Athens Institute for Education and Research
ATINER

ATINER's Conference Paper Series
IND2015-1640

New Developments in Industrial Robot Programming

Peter Hess
Professor
Nuremberg Institute of Technology
Germany

Maximilian Wagner
Scientific Assistant
Nuremberg Institute of Technology
Germany

An Introduction to ATINER's Conference Paper Series

ATINER started to publish this conference papers series in 2012. It includes only the papers submitted for publication after they were presented at one of the conferences organized by our Institute every year. This paper has been peer reviewed by at least two academic members of ATINER.

Dr. Gregory T. Papanikos
President
Athens Institute for Education and Research

This paper should be cited as follows:

Hess, P. and Wagner, M. (2015). **"New Developments in Industrial Robot Programming",** Athens: ATINER'S Conference Paper Series, No: **IND2015-1640.**

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece
Tel: + 30 210 3634210 Fax: + 30 210 3634209 Email: info@atiner.gr URL: www.atiner.gr
URL Conference Papers Series: www.atiner.gr/papers.htm
Printed in Athens, Greece by the Athens Institute for Education and Research.
ISSN: 2241-2891
13/10/2015

New Developments in Industrial Robot Programming

Peter Hess
Professor
Nuremberg Institute of Technology
Germany

Maximilian Wagner
Scientific Assistant
Nuremberg Institute of Technology
Germany

Abstract

Industrial robots are normally used in mass production, but to a greater extent the tasks of robots have more variations. A key driver to greater flexibility is the future of the production industry 4.0. The main issues in industrial robot programming to fulfill the requirements of industry 4.0 are automated robot programming, intuitive programming and operation and easy calibration. These new programming concepts are presented with two projects (RoboPainter and automated programming of cooperation robots).

Keywords: Industrial robots, Industry 4.0, Intuitive robot programming

Introduction

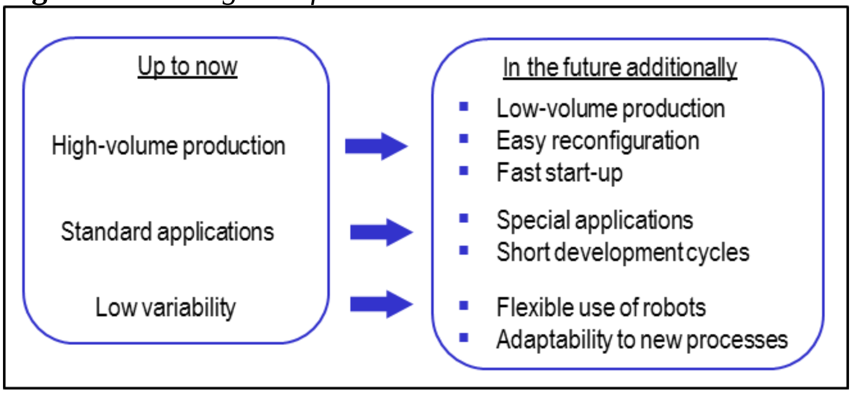
Formerly industrial robots were successfully used in mass production. The robot programs covered all variants of production and were elaborately programmed and tested. Especially in the automotive industry many robots were used for welding, painting, assembly, and other applications.

Now there is a paradigm shift in the use of industrial robots (Figure 1). Not only mass production, but also low-volume production up to batch one focuses on the automation of the robots. For this the development a new robot task in short time is necessary, so that the robot can flexibly change from one task to another. Often this is called “plug and produce” (Naumann and Verl, 2012). More and more sensors are integrated in the robot cells. With these sensors the robot can react autonomously on variations of the work pieces (Haag et al., 2011). So many new applications are opened to industrial robots.

In 2011 the production initiative industry 4.0 started (DKE, 2013; Lee et al., 2014). By using the processors in machines and parts (cyber physical systems) the production became more flexible. Machines and parts recognized problems and errors as early as possible and were able to repair these errors.

First, this paper shows, how industrial robots are programmed today. Then it gives a short introduction of industry 4.0 and its requirements on robot programming. The Chapter “*Development Directions of Robot Programming”* discusses new developments in robot programming including two examples. The article ends with some concluding remarks.

