Available online at www.sciencedirect.com
ScienceDirect
Procedia CIRP 63 (2017) 95 – 100
www.elsevier.com/locate/procedia
The 50th CIRP Conference on Manufacturing Systems

An Integrated Planning and Programming System for Human-Robot-Cooperation

Julia Berg[a*], Gunther Reinhart[a]

[a]Fraunhofer IGCV, Provinostr. 52, 86153 Augsburg, Germany

* Corresponding author. Tel.: +49-821-90678153; fax: +49 821 90678-199. *E-mail address:* Julia.berg@igcv.fraunhofer.de

Abstract

The application of human-robot-cooperation in industrial environments is still in an initial stage. This is caused by safety reasons, the complexity of distributing tasks between the human and the robot as well as the time-consuming programming of the robot. Therefore, a concept for the combination of an automated distribution of the tasks between human and robot and a task-oriented programming of the robot is focused in this paper. The programming system consists of four parts: a task and world model, a planning system for the distribution of the tasks, a programming system to generate the robot program, and an operation module containing an action recognition. After the suggestions for the allocation of the tasks to human and robot conducted by the planning system, the tasks are presented to the user who can approve or adopt them on a human machine interface. According to the determined distribution the robot program will be generated subsequently. This automatically generated program consists of several modules where each module represents an assembly task. In order to detect the assembly task, which is carried out by the human, the movements of the worker's hands and of the object have to be detected by a camera system. In order to interpret the movements of the worker, an approach using Hidden Markov Model is used whereas the Hidden Markov Model is fed with the position data of the hands.

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems

Keywords: human-robot-cooperation; planning; programming

1. Introduction

Mainly manual processes in assembly combined with high wages lead to a demand for automation in order to further allow the production in high-wage countries. In opposition to that the demand for individual products increases which results in a higher amount of product variants combined with shorter product lifecycles. Thus there is a certain need to sustain the flexibility in the assembly process which would decrease in case of fully automated processes. Another aspect that has an impact on the manufacturing industry is the demographic change: the age of the employees increases whereas the assembly processes become more complicated. Therefore the employee needs ergonomic support during his work.

An approach to meet the different demands of automation, flexibility, and the support of aging employees is the human-robot-cooperation, whereas human and robot work together in a shared workspace as shown in Figure 1. Applications of human-robot-cooperation are still rare. This fact is caused by several reasons: one reason is the safety aspects [1]. Typically, some kind of certification process for the human-robot-cooperation has to be passed in order to start working with an application. The second reason which impedes the realization is the planning process which includes the task planning of both resources (human and robot). Heinze states that a lot of expertise is required in order to plan an appropriate human-robot-cooperation [2]. The third problem is the programming of the robot which requires programming expertise that is not necessarily available in small and medium enterprises [3]. This kind of enterprises however is often confronted by small lot sizes with high variants, so that automation is not profitable.

In order to ease the realization of human-robot-cooperation, an approach for an automated programming system is introduced. In this paper a concept for a system is introduced

Peer-review under responsibility of the scientific committee of The 50th CIRP Conference on Manufacturing Systems
doi:10.1016/j.procir.2017.03.318

that will support the planning process of the task division. These tasks will then be transferred into robot code where a task-oriented programming approach is applied. To enhance the human's acceptance towards the robot, action recognition is included in the code so that the robot is able to react to the human's working processes.

Figure 1: Scenario of human-robot-cooperation

2. State of the art

The planning of human-robot-applications has been highlighted in several papers. One of the first approaches was a task planning algorithm presented by Beumelburg [4]. In order to assign the tasks to human and robot, the skills of both are modelled and compared to the requirements of the task. Based on the work of Beumelburg, Heinze et al. set up a methodology for the combined planning and simulation of human-robot-applications which was developed during the project MANUSERV [5]. The aim of the project was the development of a system for human-robot-cooperation solutions based on the manual processes and a database where robot technologies are collected. An important aspect of the work was the description of the processes in a standardized description language [5]. The description of the processes is based on the Methods Time Measurement (MTM) whereas the five basic motions reach, grasp, move, position and release describe assembly processes. Another aspect within this project was the planning of the application based on the Planning Domain Definition Language (PDDL). Next to the planning process, simulation was a focus of the project. With the defined working plan and CAD data of the robot system, the application can be simulated using the simulation software VEROSIM [4]. Another project concerning the planning process of individual assistance systems in assembly processes is called INDIVA. The main goal of this project is the early simulation of human-robot-collaboration and the individual adjustment of the application towards the worker [6]. More approaches concerning the task assignment in human-robot-cooperation are presented in [7, 8, 9].

