Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 23 (2014) 188 – 193

www.elsevier.com/locate/procedia

Conference on Assembly Technologies and Systems

Tool oriented Robot Cooperation

Matthias Bartelt[a], Sven Stumm[a], Bernd Kuhlenkötter[a] *

[a]*Institute of Production Systems (IPS), Professorship in Industrial Robotics and Production Automation (IRPA), TU Dortmund University, Leonhard-Euler-Straße 2, 44227 Dortmund, Germany*

[a] *{matthias.bartelt, sven.stumm, bernd.kuhlenkoetter}@tu-dortmund.de*

* Corresponding author. Tel.:+ 49-231-755 5615; fax: +49-231-755 5616.

Abstract

Defining paths for robot tools is mainly limited by workspace of the applied robot, yielding to difficulties in layout design and programming, especially in complex cells. Using the approach presented in this paper, only the tool is considered while programming. Thus, there are no limitations due to robots. The tool itself is mounted to an adapter, which can be linked to multiple robots allowing a coordinated transfer of the tool between them. Hence the system can be extended to a multi-robot system by an arbitrary number of robots, which are automatically positioned within the cell in a subsequent step.

Keywords: Industrial Robot; Tool oriented Programming; Multi-Robot System

1. Introduction

Industrial robots can be used in several automation tasks, like pick and place or assembly. They are very flexible and can easily be adapted to changing conditions. When an industrial robot is used, it is placed into the assembly line with respect to the robot's task and working range. Afterwards appropriate sequences for the robot are created, so it can handle the desired task. Of course, offline programming and simulation tools can assist on locating and instructing the robot. But programming the robot is only a mean to move a tool along some curve. Therefore the concept presented in this paper is to start creating the path of the tool, while the robot is temporarily neglected. After the path is defined, the robot is placed (semi-) automatically, so that the complete path can be reached with the tool. Using this approach yields to some advantages:

- The tool's path is mostly determined by the task. Programming without taking the robot into account is easier, because restrictions caused by the robot (like limited working space) must not be considered.
- The system can be extended to a multi-robot-system, in order to take advantage of such systems without increasing the programming complexity. Thus, advantages of such multi-robot systems, like redundant fixtures or higher relative movement of robot and work object, can be used easily.
- After the path is defined, further requirements for the robot (e.g. required working space or payload) are also defined. Considering this data may result in a better choice of the robot or the robots position. As a result, the planned path is independent of any specific robot manufacturer at programming time.

Nevertheless considering only the tool may yield to some difficulties. Especially it may occur that paths are defined in such a way that no robot can reach all positions, maybe because of a limited working range or because of some structural restrictions. This is mainly caused by a manual choice of the number of robots and their positions. A possible

doi:10.1016/j.procir.2014.10.103

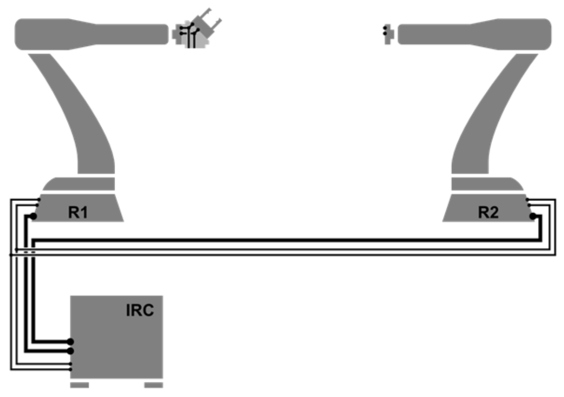

Fig. 1: Two robots (R1 and R2) controlled by one industrial robot controller (IRC). Electrical and pneumatic signals are transmitted via either robot to the tool.

solution to get this step automatized is presented hereinafter, as well as a tool adapter system to support the usage of more than one robot. This adapter is an important part of the concept, in order to simplify programming of multiple robots to solve the desired task.