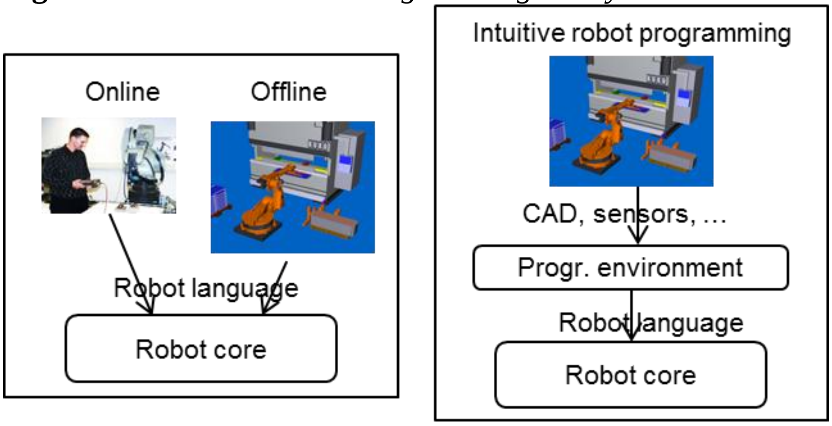
Figure 1. *Paradigm Shift in Industrial Robotics*



Industrial Robot Programming Today

Nearly all the controllers of industrial robots have their own robot language as an interface to the robot control core (Figure 2, left). Robot programs are written in this language either directly on the robot with a teach panel or with an offline programming and simulation system. But it is always necessary to program each single motion and to give exactly all the target positions.

Figure 2. *Industrial Robot Programming Today*

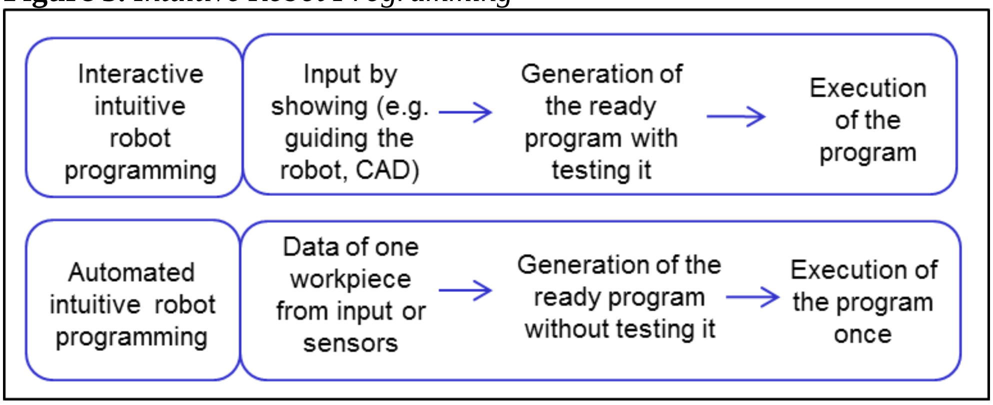


To reduce the complexity for the end user, the concept of intuitive robot programming was developed (Pan et al., 2010; Barth and Henrich, 2012). A programming environment, that is valid for a special application area, is written (Figure 2, right). The explicit robot program is generated by the programming environment out of different inputs (CAD-data, sensor data, guiding the robot, voice, etc.). It is not necessary, that the end user writes the statements in the robot language.

There are two lines of development in intuitive robot programming (Figure 3). Some programming environments support the end user with programming the application (upper line in Figure 3). With easy to use input media, like guiding the robot or showing the robot positions in an augmented reality, the end user gives the outline of the program. The exact program is generated automatically. Then it is tested and can be used many times (Soller and Henrich, 2009).