In order to ease the process of programming robots in general, not necessarily referred to human-robot-application, many approaches have been presented. Two ways to easily program robots are programming by demonstration and task-oriented programming. The aim of these programming techniques is that non-experts in programming are able to program the robot. Orendt gave an example of programing by demonstration through its Tool Center Point (TCP) which is seen as an intuitive possibility for programing [10]. The robot is guided on its TCP and out of the path the program is generated. The research shows that this way of programming is considered as intuitive and can be performed by non-experts [10]. A similar approach is presented by [11].

A different type of intuitive programming designed for non-experts is the task-oriented programming. Task-oriented programming has already been addressed in several works [12]. Approaches were lacking the fast adapting of the task-oriented planning system to changing developments. Therefore [13] expanded earlier approaches to a skill-based model to adapt the task-oriented programming system to different assembly systems. The main focus of his work was the modelling of skills which provide an independent description of the functions of a resource. Furthermore every process is described by the necessary process skills. The description and sequence of the skills is the basis for the generation of the robot and PLC code executed by a postprocessor which was developed by [14]. Other similar approaches for an easier programming of the robot whereas the user tells the robot "what" to do instead of "how to do" it, are focus of the project SMErobotics, or a drag and drop approach presented by [15-17]. In this approach the user defines the robot program by dragging blocks on a user interface. The blocks contain information such as "move to" and can be specified further by the user. More approaches for a simplified programming of robots are presented in [18-20].

In order to increase the acceptance of the collaboration with the robot in human-robot-cooperation, approaches address the anticipation of the robot. Lenz et al. present an approach where the decisions for the next steps of the robot are due to the subsequent steps that need to be performed based on the assembly plan and human actions. A data base with information about the assembly plan, tools and products supports the decision process [21]. Another approach in this context was presented by Faber et al. where the human-robot-cooperation in assembly tasks had to be supported by cognitive architecture. In this approach the planning of the tasks of the robot was performed online. The basis for the planning process is the CAD data of the product which serves as a basis for the derivation of the assembly sequence. With this assembly sequence, the costs for possible assembly solutions were took into account and calculated related to the participant. The path with the lowest costs is taken. Information about the decisions of the system are visualized on a human machine interface [22]. Other approaches use Bayesian models to estimate the workers intention to adopt the robot to the worker's behavior [23,24].

In the last two presented approaches, the autonomy of the robot system increases. However studies show that the efficiency of a worker decreases when the autonomy and speed of the robot increase [25].

Although many approaches for an easier programming of the robot have been presented, they mostly do not include the human's needs such as an adaption to the human's speed. In order to ease the programming process, but also to meet the human's requirements in collaboration, an approach for an automated programing system with integrated action recognition is introduced in this paper. The following passage describes the system architecture of the approach. After giving an overview of the architecture, each module is described.

3. System architecture

The system consists of four modules which are the task and world model, the planning module, a programming module and an operation module. The main purpose of each module is described in the following sections. Figure 2 gives an overview of the system architecture.

