Task-based programming and sequence planning for human-robot collaborative assembly

Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 52-13 (2019) 1638–1643

Task-based Programming and Sequence Planning for Human-Robot Collaborative Assembly

Khurshid Aliev, Dario Antonelli, Giulia Bruno

Politecnico di Torino, Corso Duca degli Abruzzi 24, Turin, 10129 Italy
(e-mail: dario.antonelli@polito.it).

Abstract: The paper proposes a task-based programming method to support assembly processes in a human-robot collaborative workcell. The method for assembly sequence generation is implemented as a robot program homebuilt by composing pre-programmed elementary working steps. Developed system can support human operator as an interactive tool for collaboration with robot. Robot program and developed software are used to interface the robot with the human during the execution of the assembly tasks starting from single mechanical parts. The proposed Human Robot Interaction System (HRIS) performs an assembly job with the contemporary collaborative work of robots and humans that share tasks in the same workspace. The system is tested on a case study made on purpose and constituted by several different flanges mounted on a support in a variety of configurations. Thus, the method is general and can be easily adapted to a multiplicity of assembly tasks.

Keywords: Robotics in manufacturing, Human-Automation Integration, Production Control, Control Systems

1. INTRODUCTION

Human-Robot collaboration (HRC) has become an expanding trend in industry since the emergence of Industry 4.0 (I4.0).

The industrial applications of collaborative robots are frequently related with assembly or assembly-related tasks, like handling, pick and place, welding, gluing, joint quality testing. The majority of assembly applications are found in the automotive world, see Michalos et al. (2010). Due to the high volume of production, in the automotive field, the assembly process is carefully designed and optimized well before the production phase.

Present study deals with the application of HRC to assembly of small series of products belonging to sometimes numerous product families. It is a common case in other industrial sectors apart from the automotive one: aerospace, machine tool manufacturing, electronic. The problem to be faced is to allow the HRC assembly of a family of products for which there isn't a predefined assembly sequence, due to the number of variants. It is the case of machine tools production, where every client requires some customization to the basic machine tool configuration, adding or excluding some features during the assembly. The objective of the study is to find an automatic way to define the assembly sequence and produce the corresponding robot program for whichever part variant of the same product.

The solution to the problem is found by exploiting already existing and effective methods developed for automatic assembly optimization, by introducing the additional constraint that some tasks could be executed either by robot, or by human, or by a collaborative effort of both. Assignment of tasks to robots and humans alike is a side problem that recently received interest. At the level of cell layout definition it is analysed by Michalos et al. (2018). At the level of workload assignment the problem is formalized by Ding et al. (2014) and solutions can be found in the studies of Bänziger et al. (2018) or Bruno and Antonelli (2018). By adopting a convenient representation of the product and of the assembly tasks (Tan, 2009) it has been possible to automatically build the assembly sequence for all the possible variants of the same product family. In this way the robot program can be executed as a composition of elementary tasks that have been programmed in advance, with the additional help of modern manual guidance programming (Massa et al., 2015). The novelty in the proposed solution is to embed in the assembly sequence planning the constraints introduced by collaboration between human and robot and in the change of problem objective, from time minimization to quality of collaborative work.

The proposed method has been verified and validated on a case study, developed ad hoc in order to constitute a benchmark for subsequent extensions

2. RELATED WORKS

Different ways to implement HRC in a factory are thoroughly described by numerous surveys, like in Goodrich (2007) or in Villani (2018). In particular, factory applications are driven by the necessity to assure the higher safety standards during works that are often inherently dangerous. There is a general agreement on classification of collaborative modes in industry based on the safety levels of interaction: safety-rated

10.1016/j.ifacol.2019.11.435

monitored stop, hand guiding, speed and separation monitoring, power and force limiting (following ISO-TS 15066). Another important issue in industrial HRC is the human robot interface that should be intuitive and easy to program, even for inexperienced robot programmers. The state of the art is described in Pan et al. (2012). For sake of simplicity, it is possible to assume that robot programming is evolving from trajectory based to task-based programs, where the programmer should only be concerned about the goal of task execution and not about the details of execution.