2. Tool Change System

As described, at some point it may occur, that more than one robot is required. For example this might happen, when there are interfering contours in such way, that they make it impossible for one robot to reach all positions. A use case may be a sealing, which must be applied to a car's roof from the inside. Depending on the size of the sealing, some pillar may prevent the robot from reaching all positions. Stopping at some point, re-positioning from another side in order to finish the original path is far from desirable. Thus some other solution is required, and a possible solution might require the hand-over of the tool from one robot to another. A further use case is a tool, which is required in subsequent processes. Passing the tool between robots may be an appropriate solution, if using two tools is no option.

Automatic tool change systems are available as industry standard components. Using one of these, a robot can attach to a tool mounted in a fixture. Usually due to a pneumatic fixture the tool is mounted to the robot's flange and can be released from the fixture afterwards. Pneumatic as well as electrical signals can be forwarded through those tool change systems. However, all these automated tool change systems require a kind of magazine the tool is placed whenever not connected to a robot, since usually handing-over is not performed directly. In particular a direct hand-over to another robot is not possible just using one tool change system. Thus, a tool can only be used within the workspace of the robot it is mounted to, while the main idea of this paper is to extend those workspaces by merging the individual ones. Whereas indirect hand-over (i.e. releasing the tool at a specific position and picking it with another robot) can only achieve a lose connection of robotic workcells, direct hand-over allows a continuous processing and a non-stop toolpath can be planned without the requirement to consider workspace limitations.

In order to enable a hand-over, an adapter is interposed. It can connect to a tool as well as to a robot. Fig. 1 shows a sketch of the described system. Mainly it is a combination out of three change systems. A tool can be connected from one side, while two positions are available for a robot to connect to. All connection points are oriented with some angle to each other to prevent collisions while handing over. Obviously, electrical and pneumatic signals must be forwarded from the robot to the tool. The challenge is to switch those signals reliable and interruption-free while the system is moved to the next robot. Of course this system can also be used to connect two tools to one robot with the limitation, that a tool change between robots is not possible any more.

3. Use Case

Using the described tool change system will certainly increase the complexity of a given workcell, and for simple tasks, like pick and place tasks where only little program code is required to prompt a robot to perform the task, most likely it won't be used. Anyhow, the more complex a task is the more interesting this concept becomes. Especially when using more robots, programming complexity increases with each robot, while the described approach keeps complexity constant. Within the following section we describe two hypothetical use cases – one describing handling of a workobject and another describing an assembly task, where a cover should be assembled to a frame with the need of a sealing. The main aim of both cases is to illustrate the described concept.

3.1. Use Case 1 – Handling of workobject

As a first use case, we consider an assembly part, which has to pass some surface rework like grinding or polishing before it can be handled in subsequent processes. Fig. 2 illustrates the setup, which mainly consists of a vision system and a rework station. To process a part, it has to be grasped and moved along the vision system with a constant velocity for analyzing the current state of the surface. Afterwards the part is reworked according to the measurement data. To ensure quality, the part will be scanned a second time, which may require another rework, before the part can be moved to the next station.

Of course, this process could be realized with some clamping device and a conveyor system. But especially the return of unsatisfying parts yields to expensive conveyor solutions, while workpiece hand-over creates a flexible flow of work pieces and might establish concepts for a more product variant oriented assembly in automated work cells, like a one-piece flow oriented programming. Furthermore, this approach is more flexible in terms of distribution of tasks to different robots and extending the process chain by additional tasks.

Although a complete simulation of the proposed concept was not done yet, we started to demonstrate proof of concept by defining some specific positions as well as a path for a

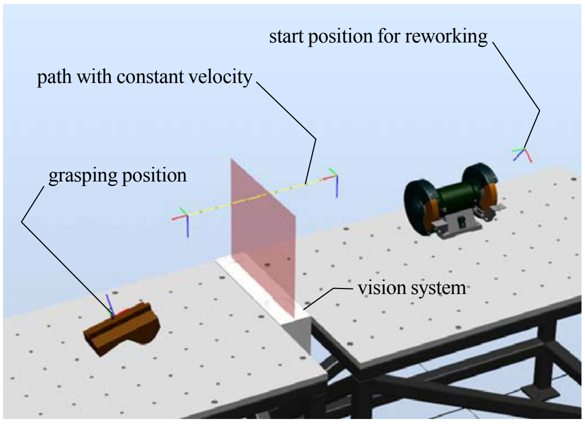

Fig. 2: Schematic setup for a rework station with preceding vision system including a description of a tool path model with process constraints. Only a few adjustments like the grasping position must be done manually.