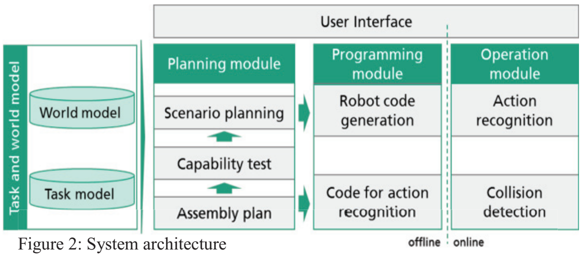

Figure 2: System architecture

The system can be divided into an online and offline part. The task and world model, the planning module and the programming model operate offline. The task and world models represent the product structure and workspace in a CAD-environment. The task model is the part of the CAD-model which shall be assembled in the considered step . With input from the task and world model, the planning module is responsible for the derivation of an assembly sequence and the task description. After this a capability test is conducted to analyze which resource (human or robot or both resources) is capable for executing a certain task. Subsequently, the human-robot-scenario is developed due to command variables such as lead time, or maximum automation due to ergonomic aspects. Having a solution for the scenario, the user will be included into the process: the scenario will be shown on a user interface where the worker can review the proposed scenario and can either agree or decline. After an agreement, the scenario will be sent to the programming module which then will generate the robot code for the robot tasks. The programming module consists of two parts. One part is responsible for the path planning of the robot and the other one is for the integration of the action recognition. A more detailed description of the programming module is given in the corresponding section. The operation module in which the action recognition takes place runs online during operation. Its main task is to evaluate the workers movements and to identify his actions. According to the executed actions the module changes the value of the variables in the robot code. The functions of the modules are presented in the following sections.

3.1. Task and World Model

The basis for the simulation within the program generation for the robot is the product and the world represented by a CAD-model. In order to decrease the complexity of the task planner in the planning module, the CAD-model of the product should be limited to the necessary parts that are supposed to be assembled in the considered steps. The world should also be represented in a CAD-world. and contains the modelling of the working space [26]. In this approach, the world model describes the layout of the working space if it is fixed, the available robot, the gripper, and other components such as feeding systems. If the layout of the scene is already fixed, the layout definition will be taken into account in the planning module when the tasks are assigned to human or robot.

3.2. Planning module

As previously mentioned the purpose of the planning module is the generation of the human-robot-cooperation scenario. This section mainly describes the purpose of the planning module which is not yet fully implemented. The main tasks of the module are the derivation of the tasks out of the CAD-model and the description of the tasks, the alignment of requirements and skills and, furthermore the assignment of the tasks to human and robot. The input for the planning module is the task and world model. For the planning module, important information taken from the world model are conditions such as the setup of the workspace in case it cannot be changed for the human-robot-cooperation. Resources describe available technical equipment such as robots, grippers or feeding technologies. The available robot is described with properties such as payload, range, degrees of freedom, etc. This information is used to describe the skills of the robot for a further capability test. The task model describes the product in a CAD-environment and provides the basis for the derivation of the assembly sequence combined with the task description in textual form. The assembly plan and the task description are the basis for the capability test, the programming and also the action recognition described later on. The task description includes information about the corresponding product such as weight, measurements, or form stability. and process properties. To ensure the safety of the worker in the human-robot-cooperation some safety specific information about the product are necessary: this includes sharp edges or peaking parts on the product. Using this information, the capability test can be performed which results in a profile where the tasks and the possible resource are listed. A first approach for the capability test is a flow chart with yes- or no-conditions that decides whether the task can be performed by human, robot or human and robot.

If a task can be performed by human and robot, the possible resource is still both. The assignment of the tasks is performed in the next step. In order to assign tasks to human and robot assembly steps, such as “screw casting top” must be divided into smaller sections. According to Lotter the assembly process for big parts can be divided into seven tasks where each task includes some of the basic movements from MTM [27]. This description will be adapted to the following steps: haulage of the component, haulage of the supporting assembly parts such as screws, and the assembly process itself where an assembly process can also be a checking process. The assembly step can now be assigned to human or robot. In order to allocate the tasks, the user can choose between three targets: maximization of parallel tasks, maximization of automated tasks and least mean time. According to the chosen target, a scenario for the allocation is generated. The assignment of the tasks gives one specific solution for the setup of the human-robot-cooperation for the chosen task. In order to include the worker in the process, the task assignment is presented on a user interface by blocks on a time bar which is similar to the representation of the tasks in a Gantt Chart presented in [7] The worker now has the possibility to agree with the assigned tasks or to arrange the tasks by moving the blocks on the screen. . After the agreement the tasks will be transferred into robot code in the programming module which is described in the following section.

3.3. Programming module

Within the programming module the code for a robot is generated. It is divided into two parts: program code according to [26] and variable definition for the action recognition. The programming module described by Weck & Brecher consists of an action planner, a gripping planner, a path planner and an assembly planner. The main task of the action planner is to divide the tasks into smaller units which can be transferred into robot program code. If the task is a gripping task, the gripping planner is responsible for the detailed planning of the best gripping position in order to ensure a safe transport and no damage to the part. The choice of the gripper is also task of the gripping planner. The focus of the path planner is mainly the evaluation of a possible path without any collision with any objects in the robot surroundings [26].