Kruger et al. (2009) show a comprehensive survey of possible assembly processes where HRC can lead to benefits both for the quality of the process and for the quality of the work performed by human worker. Examples are: axle sequencing, cardboard blank handling, catalytic converter test cell, engine block or transmission handling, floor pan or front sub-frame transfer. Assembly processes are usually classified based on the sharing either of space or of time between human workers and robots. Safety-rated monitored stop prevents every form of sharing, while power and force limiting mode allows both the time and space sharing. An example of application in a modern factory can be seen in the work of Tsarouchi et al. (2017).

The representation of assembly sequences is a longtime field of study that has a first practical and effective solution in the AND/OR graph proposed by de Mello and Sanderson (1986). Subsequent works tried to solve the problem in a more computer-oriented way, in order to allow automatic optimization of the assembly sequence, as shown by Thomas et al. (2018).
Unfortunately, in the field of formal representation of the assembly sequence for mechanical products, there isn't an overall agreement on a standard formulation. For the study of human-robot collaboration assembling, all the formalization methods proposed in the review of Bahubalendruni (2015) has been considered. The assembly sequence representation methods easier to automate are: Sanderson et al. (1990); Sinanoğlu and Rıza Börklü (2005); Gottipolu and Ghosh (2003). The last work proposes a notation that is simple to apply automatically, starting from the CAD model of the assembled product, without the need of manual additional work. For this reason, it has been adopted in present study.
The assembly sequence representation is a list of parts, assembly order, direction of movement or joining constraints. The executor, either human or robot, in this phase is not significant.
During the collaborative work it is important to efficiently manage the communications between human and robot and other kind of exchanges of tool, parts or other. This is accomplished through an 'ad hoc' built robot interaction system

3. HUMAN ROBOT INTERACTION SYSTEM

The collaboration between humans and robots for the execution of the assembly job requires the management of interactions and communications between two different agents. In this paper the task is managed by a communication interface that is called Human Robot Interaction System (HRIS). The flowchart of the HRIS is shown in Fig.1. Actual implementation of HRIS is based on Matlab GUI that constitutes the interface between the human operator and the robot.
At the beginning of the process, the human operator runs the HRIS. Subsequently, a robot preprogrammed script is executed on the robot control unit. The connection between HRIS and the robot is actuated by TCP/IP protocol.

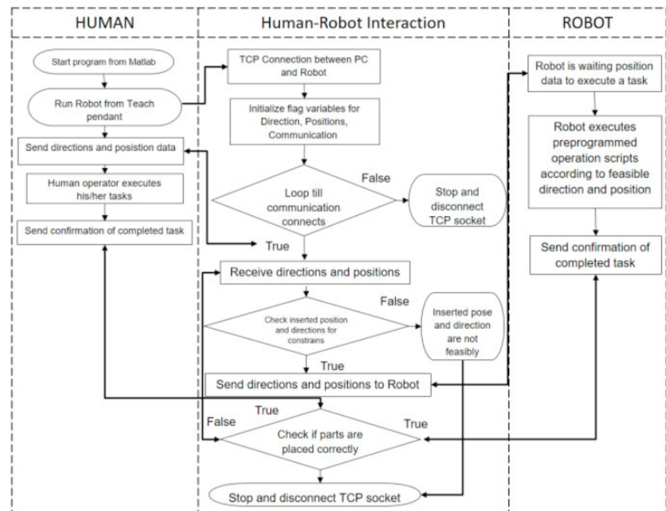

Fig. 1. The flowchart of the HRIS.

HRIS has functions for communication, sending commands to execute tasks, checking for constraints and other functions to execute directional (horizontal/vertical) operations. The system initializes flag variables of communication, directions and positions and it goes to communication loop. GUI of the system and command lines guides human operator during the assembly execution.
Human operator receives text messages about chosen direction, position and type of assembly objects in sequence and program checks selected options for constraints. Human operator sends a command to execute the task and waits for robot until it has finished its task. After successful assembling each task, program remembers which parts are assembled. According to this information, HRIS makes decision on assembling parts by checking placed flanges for correctness (software remembers every task which has been performed with collaboration and informs operator about all processes). If parts are assembled correctly human and robot agrees that assembling is completed successfully and the system disconnects TCP socket.
Next section describes the exploitation of HRIS in an assembly use case.