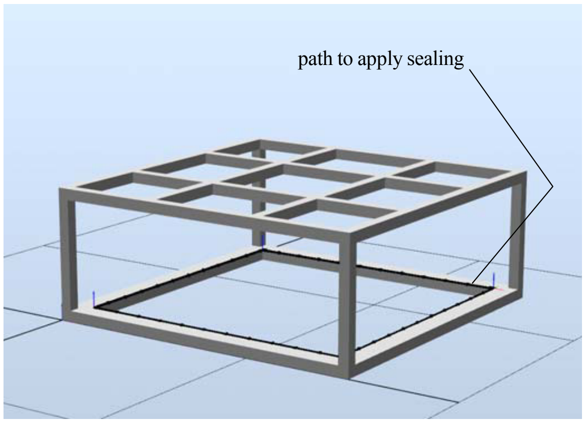

Fig. 3: Visualization of the path of a sealing, which should be applied to a mechanical structure. The path is defined on the inner edge of the bottom of the construction.

given tool (compare Fig. 2). Within a simulation environment we used two robots to perform a measurement of the surface of some workpiece with possible subsequent reworking. Thereby the main focus was to create the possibility to perform the described hand-over, which could be realized by the use of pre-defined program blocks. Nevertheless a programming interface for automated robot path generation based on the required tool path needs to be implemented. With this, a direct analysis of the described concept in comparison to "normal" robot programming can be made.

Next to simulation and programming the proposed concept may also affect accuracy and quality of the performed task, due to the non-symmetrical way the tool is mounted to the robot. Although this information is essential within real setup, measurements with a test bed must be done to obtain those information, while current work focuses on simplification of programming, especially for synchronous multiple robot movements.

3.2. Use Case 2 – Apply a Sealing

The second use case illustrates an assembly task. A cover should be mounted to a welded steel construction. The cover should be mounted on the inner bottom side and can be inserted from one side. Before it is mounted, a sealing should be applied to the construction, as shown in Fig. 3. Due to possible collisions between the robot and the construction, the robot cannot move the required tool along the complete path without interruption. Of course, one solution would be to use some special kinematic, but since the definition of the tool's path is quite simple, the more intuitive solution would be to define the tool path and place the robots afterwards as required.

4. Tool-Centered Programming

Today, there is an increasing demand in product variety combined with low and fluctuating product quantities. In order to handle this, new flexible and innovative solutions are required for a time and cost efficient engineering. A high amount of effort is expended on improving the real production system, while the engineering itself is far from an optimal process. For example, the collaboration between different experts, like designer, electricians or programmers still lacks in inter-process exchange of data, which is of course a basic requirement for collaboration.

Among others, this is a research focus within the conexing project, which is developed at the Institute for Production System (IPS) of the TU Dortmund University. An integrated data infrastructure is created based on a standardized data format, where manufacturers of components may offer their range of products as so called SmartComponents. Those components combine product information and geometrical representation as well as dynamic behavior [1]. Thus, these virtual components can be used to describe manufacturing and assembly components. The included product data and simulative representations is stored using the standard XML based Automation Mark-up Language (AutomationML) [3]. Due to the development kit, which is developed within the research project, different engineering tools can be extended to import and use the described components. In particular, vendors can build a virtual catalog with their components. Those resulting product databases can then be used to build virtual production systems. Having a look at our use cases, we create the above described adapter as SmartComponent. Using the product databases, appropriate components like grippers on the one side or robots at the other side can be selected and considered in the simulation.