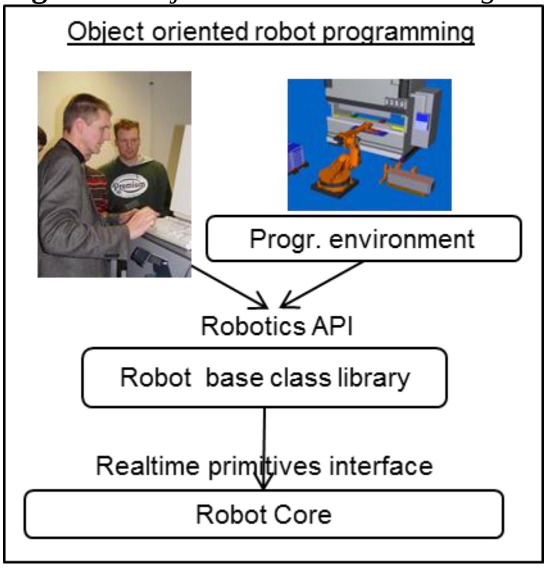
Other methods (lower line in Figure 3) use batch-one-production with industrial robots. Necessary information about the work piece is acquired with a camera or other sensors. Out of this information a special robot program is generated and directly executed. So, any variation of the work pieces can be processed, without writing a program for each work piece (see Chapter "*Development Directions of Robot Programming*").

Figure 3. *Intuitive Robot Programming*



Many of the languages that are used in industrial robot programming today have deficits, e. g. in parallelism or object orientation. So, in different projects object oriented programming environments for robots were developed. In the project SoftRobot (Angerer et al., 2010) the robot core has an interface, called the real-time primitives interface (Figure 4). The functions of this interface are path generation, sensor input, signal output and so on. These functions are used in the robotics base class library that provides a robotics API. This robotics API is the interface of the user or of a programming environment for intuitive robot programming.

Figure 4. *Object Oriented Robot Programming*



Industry 4.0 and its Demands on Robots in the Production

The idea of industry 4.0 is that the production will be more flexible and more autonomous. The base of this development is cyber-physical systems (cps). A cps is an object that has a processor with knowledge about its surroundings and that is connected to other cps via internet. Additionally it has sensors and actors that physically interact with its surroundings.

To reach that vision, work pieces, controls, and other objects of the production must be built as cps. Knowledge about the organization of the production must be integrated in the processors and, where necessary, sensors and actors must be added.

Industrial robots are important players in this concept. They will be complemented by mobile robots which take over the material transfer and can support the industrial robots, if necessary (Bogh et al., 2014).

From the scenario of industry 4.0 there are three demands on robots in the production.

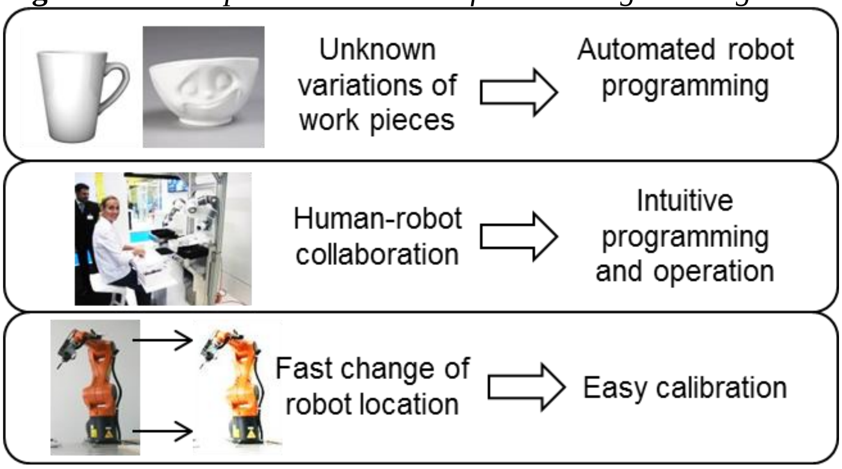
- Robot controls must have knowledge about the organization of the production and about possible failures, to autonomously react without the intervention of a worker.
- More and more humans and robots will work together. So, the concept of human-robot collaboration HRC is one of the main topics in industry 4.0.
- Variations in the utilized capacity of the production will grow. For that, it is important, that production capacities can be switched from one product to another. Therefore, industrial robots must be changed fast from one product to another or from one place to another. The key concept for this is “plug and produce”.

Development Directions of Robot Programming

From Industry 4.0 to Robot Programming

Taking into account the demands of the previous chapter, end user robot programming should be developed in these directions (Figure 5).