The inputs for the programming module are the tasks of human and robot in a certain description. The task description is conducted by the planning module, but still on a basis which is not detailed enough to generate robot program code. As described in the action planner by [26], the tasks have to be subdivided into fractions which can be assigned to gripper, path and assembly planner. Each task will be transferred into one program module, so that the program will consist of several modules. Backhaus describes a post processor which conducts several steps to generate the robot code [14]. The post processor uses a task description model with six levels. It starts from a primary process level, where the assembly process and the starting state and end state of the product are described. The last level is the code level where the program code is generated [16]. The presented approach will lean on the concept of Backhaus. The robot code will neither contain any waiting times nor waiting for a signal from a button in order to start the robot. The modules will be started by the action recognition. The action recognition itself is described in the following section, here only the variables for the action recognition that are integrated in the code, are introduced.

In order to realize the action recognition during operation, the code modules have to be prepared. Therefore variables will be generated. A module to fulfil a task by the robot can only be started when the requirements according to the assembly priority plan are met. This means that the basis for the generation and assignment of the variables is the assembly plan of the product. The idea is to set up the modules with if-conditions. For example if task 2 can only be started after task 1 has finished, the module for task 2 would start with the condition: *If task_1 = True Then…*

During the assignment of the tasks, task 1 will be assigned to either human or robot. If the task is assigned to the robot, the change of the variable *task_1* will be conducted in the module for task 1:

If task 1 however is conducted by a human, the variable has to be changed by the action recognition. The robot program itself runs on the robot control; the analysis of the action recognition conducts a PC online . When the execution of task 1 by the worker is identified by the system, the variable *task_1* will be changed to *True* which means that all requirements for task 2 are fulfilled and the module can be started.

According to [25], a big issue for the worker in human-robot-cooperation is unpredictable robot movements [25]. To help the worker to get used to the assembly together with the robot, the sequence of the robot tasks will be defined once and not changed. However when there is a change in sequence caused by the worker which results in a change of tasks on the tasklist of the robot, the necessary task can be conducted if the requirements according to the assembly plan are met.

3.4. Operation module

The operation module consists of the two parts action recognition and collision detection. Both elements are described in detail within the next two sections.

3.4.1. Action Recognition

As described above, the generated robot code shall include a basis for an action recognition performed during operation. The action recognition aims to identify tasks performed by the worker, so that a slight adaption towards the workers behavior can be fulfilled. At first, the basics of the action recognition based on Hidden Markov Models are explained which is followed by the adoption of this model for an assembly work place.

The basics for Hidden Markov Models have already been presented by Rabiner [28]. A Hidden Markov Model (HMM) is a special case of a Bayesian Network. Sequences of certain events can be detected without knowing their real states. The basis for HMM is the Markov-Chain. The possibility to change to a different state only depends on the actual state and the next state itself. In a HMM the actual states are not visible, rather they are hidden (cf. Figure 3). However every state sends out emissions that can be detected. The emission matrix describes the probability to detect a certain emission in a certain state [28].

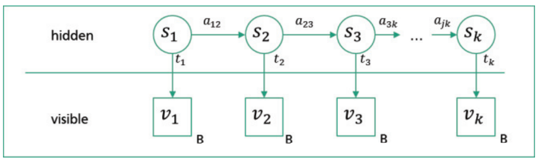

Figure 3: Hidden Markov Model (cf. [28])

In order to increase the accuracy and robustness of an HMM, several HMM can be connected on several levels to build up a layered hidden markov model (LHMM) [29]. A layered hidden markov model consists of at least two layers whereas one is the top-layer model and one is a layer with several models. Layered hidden markov models have been issue of research in action recognition. [29] uses LHMM to identify the workers actions in an assembly line [30]. One layer represents the subtasks of the worker which are combined to a task in the above layer. In order to apply this LHMM on the assembly line, a big set of trainings had to be fulfilled. In order to integrate the action recognition using HMM to this task-oriented programming system, the training effort of the model has to be reduced. The presented approach applies the HMM to an assembly workspace and gives an approach to reduce the training. The LHMM consists of two layers whereas in the top-layer the assembly tasks are identified. The bottom layer contains of two HMM, one for the right and one for the left hand. The input for the bottom layer is the position of right or left hand. Figure 4 shows the setup of the layers for the considered structure. For the identification of the actions, the assembly step is divided into the five basic motions according to MTM. To identify the assembly step, the reach and move motion must be identified. If the hand position of either the right or left hand moves to an object which is assigned to a task, there is a certain possibility that this certain task is being executed. To estimate the probability that a hand moves to an object, a linear slope from the point where the assembly takes place to the objects is defined. If the hand mostly moves on this line, the probability for the reach movement is high. This identification takes place in the bottom layer as seen in Figure 4.

