable calculation time in larger contexts. We therefore want to combine the advantages of these approaches and overcome the disadvantages, in order to achieve intuitive means for robot programming in unstructured environments for in-situ assembly at construction sites. Previously we described the advantages of learning by demonstration for intuitive teaching of simple tasks, which we therefore employ for the creation of assembly skill sets. These skills are in turn generalized with human feedback for later parametrization. The principles of parametric robot programming are then employed for constraint definition. Sensor based environment modeling and object detection allows us to define these constraints as relative to features of real assembly parts. Applying a solver for on-site optimization of only simple assembly steps and not the overall assembly strategy reduces calculation time significantly. Based on assembly plans an overall strategy can be created, this strategy however needs to be dynamically adapted to the current state of the environment. For a more detailed explanation of the approach and a first verification of the possibilities we will focus on a simplified look at brick laying tasks. Excluding a number of constraints as well as necessary application of mortar the process can be viewed as a pick and place task.

6. Teaching Process and Analysis of Generalization

As illustrated in figure 1 the teaching process is straightforward. Before the teaching by demonstration process of an assembly skill begins, a skill class is selected. These classes are defined in accordance with standardized assembly operations (As given by VDI 2860 and DIN 8593) allowing also the definition of new custom classes. For the picking process the Pick Operation class is selected. After this simple selection for later assembly step planning the skill is taught and the motion is recorded. The skill is then replayed and analyzed, changes in direction or I/O control, i.e. gripper control, are detected. Using these points of interests in the path as well as sensor based detection of environmental changes a number of possible parameters are suggested. This Process generates a variety of possible generalization scenarios. Through user feedback specific parameters can be constraint or excluded, i.e. removing rotation around the Y- or X-axis of the target positions for a specific skill. Within the context of motion sequence required parameters and the likelihood of each generalization scenario is determined. A currently manual parametrization selection of skills leads to an execution strategy. The parameters of each skill are dynamically optimized on-site using a solver to ensure its execution within a dynamically changing environment. The initial state and the initial construction constraints are parametrized using sensor feedback. Figure 5 illustrates a simple scenario of parameter fitting on-site.

![Offline planning](image1.png) ![On site planning](image2.png)

Fig. 5. On-site assembly fitted to room layout avoiding collision with the environment.

In application the process consists of the following scenarios: Selecting Skills, Parametric planning skill-constraints on-site and Teaching of new skills. If a skill set already exists the skill sequence or rather execution strategy is currently set manually. Meaning that skills or actions, i.e. gripping an object from a specific approach vector or similar, are associated with specific positions. These positions are defined within the second scenario through constraints. Employing simultaneous localization and mapping techniques a 3D occupancy map of the environment is created. The occupancy map as well as a more detailed approximation of the environment employing a 3D reconstruction of last scanner point clouds are used to allow for parametric programming on-site. The fabrication logic associated with parametric programming allows for a simplified definition of constraints based on parametric assembly part geometry. The parameters of the model are fitted to reality using a comparison with the sensor information of the part. Therefore the fitting parametrization of the skill is chosen for a specific part and further optimized during execution using the constraint based solver. The teaching of new skills is based on existing techniques. Niekum et.AI. [11] uses a Hidden-Markov-Model to segment a taught motion into a sequence of skills. We propose following a similar approach when analyzing the time series of the motion for possible generalization scenarios. Within a skill a number
of smaller state changes can be observed. In case of a pick skill, the robot is at the gripping position and the I/O signal for the gripper is changed, the direction of the gripper is changed away from the object and an object is removed from the workspace. Based on the skill class most of these points of interest could already be inferred, but are reaffirmed by path analysis. The skill itself can be seen as a sequence of motion primitives. The approach and retreat motion primitives towards these positions are of special interest and can be viewed as dependent parameters. Accordingly the degrees of freedom for the approach and retreat motion primitives can easily be defined with regards to the tool frame. These motions can therefore be defined as a set of approach and retreat angles, distances and speeds, or as relative six degree of freedom Cartesian positions. Employing a vision based reconstruction of the workspace, the points of interest can be defined relative to object features, either of the moved object, or another workspace feature. Time dependent periodic motion behavior can only be modeled through I/O changes, which is common practice in industrial robotics. In static environments the execution time is also static and can easily be determined beforehand. In dynamic environments the execution time can either be calculated after parametrization of the task or to achieve static execution times distance and speed need to be modeled as dependent variables, which only works if the robot does not already work close to its maximum speed.