After a definition of the available environment we propose a model based approach to industrial robot programming. Employing a specified assembly model allows a generation of detailed process descriptions as described for example in [5, 10, 11]. Especially in assembly a wide variety of process descriptions is available, but mostly not used for model based robot programming. Through linking the tool path planning with semantically task descriptions, we try to assert constraints in the tool path. Using this approach a set of tool-paths can be generated, that describe the solution set for our path planning problem. For a simpler description of the solution concept we will focus on only one specific path.

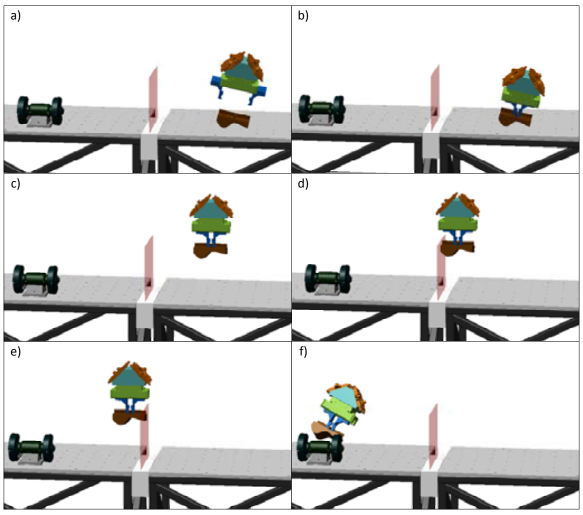

Fig. 4: A possible tool path generated by the described tool path model.

Creating a programming interface which allows for a simple off-line programming of robots based on known work piece information is desirable. Currently programming industrial robots is a task for industrial robot specialist, while programming multi-robot systems requires even more specific training and experience due to the complexities of synchronized robotic movements. By focusing on the tool path, this complexity should be reduced, in order to allow robot experts to easily use multi-robot systems.

Fig. 2 describes a possible model of the path for the first use case described above. Starting with an abstract tool path model, areas are defined, in which for example constant velocities or other conditions are required. These conditions add constraints to the tool path itself, and must be considered when a possible path is defined. Although this step is done manually at present, it will become automated in future work.

This focus on the tool-path as opposed to the planning of a robot path yields to significantly higher degree of flexibility. This degree of flexibility however has to be compensated in order to keep programming complexity low. Using only tool path planning, some elements of standard robot programming are not explicitly covered by the model. In order to generate robot programming certain aspects need to be analyzed, which are described hereinafter:

- Robot configurations used to reach a position
- Collision free robotic movement
- Limited robotic workspace

The problem of defining a robot configuration used to reach a position is a typical problem of robotic path planning and is mostly solved through manual input as well as path specific dependencies and collision avoidance. We therefore propose to (semi-)automatically generate a suggested path allowing a manual definition of configurations for specific or critical positions within the generated path.

A collision free movement can sometimes be achieved using multiple robot using different approach angles. This means, while a position may not be reachable without collision from one robot's position it can be reachable from another robot's position. But especially if obstacles are too large it is possible that no solution can be found. Adding constraints for collision avoidance increases problem complexity significantly. We therefore suggest analyzing possible collision areas in advance and informing the user if these areas are high in complexity or cannot be solved automatically for user feedback and a possible modification of the tool path model.

Concerning the previously addressed problem of limited robotic workspaces, a major aspect of the proposed concept is bypassing these limits. In section 2 we proposed a combination of a tool adapter as well as robotic tool change systems to allow the passing of the tool between robots. In the following section we describe a concept for (semi-) automatic robot placement and determination of tool transfer areas. The passing of object between robots itself can be achieved through a synchronized movement of a multi-robot system. But in order to efficiently use the passing of the tool we propose modelling movements of the tool in a cyclic layout which allow nonstop motion of the tool. Problems concerning failover safety and reliability, which is a concern in nonstop motion will be considered in future work.