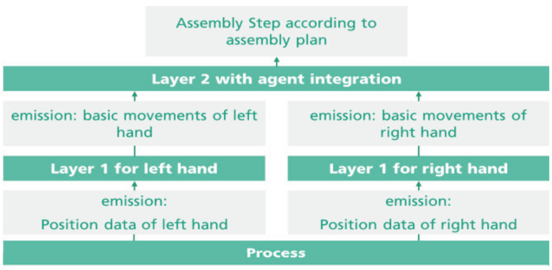

Figure 4: Structure of the layered Hidden Markov Model

The positions of objects to corresponding tasks have to be given to the system by camera detection.

In order to receive the position of the hands, a Leap Motion Controller is used. Coming from the consumers industry, the Leap Motion Controller is a device to detect the hand positions and gestures. Studies of the Leap Motion Controller show a high accuracy respective to the hand positions and processing time [31]. First studies of the action recognition also show that the Leap Motion Controller is an appropriate device to fulfill the requirements. The disadvantage of the usage of one Leap Motion Controller is the working space which is compared to the assembly space to small.

3.4.2. Collision detection

In order to enhance the acceptance and the safety of the worker in a close collaboration with the robot, collision detection has been set up. Using the Leap Motion Controller which detects the hand positions for the action recognition, collision detection between the hands and the TCP of the robot has been developed. To detect possible detections, the coordinates of the hand middle and the coordinate of the TCP are compared. The hands are modelled by balls with the position of the hand as center. The arms of the human are modelled as cuboids. The robot is also modelled by a ball with the TCP coordinate as center as shown in Figure 5. If the ball of a hand and the ball representing the robot overlap, a collision is detected which results in a stop of the robot. The robot will pause its program as long as the collision is detected. Figure 5 shows the visualization of the collision detection. In this case no collision is detected, because there is no overlapping of the robot and the hands.

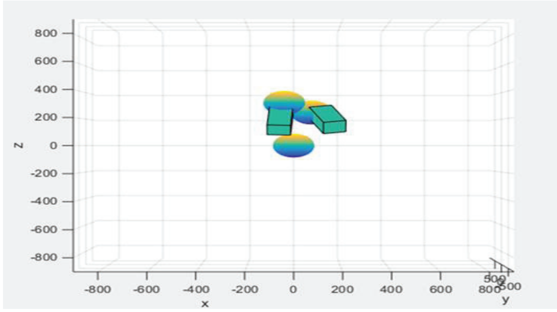

Figure 5: Visualization of collision detection

4. Conclusion and future work

In this paper an approach for a task-oriented programming system has been introduced. The programming system consists of four modules which are: the task and world model, the planning module, a programming module and an operation module. The planning module's main task is the assignment of the tasks to the resources human and robot and the planning of the application. The programming module is responsible for the code generation which includes the preparations for the action recognition performed in the operation module. Within the operation module, an action recognition based on layered Hidden Markov Models is executed. The action recognition identifies tasks executed by the worker, so that the robot system can slightly adopt to the worker's behavior, for example by starting its task in the right moment without getting a signal from a button push. Regarding the action recognition and collision detection, the system has been implemented and tested with a Leap Motion Controllers. To expand the detected space, the approach will be enhanced by a Kinect. Thus the safety spaced can be expanded and bigger parts of the body can be detected to feed the model for action recognition.

The presented approach is concept for an architecture for an automated programming system for the human-robot-cooperation. The programming module will now be implemented. This includes the development and design of the human machine interface where the tasks will be presented to the user. Furthermore it is planned that more tasks can be added on the interface in case tasks are missing within the process. Within the project some scenarios for a human-robot-cooperation will be identified. Through the implementation of the presented system in these applications, the system will be evaluated.