Figure 5. *Development Directions of Robot Programming*



- The number of variations of work pieces is increasing. Besides, a robot in industry 4.0 should be able to treat deviations of prior process steps. So, it is not possible to program each variation in advance. Instead, using sensors, cameras or anything else the actual robot program should be automatically generated (see chapter “*Example: RoboPainter”* for an example).
- Three areas of research are important for HRC. The first area is safety. Mainly there are two safety concepts: Safe robots and safe supervision of the robot cell. But preserving the safety of humans is not the topic of the end user.
  New production concepts for the HRC are the second area of research. The third area is the communication between humans and robots. Many

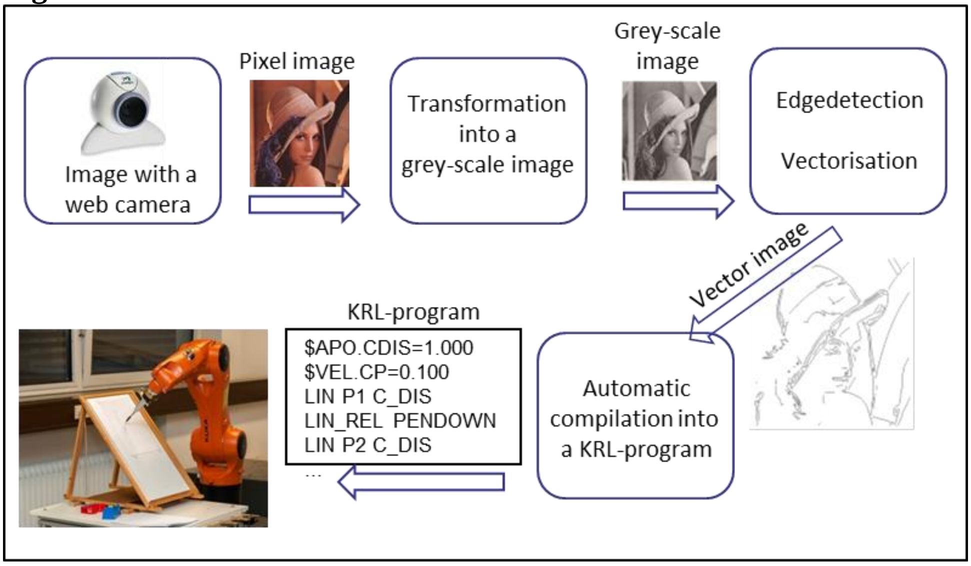
intuitive concepts of programming and operation are possible, from programming by demonstration over augmented reality until multimodal communication with text, voice, gestures and more (see chapter "*Example: Automated Programming of Cooperating Industrial Robots*" for an example of intuitive programming).

- Today, it normally needs much time to build up and program a robot cell. But, it is important to the flexible use of robots, that a robot with some interaction can be installed in a short time (plug and produce). First, it is necessary to standardize the hardware and the electronic connections. For the programming, that means easy and automatic calibration of the positions in the program to the positions in the real cell.

Example: RoboPainter

For the long night of science in Nuremberg, Fürth, and Erlangen the project RoboPainter was developed (Roos and Kunze, 2013) (Figure 6). An image of a visitor is taken by a simple webcam. With different image analysis algorithms, a vector image is calculated out of the light/dark transitions in the image. A KRL (Kuka Robot Language) program is generated and the robot portrays the visitor like a street painter.

Figure 6. *RoboPainter*



More than 100 drawings are painted during the long night. Each portray shows another visitor. So, more than 100 robot programs are generated, executed one time, and thrown away, because the next visitor is another one. This is an example of automated robot programming for batch 1 production.

Example: Automated Programming of Cooperating Industrial Robots

Two cooperating robot arms should apply glue on complex work pieces, by one robot holding the work piece and the other robot guiding the tool. There are two advantages to do this with cooperating robots. First, the orientation between the tool and the work piece could be optimal along the whole path. Second, the process time could be shortened, because the two robots can move against each other.

The common manual programming of such tasks is connected with a lot of effort. One solution to this problem is the automated robot programming (Wagner et al., 2014). Doing so, the programming of the robots should be interactive and intuitive. The end user does not need to program directly in the robot programming language.