4. ASSEMBLY COLLABORATION

4.1 Assembly job formalization

An assembly job (J) is the assembly of a set of parts each of which in a specific position with respect to the others. A complex assembly job can be decomposed in a hierarchical tree of sub-assembly (Makris et al., 2014). A job can be decomposed in a set of tasks (T) to be executed to accomplish the final goal, e.g., assembly two parts together, or pick a part and move it to a specific position. Tasks in turn are composed by an ordered set of operations (O). An operation is a basic building block, which can be directly programmed to be executed by a robot (e.g., move tool to a specific position, grasp an object, etc..) and by a human (e.g., grasp and placing object, screw bolt and nuts, etc..).

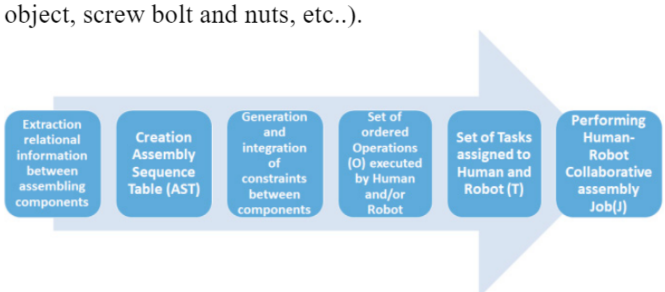

Fig. 2. Human-robot collaboration assembly process steps.

In Fig. 2, human robot collaborative assembling process is presented in six steps. In order to choose an optimal assembly sequence, steps 2 generates all the possible assembly sequences. Among them the feasible sequences should be extracted in step 3 by applying assembly constraints from step 1, thus creating the solution space. The optimal sequence is found in the solution space by implicit complete enumeration (step 5), using as a criterion the collaboration performance index proposed by Bruno and Antonelli (2018).

.

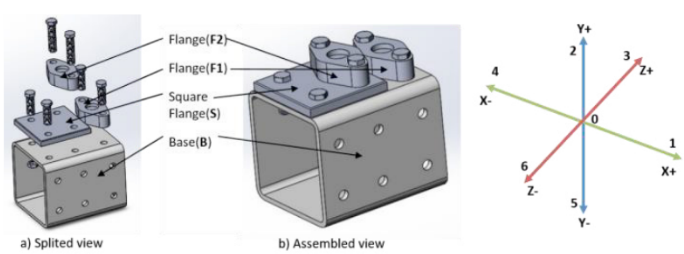

Fig. 3. Human-robot collaboration assembly case study.

Fig. 3 shows the assembled and the exploded view of assembling CAD model of a case study assembly. The part is made up of different components, i.e., a base (B) on which three flanges (F1, F2, S) are mounted and joined by screwed bolts. The part is assembled in a collaborative cell that includes a UR3 robot with OnRobot RG2 haptic gripper and a human operator that communicate through HRIS. The case study has been designed to provide different variants in the assembly sequence and in the assembly directions. The method for sequence generation was initially proposed by (Gottipolu and Ghosh, 2003) for the general representation of assembly sequences. Different types of constraints are extracted from the CAD model and two uni-directional matrices are created, the contact function (C-function) and the translational function (T-function). The C-function provides contact conditions of pairs, while T-function indicates the existence or absence of collision-free path. C-functions demonstrate the local assembly feasibility while T-functions define the global feasibility in the directions ±X, ±Y and ±Z (represented as 1to 6 axes, see Fig. 3). Afterwards, all feasible assembly sequences from these two functions are generated. The sequences are then represented as a table of assembly states and assembly tasks starting from individual components in unassembled state to finished assembly.

In order to identify geometrical feasibility and collision free assembly parts C (Table 1) and T (Table 2) functions of assembly pairs are created. Furthermore, we exploit C and T function to simplify the robot programming. Instead of programming a complete assembly sequence, we separately programmed each operation, and then combined them in the form of tasks.