Acknowledgements

This research was conducted within the project Rokoko (02P15A001) funded by the German Federal Ministry of Education and Research (BMBF). The authors would like to thank the BMBF for their financial support.

References

[1] M. Faber, J. Bützler und C. M. Schlick, "Human-robot Cooperation in Future Production Systems: Analysis of Requirements for Designing an Ergonomic Work System," Procedia Manufacturing, 3. Jg., S. 510–517, 2015.
[2] F. Heinze, M. Klöckner, N. Wantia, J. Rossmann, B. Kuhlenkötter und J. Deuse, "Combining Planning and Simulation to Create Human Robot Cooperative Processes with Industrial Service Robots," AMM, 840. Jg., S. 91–98, 2016.
[3] R. D. Schraft und C. Meyer, "The need for an intuitive teaching method for small and medium enterprises," in Robotik 2006, S. 95–105.
[4] K. Beumelburg, "Fähigkeitsorientierte Montageablaufplanung in der direkten Mensch-Roboter-Kooperation," Universität Stuttgart, 2005.
[5] J. Deuse, J. Roßmann, B. Kuhlenkötter, A. Hengstebeck, O. Stern und M. Klöckner, "A Methodology for the Planning and Implementation of Service Robotics in Industrial Work Processes," Procedia CIRP, 23. Jg., S. 41–46, 2014.
[6] C. Thomas, L. Stankiewicz, A. Grötsch, S. Wischniewski, J. Deuse und B. Kuhlenkötter, "Intuitive Work Assistance by Reciprocal Human-robot Interaction in the Subject Area of Direct Human-robot Collaboration," Procedia CIRP, 44. Jg., S. 275–280, 2016.
[7] P. Tsarouchi, S. Makris und G. Chryssolouris, "On a Human and Dual-arm Robot Task Planning Method," Procedia CIRP, 57. Jg., S. 551–555, 2016.
[8] P. Tsarouchi, S. Makris, G. Michalos, A.-S. Matthaiakis, X. Chatzigeorgiou, A. Athanasatos, M. Stefos, P. Aivaliotis und G. Chryssolouris, "ROS Based Coordination of Human Robot Cooperative Assembly Tasks-An Industrial Case Study," Procedia CIRP, 37. Jg., S. 254–259, 2015.
[9] R. Müller, M. Vette und O. Mailahn, "Process-oriented Task Assignment for Assembly Processes with Human-robot Interaction," Procedia CIRP, 44. Jg., S. 210–215, 2016.
[10] E. M. Orendt, M. Fichtner und D. Henrich, "Robot programming by non-experts: Intuitiveness and robustness of One-Shot robot programming," in 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN): IEEE, 2016, S. 192–199.
[11] C. Meyer, R. Hollmann, C. Parlitz und M. Hägele, "Programmieren durch Vormachen für Assistenzsysteme – Schweiß- und Klebebahnen intuitiv programmieren (Programming by Demonstration for Assistive Systems – Intuitive Programming of Welding and Gluing Trajectories)," it - Information Technology, 49. Jg., Nr. 4, 2007.
[12] R. Humburger, Konzeption eines Systems zur aufgabenorientierten Roboterprogrammierung. Zugl.: Aachen, Techn. Hochsch., Diss., 1997. Aachen: Shaker, 1998.
[13] Julian Christoph Sebastian Backhaus, Adaptierbares aufgabenorientiertes Programmiersystem für Montagesysteme. München: Herbert Utz Verlag, 2016.
[14] M. Ehrmann, Beitrag zur Effizienzsteigerung bei der Programmierung flexibler, roboterbasierter Montagezellen: Konzeption und Realisierung eines nutzergerechten Programmiersystems. Kaiserslautern: Techn. Univ., Pak, 2007.
[15] Thomas Dietz, Ulrich Schneider und Marc Barho, "Programming System for Efficient Use of Industrial Robots for Deburring in SME Environments," in Robotics ; Proceedings of ROBOTIK 2012 ; 7th German Conference on, [Berlin]: [VDE Verlag], 2012.