Figure 7 shows the whole process for this project. The end user makes a 2-D sketch of the path of the glue application. This sketch is projected onto the 3-D model of the work piece. This model can either be a CAD model or result from a 3-D scan. The result of the projection is the so called 3-D application model. With respect to boundary conditions of the process (e.g. constant vertical guiding of the tool or an optimized utilization of the robot axes), the programs for the two robots are automatically generated. The generated motions should be faster and smoother than with one robot. In addition, the accessibility can be improved.

Figure 7. *Automated Programming of Cooperating Robots (Wagner et al., 2014)*

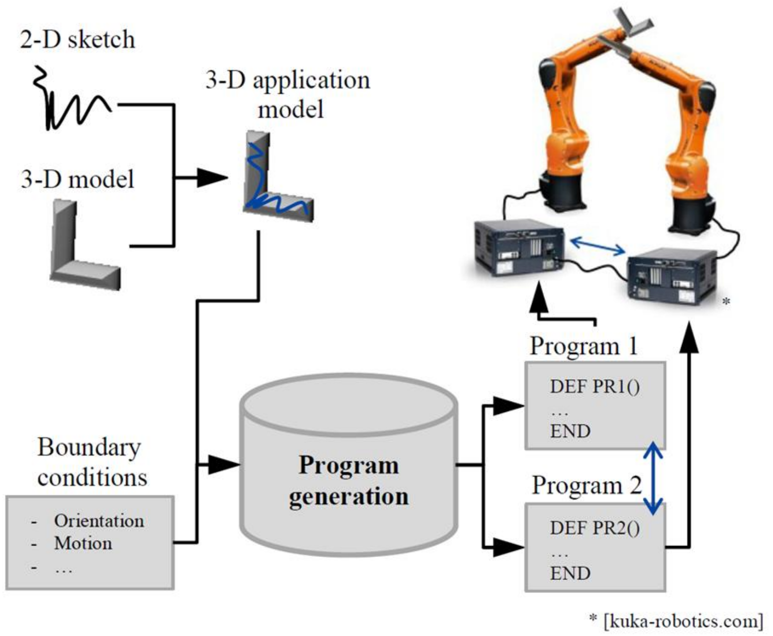


Figure 8. C*ooperating Robots*

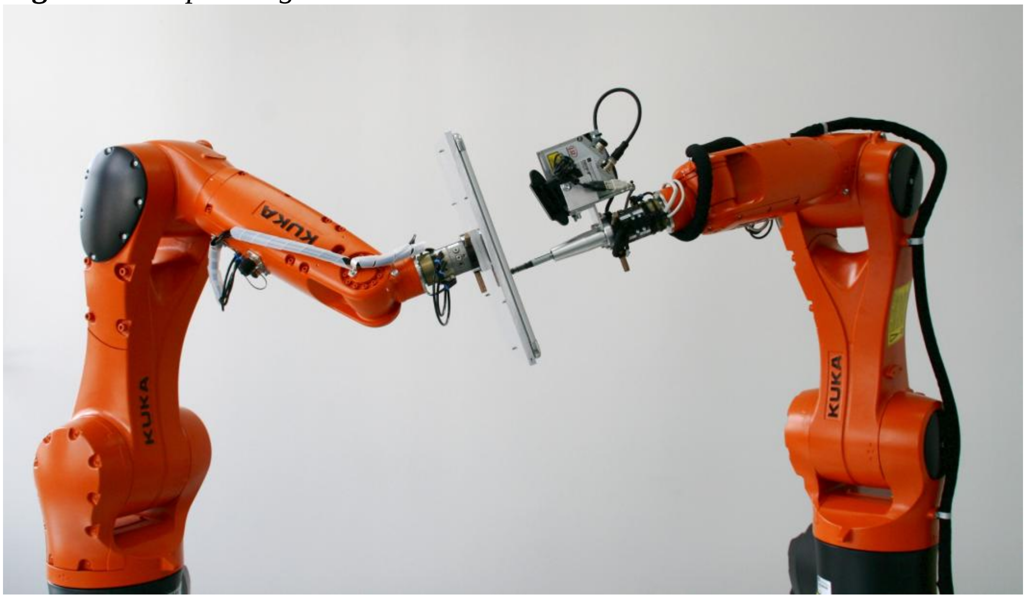


Figure 8 shows the first prototype, at that stage of the project without glue. Both robots move together, but the programs are not automatically generated. This will be the next step in our project. With that automatically generated robot programs in the next step, it is not necessary, that the end user, who realizes the gluing application, to have robot programming knowledge. He only must have knowledge about the gluing process.