5. Analysis of tool paths and solution concept

The above mentioned tool path must be analyzed, in order to determine the required robots as well as their position. Thus, we introduce a concept for systematic analysis of a given tool path. The results are used as a hint for the placement of used robots. At the current state, this must be done manually, but it can become automatized later on. The main benefit within this approach is simple programming of multi-robot systems while ensuring reachability of the complete tool path. Additionally, the quality of the path movement may be improved. A possible process is illustrated in accordance with use case 1. As discussed in the previous section the first step for generating the robot programming is planning of the tool path. As described, we therefore choose a model-based approach to create the tool path. In our case the model-based approach allows for a combined planning of specific tool positions as well as defined movement specifications, while considering usability for a possible user interface. Employing a given tool path model in combination with a given cell layout, results in a number of constraints. These constraints range from a collision free movement to constant tool velocities. Based on the tool path model the planned tool movement can automatically be generated through motion, path and manipulation planning, as shown in Fig. 4. Based on the tool movement the number of necessary robots as well as their positions needs to be determined. This problem shows significant methodical similarities to typical problems in cluster analysis.

The specific problem of determining the number of robots as well as their position based on a given tool path can be described through cluster analysis. In this case the problem consists of determining the robot positions as cluster center while establishing the number of cluster [6]. Additionally two

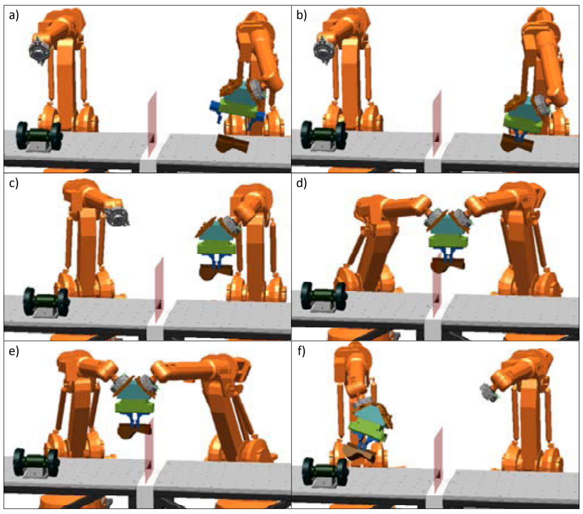

Fig. 5: A possible robot path solution with tool exchange for the first use case described.

constraints must be established: the maximum size of a cluster as determined by the robots workspace and the maximum distance between clusters, because the workspace need to overlap for successful tool transfer between robots. Therefore we propose the use of overlapping clustering algorithms. Especially an existing graph based variant of an overlaying clustering algorithm, as described in [7], is optimized to find a low amount of cluster with minimal overlapping. Other possible algorithms for overlapping clustering range from an Overlapping k-means clustering (OKM) over a Restricted-OKM (R-OKM) to fuzzy c-means clustering [2], and can be differentiated according to cluster fuzziness. A fuzzy based approach can be useful for a smooth definition of overlapping work spaces, and is especially of interest because of diminishing reachability at the edges of robot workspaces.

After the number of robots and their positions are determined, regions within the toolpath used for tool hand-over need to be specified in a subsequent step. This can be done using intersection analysis between two robot workspaces themselves as well as the tool path. Fig. 5 describes a possible solution for the given use case. The transfer of the tool has to be a synchronized movement, in order to ensure a collision free tool transfer as well as a constant and ongoing tool velocity. The tool transfer itself is the crucial aspect of the proposed tool oriented path planning and requires a cooperative synchronized movement of two robots. [9] proposes a force and torque control for robot cooperation in order to minimize stress between robots, which in our case can be employed for a controlled tool transfer procedure to minimize tension taking effect on the adapter. A force and torque monitored approach for tool transfer also allows a fast responding error handling, for example in the case, when a tool transfer is not possible due to misalignment of the multi-robot system.

In most cases of industrial robot cooperation, only a distribution of the workpiece weight is considered [4, 8]. A possible transfer of a work piece between robots can only be implemented through a product-specific complex synchronous movement of both robots. The proposed method and the described adapter facilitate a possible workobject transfer, which is not product specific and can be used for different workobjects. Extending this approach for other applications like adhesive bonding or sealing, as described in the second use case, can create a nonstop tool movement, which could also improve quality especially for larger surfaces.