[16] C. Brecher, B. Schröter, C. d. O. Almeida, F. Dai, B. Matthias und S. Kock, "Intuitiv bedienbare Programmiersysteme zur effizienten Programmierung von Handhabungsaufgaben," in VDI-Berichte, Bd. 1841, Robotik 2004: Leistungsstand, Anwendungen, Visionen, Trends : Tagung München, 17. und 18. Juni 2004 : Deutschland Robotik 2004, Düsseldorf: VDI-Verl., 2004, S. 303–310.
[17] Martin Naumann, Matthias Bengel und Alexander Verl, "Automatic Generation of Robot Applications Using a Knowledge Integration Framework," in Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK): Date, 7-9 June 2010, [Berlin]: [VDE Verlag], 2010, S. 631–638.
[18] T. Nagai, S. Aramaki und I. Nagasawa, "Representation and Programming for a Robotic Assembly Task Using an Assembly Structure," in 7th IEEE International Conference on Computer and Information Technology (CIT 2007), S. 909–914.
[19] Nikhil Somani, Markus Rickert, Andre K. Gaschler, Caixia Cai, Alexander Clifford Perzylo und Alois Knoll, "Task Level Robot Programming Using Prioritized Non-Linear Inequality Constraints,"
[20] T. Arai, H. Izawa und Y.Maeda, Proceedings of the 2003 IEEE International Symposium on Assembly and Task Planning (ISATP2003): From the assembly and disassembly of manufactured products to the design and manufacturing of micromachines : July 10-11, 2003, Besançon, France. Piscataway, N.J.: IEEE, 2003.
[21] C. Lenz, S. Nair, M. Rickert, A. Knoll, W. Rosel, J. Gast, A. Bannat und F. Wallhoff, "Joint-action for humans and industrial robots for assembly tasks," in The 17th IEEE International Symposium on Robot and Human Interactive Communication, 2008: RO-MAN 2008 ; 1 - 3 Aug. 2008, Technische Universität München, Munich, Germany, Piscataway, NJ: IEEE, 2008, S. 130–135.
[22] C. Schlick, M. Faber, S. Kuz und J. Bützler, "Erweiterung einer kognitiven Architektur zur Unterstützung der Mensch-Roboter-Kooperation in der Montage," in Schriftenreihe der Hochschulgruppe für Arbeits- und Betriebsorganisation e.V. (HAB), Industrie 4.0: Wie intelligente Vernetzung und kognitive Systeme unsere Arbeit verändern, W. Kersten, Ed., Berlin: Gito, 2014, S. 239–263.
[23] Oliver C. Schrempf und Uwe D. Hanebeck, "A Generic Model for Estimating User-Intentions in Human-Robot Cooperation," Proceedings of the 2 nd International Conference on Informatics in Control, Automation and Robotics, ICINCO 05, 2005.
[24] R. Kelley, A. Tavakkoli, C. King, M. Nicolescu, M. Nicolescu und G. Bebis, "Understanding human intentions via hidden markov models in autonomous mobile robots," in Proceedings of the 3rd ACMIEEE international conference on Human robot interaction, New York, NY: ACM, 2008, S. 367.
[25] Koppenborg, Auswirkung von Autonomie und Geschwindigkeit in der virtuellen Mensch-Roboter-Kollaboration. Verfügbar unter: http://www.dguv.de/medien/ifa/de/pub/grl/pdf/2013_103.pdf.
[26] M. Weck und C. Brecher, Werkzeugmaschinen 4: Automatisierung von Maschinen und Anlagen, 6. Aufl. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2006.
[27] B. Lotter und H.-P. Wiendahl, Montage in der indusriellen Produktion: Ein Handbuch für die Praxis. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg, 2006.
[28] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE, 77. Jg., Nr. 2, S. 257–286, 1989.
[29] S. Perdikis, D. Tzovaras und M. G. Strintzis, "Recognition of Human Actions using Layered Hidden Markov Models," 2008.
[30] Yi Shen, System für die Mensch-Roboter-Koexistenz in der Fließmontage. München: Herbert Utz Verlag, 2015.
[31] Eduardo S Silva, Anderson O De Abreu, Henrique P De Almeida, Veronica Teichrieb und Geber L Ramalho, "A Preliminary Evaluation of the Leap Motion Sensor as Controller of New Digital Musical Instruments," 2013.