Table 1. Representation of C function for flange assembly

Pairs	Touch contact function(C-function)					
	C_1	C_2	C_3	C_4	C_5	C_6
(B,S)	1	1	1	0	0	0
(B,F_1)	1	1	1	0	0	0
(B,F_2)	0	0	0	0	0	0
(S,F_1)	0	0	0	0	0	0
(S,F_2)	1	1	1	0	0	0
(F_1,F_2)	0	0	0	0	0	0
(S,B)	0	0	0	1	1	1
(F_1,B)	0	0	0	1	1	1
(F_2,B)	0	0	0	0	0	0
(F_1,S)	0	0	0	0	0	0
(F_2,S)	0	0	0	1	1	1
(F_2,F_1)	0	0	0	0	0	0

Table 2. Representation of T function for flange assembly

Pairs	Translational motion function (T-function)					
	T_1	T_2	T_3	T_4	T_5	T_6
(B,S)	1	1	1	1	0	1
(B,F_1)	1	1	1	1	0	0
(B,F_2)	1	1	0	1	0	0
(S,F_1)	1	1	1	1	0	0
(S,F_2)	1	1	0	1	0	0
(F_1,F_2)	1	1	1	1	0	0
(S,B)	1	1	1	1	0	1
(F_1,B)	1	1	1	1	0	0
(F_2,B)	1	1	1	1	0	0
(F_1,S)	1	1	1	1	0	0
(F_2,S)	1	1	1	1	0	0
(F_2,F_1)	1	1	1	1	0	0

In this way, previous programs are reused to speed up the process. Having four assembly components, there are 12 pairs of sub-assemblies. After applying contact (C) and precedence (T) functions, assembly sequence table (AST) shown in Table 3 represents total number of possible assembly states which is 9 and of assembly tasks which is 15.

Table 3. HRC sequences after C and T functions.

Level	Assembly States	Assembly Tasks
I	{B},{S},{F1},{F2}	-----
II	{B, S} {B, F1} {S, F2}	{(B), (S)} {(B), (F1)} {(S), (F2)}

	{S, B} {F1, B} {F2, S}	{(S), (B)} {(F1), (B)} {(F2), (S)}
III	{B, S, F1}	{(B, S), (F1)}; {(B, F1), (S)} {(S, B), (F1)}; {(F1, B), (S)}
VI	{B, S, F1, F2}	{(B, F1, S), (F2)}; {(B, S, F1), (F2)}; {(B, S, F2), (F1)}; {(S, B, F1), (F2)}; {(F1, B, S), (F2)};

Topological constraints are applied to remaining assemblies.

Constraint 1 - assembly is not possible: if there is no placed base(B) first and other flanges (S, F1, F2) on the top of the base(B).

For this reason, we delete from Table 3 {(F1), (B)}, {(S), (B)} and {(F2), (S)} assembly states from the second level, {(S, B), (F1)}; {(F1, B), (S)} assembly tasks from the third level and assembly tasks from fourth level {(S, B, F1), (F2)}; {(F1, B, S), (F2)} The remaining states and tasks are represented in Table 4.

Table 4. HRC sequences after topological constraints.

Level	Assembly States	Assembly Tasks
I	{B},{S},{F1},{F2}	-----
II	{B, S} {B, F1} {S, F2}	{(B), (S)} {(B), (F1)} {(S), (F2)}
III	{B, S, F1}	{(B, S), (F1)}; {(B, F1), (S)}
VI	{B, S, F1, F2}	{(B, F1, S), (F2)}; {(B, S, F1), (F2)}; {(B, S, F2), (F1)};

Then functional constraints are applied.

Constraint 2 – it is not allowed for robot gripper to execute the task: place flange (F1) before square flange (S).

Applying constraint 2, enough space must be left to release the flange using robot gripper. After applying constraint 2 into remained AST, level VI {(B, F1, S), (F2)} is considered as not feasible. The remaining states are in Table 5.

Table 5. HRC sequences after functional constraints.

Level	Assembly States	Assembly Tasks
I	{B},{S},{F1},{F2}	-----
II	{B, S} {B, F1} {S, F2}	{(B), (S)} {(B), (F1)} {(S), (F2)}
III	{B, S, F1}	{(B, S), (F1)}; {(B, F1), (S)}
VI	{B, S, F1, F2}	{(B, S, F1), (F2)}; {(B, S, F2), (F1)};

Constraint 3 is about stability of assembling states {B, S, F1, F2} of level VI.

Constraint 3 - assembling states are stable: if assembling parts are assembled in sequence order.

Components must be placed taking account of stability of assembling parts. As a result, the assembly sequence left is {(B) – (S) – (F1) – (F2)} and the optimal AST is shown in Table 6.

Table 6. HRC sequences after stability constraints.