Conclusions

Besides robots in mass production, industrial robots will be used more and more as tools and assistants in the production. For this, new concepts of industrial robot programming like intuitive programming are essential. Other main topics are the safety of the worker, the security of the communication, and robot specific learning.

References

Angerer, Andreas; Hoffmann, Alwin; Schierl, Andreas; Vistein, Michael; Reif, Wolfgang: *The Robotics API: An Object-Oriented Framework for Modeling Industrial Robotics Applications*. In: IEEE (Hg.): International Conference on Intelligent Robots and Systems. IROS 2010. Taipei 2010, S. 4036–4041.

Barth, Katharina; Henrich, Dominik: *A GOTO-based Concept for Intuitive Robot Programming*. In: IEEE (Hg.): International Conference on Intelligent Robots and Systems. IROS 2012. Vilamoura, Algarve, Portugal 2012, S. 2338–2345.

Bogh, Simon; Schou, Casper; Rühr, Thomas; Kogan, Yevgen; Dömel, Andreas; Brucker, Manuel et al.: *Integration and Assessment of Multiple Mobile Manipulators in a Real-World Industrial Production Facility*. In: ISR / Robotik

2014. Proceedings for the joint conference of ISR 2014, 45th International Symposium on Robotics and Robotik 2014, 8th German Conference on Robotics. München, 2.-3.6. Berlin [u.a.]: VDE-Verl 2014, S. 305–312.

DKE: *Die Deutsche Normungs-Roadmap Industrie 4.0* [The German Standardization Roadmap Industry 4.0], 2013.

Haag, Sebastian; Pyschny, Nicolas; Brecher, Christian: *Flexible Assembly Robotics for Self-optimizing Production*. In: Sabina Jeschke, Honghai Liu und Daniel Schilberg (Hg.): Intelligent robotics and applications. 4th International Conference, ICIRA 2011, Aachen, Germany, December 6-8, 2011, proceedings, Part II. Berlin ;, New York: Springer 2011, S. 189–198.

Lee, Jay; Bagheri, Behrad, Kao, Hung-An: *Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics*. In Proc. of the Int. Conf. on Industrial Informatics (INDIN) 2014.

Naumann, Martin; Verl, Alexander: *High-level integration of components into a robot cell using semantically annotated state chart descriptions*. In: Robotik 2012 - 7th German Conference on Robotics. Berlin [u.a.]: VDE-Verl 2012, S. 440–445.

Pan, Zengxi; Polden, Joseph; Larkin, Nathan; Van Duin, Stephen; Norrish, John: *Recent Progress on Programming Methods for Industrial Robots*. In: ISR/ROBOTIK 2010. Proceedings for the joint conference of ISR 2010 (41st International Symposium on Robotics) und ROBOTIK 2010 (6th German Conference on Robotics); 7-9 June 2010; parallel to AUTOMATICA. Berlin; Offenbach: VDE-Verl. 2010, S. 619–626.

Roos, Toni; Kunze, Philipp: *RoboPainter*. Projektarbeit. [*RoboPainter*. Project work] Georg-Simon-Ohm Hochschule Nürnberg. 2013

Soller, Katharina; Henrich, Dominik: *Intuitive Robot Programming of Spatial Control Loops with Linear Movements*. In: Torsten Kröger und Friedrich M. Wahl (Hg.): Advances in Robotics Research. Berlin, Heidelberg: Springer Berlin Heidelberg 2009, S. 147–158.

Wagner, Maximilian; Heß, Peter; Reitelshöfer, Sebastian: *Automated programming of cooperating robots*. In: ISR / Robotik 2014. Proceedings for the joint conference of ISR 2014, 45th International Symposium on Robotics and Robotik 2014, 8th German Conference on Robotics. München, 2.-3.6. Berlin [u.a.]: VDE-Verl 2014, S. 129–136.