6. Conclusion

Within this paper we presented an alternative concept for programming of industrial robots, which can be used particularly for multi-robot systems. Moving from the common robot focused programming to tool-oriented programming yields to a separation of the programming of a path, which depends mainly on the tool, and the final robot programming, which is of course vendor specific. As a mean for an easy extension of a robot system by multiple robots we presented a tool change adapter, which allows a hand-over of tools between robots. Thus the tool becomes decoupled from the robot and the user has only to consider the tool on programming, while the resulting implementation for the used robots may be automatized, with the vision of a collaborative working robot collective.

Extending the concept for applications like adhesive bonding or sealing, will also be a focus of future work. A main aspect in this scope is the flow of material, since we currently only considered electrical and pneumatic medium. Furthermore, with a functional programming and simulation system, we need a real setup to finally proof the concept. With this also different measurements, like accuracy and stability within process, will be performed.

Nevertheless we started to implement and evaluate our concept within a simulation system, various points must be substantiated later on, as described above. In Future work we will focus on implementing and optimizing the programming concept as well as a prototypical implementation of the described adapter. Further on, the steps currently done manually must be automatized. Even though the presented use cases (especially the polishing one) may also be realized with tools directly mounted to robots, the proposed concept may yield to a simpler and intuitive robot programming especially for multi-robot systems, which in fact increases flexibility of those systems.

Extending the concept for applications like adhesive bonding or sealing, will also be a focus of future work. A main aspect in this scope is the flow of material, since we currently only considered electrical and pneumatic medium. Finally, the proposed concept introduces an alternative way of programming robots, which may help in future applications.

Acknowledgments

The research and development project “conexing” is funded by the German Federal Ministry of Education and Research (BMBF) within the Framework Concept “Research for Tomorrow’s Production” and managed by the Project

Management Agency Karlsruhe (PTKA). The author is responsible for the contents of this publication.

References

[1] M. Bartelt, A. Schyja, B. Kuhlenkötter, and T. Benkner. Konzept und Möglichkeiten der virtuellen Anlagenplanung. *PRODUCTIVITY Management*, (3):31–33, 2013.

[2] C.-E. Ben N'Cir, G. Cleuziou, and N. Essoussi. Identification of non-disjoint clusters with small and parameterizable overlaps. In *Computer Applications Technology (ICCAT), 2013 International Conference on*, pages 1–6, 2013.

[3] R. Drath. *Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA*. VDI-Buch. Springer, 2010.

[4] Y. Gan, X. Dai, and J. Li. Cooperative path planning and constraints analysis for master-slave constraints analysis for master-slave industrial robots. *International Journal of Advanced Robotic Systems*, 9(88), 2012.

[5] Q. Liwei, Y. Xingguo, W. Hai Peng, and L. Tao. Virtual engineering: challenges and solutions for intuitive offline programming for industrial robot. In *2008 IEEE Conference on Robotics, Automation and Mechatronics*, pages 12–17, 2008.

[6] P. McCullagh and J. Yangy. How many clusters? *Bayesian Analysis*, 3(1):101–120, 2008.

[7] A. Pérez-Suárez, J.A. Martnez-Trinidad, J.A. Carrasco-Ochoa, and J.E. Medina-Pagola. A new overlapping clustering algorithm based on graph theory. In Ildar Batyrshin and Miguel González Mendoza, editors, *Advances in Artificial Intelligence*, volume 7629 of *Lecture Notes in Computer Science*, pages 61–72. Springer Berlin Heidelberg, 2013.

[8] G. Reinhart and S. Zaidan. A generic framework for workpiece-based programming of cooperating industrial robots. In *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*, pages 37–42, 2009.

[9] A. Spiller and A. Verl. Superimposed force/torque-control of cooperating robots. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–7, 2010.

[10] U. Thomas and F.M. Wahl. Assembly planning and task planning – two prerequisites for automated robot programming. In D. Schütz and F.M. Wahl, editors, *Robotic Systems for Handling and Assembly*, volume 67. Springer Tracts in Advanced Robotics, 2010.

[11] F.M. Wahl and U. Thomas. Robot programming – from simple moves to complex robot tasks, 2002.