The teaching process is reduced to simple skill teaching by demonstration. Constraints can be modified through human gestures by making positions relative to object features. The current concept plans for the gesture based definition of functions starting with base function definition and adding gesture based parameter selection for these base functions. Figure 6 (a) illustrates a simple block stacking operation with human interaction based parametrization of constraints, i.e. size and course off the wall, as well as the parametrized stacking pattern. The corresponding assembly plan is shown in figure 6 (b).

7. Synopsis

In this paper we described the advantages of different programming approaches for robot assembly path planning in unstructured environments, i.e. construction sites. A possible combination of methods was illustrated and a first overview was given in the form of a pick & place stacking operation of building elements similar to brick laying. Figure 7 shows an approach for manufacturing and assembly of Styrofoam building blocks. By building upon the interaction based procedure of learning by demonstration and gesture based parametrization of simple functions we actively try to reduce the required knowledge of robotics, which is a major obstacle in the application of robotics in construction. By visualizing the problem in a CAD tool we reduce robot programming to parameter based CAD modeling, while simultaneously allowing parameter based modeling on site, using a sensor based environment model as illustrated in Figure 4. We therefore actively want to bridge the gap between theoretical modeling and practical realization. While parametric robot control requires extensions to work in unstructured environments. We demonstrated that this can be achieved intuitively through a combination with other existing approaches. While simultaneously ensure the existing scalability and adaptability of the method as shown in figure 7. In Future work we will focus on the extension of the approach in the area of modeling force-torque based assembly procedures.

A demonstrative prototype is developed as part of the Consortium for Intelligent Robotic Assistants (CIRA - http://www.robotic-assistants.de/) to be presented within the KUKA Innovation Award 2016 at the Hanover Fair.

Fig. 6. (a) Assembly planning with man-machine interaction; (b) The corresponding assembly plan.

Fig. 7. Small-scale prototyping of a combined milling-hotwirecutting-assembly process (above), large-scale fabrication (below).
References

[1] Schutter, J.D., Laet, T.D., Rutgeerts, J., Deer, W., Smits, R., Aertbelin, E., et al. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. The International Journal of Robotics Research 2007;26:433–455.

[2] Pedersen, M.R., Nalpantidis, L., Andersen, R.S., Schou, C., Bgh, S., Krger, V., et al. Robot skills for manufacturing: From concept to industrial deployment. Robotics and Computer-Integrated Manufacturing 2016;37:282–291. doi:http://dx.doi.org/10.1016/j.rcim.2015.04.002.

[3] Braumann Johannes; Brell-Cokcan, S. Adaptive robot control. eCADe 2015 Real Time- Proceedings of the 33rd International Conference in Computer Aided Architectural Design in Europe 2015;:P .243–251.

[4] Eastman, C., Teicholz, P., Sacks, R., Liston, K., BIM Handbook - A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors. Wiley; 2011.

[5] Braumann Johannes; Brell-Cokcan, S. Visual Robot Programming - Linking Design, Simulation, and Fabrication. 2014.

[6] Davis, D.J.B., Burry, M. Understanding visual scripts: Improving collaboration through modular programming. International Journal of Architectural Computing 2011;Vol.9:361–375.

[7] Braumann Johannes; Brell-Cokcan, S. Robotic production immanent design: Creative toolpath design in micro and macro scale. CADIA 2014: Design Agency, Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture 2014;:P 579–588.

[8] Bechthold, M. The return of the future: A second go at robotic construction. Architectural Design 2010;Vol.80:116–121.

[9] Thrun, S., Burgard, W., Fox, D. Probabilistic robotics. MIT Press; 2005.

[10] Stumm, S., Kuhlenkötter, B., Hoffmann, A., Seebach, H., Reif, W. Towards combining layout and process models for mixed assembly facilities. AUTOMATION 2014 Smart X Powered by Automation 2014;

[11] Niekum, S., Osentoski, S., Konidaris, G., Barto, A.G. Learning and generalization of complex tasks from unstructured demonstrations. IEEE/RSJ International Conference on Intelligent Robots and Systems 2012;