Level	Assembly States	Assembly Tasks
I	{B},{S},{F1},{F2}	-----
II	{B, S} {B, F1} {S, F2}	{(B), (S)} {(B), (F1)} {(S), (F2)}
III	{B, S, F1}	{(B, S), (F1)}; {(B, F1), (S)}
VI	{B, S, F1, F2}	{(B, S, F1), (F2)};

4.2 HRIS execution

Tasks assigned to human includes placing base and flanges on the reference positions and screwing operations. To execute collaborative tasks, human operator sends commands to robot through HRIS.

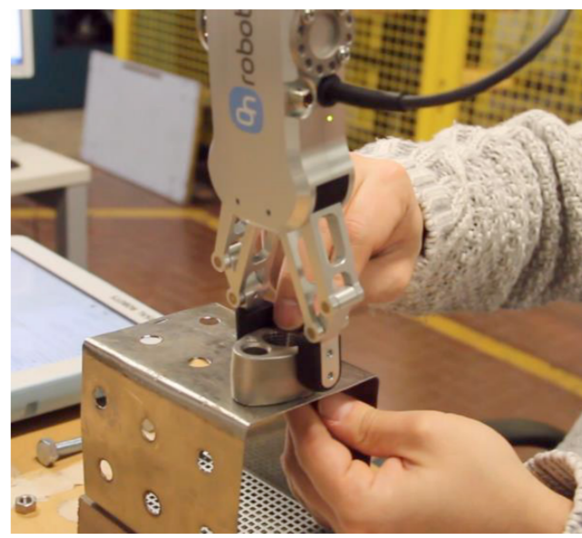

Fig. 4. Collaborative joining. Human and robot share the same workspace at the same time.

The execution of the task often requires the collaboration of human and robot, as shown in Fig.4. Robot keeps the flange steady and human join the flange to the base, exploiting her/his superior dexterity. In Fig. 5 the assembly system collaboration diagram is shown, which highlights all the interactions required to execute the collaborative task.
In human operator column, number of tasks to execute are shown and human operator intuitively makes choice of task commands according to the type of flange and position on the base. According to selected task number by human operator, robot understands which script is selected to perform assembly. Before executing a task, human and system check the selected task for constraints (constraints doesn't pass: if position is already filled with flanges, there is no flange on home position or flange is geometrically no feasible etc..). If there are no constraints, robot performs the task with human operator, otherwise the software informs about constraints and human operator decides to continue or to relax the constraints. In this way, human and robot perform task sequences in collaboration and do the assembly job.

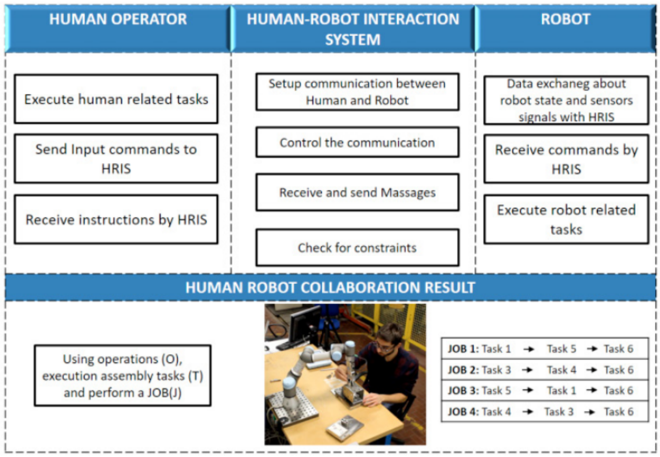

Fig. 5. Human-Robot Assembly collaboration diagram.

For instance, to perform one assembly task, assembling square flange(S) to base(B) in human robot collaboration chosen from the optimal assembly sequence generated for our case study: First human operator places base (B) and square flange (S) to reference positions after installing TCP/IP communication between HRIS and UR3 teach pendant, human operator presses button named horizontal(H) to choose direction and button named “4” which means to place square flange (S) to base (B).

If there is no constraints robot performs its task using operation: grasp square flange(S) and move to base(B) when robot finishes its task human operator takes bolts and nuts and execute the his/her tasks. In this sequence human and robot perform other tasks, in order to complete optimal assembly sequence and perform a complete job. For this case study, JOB 3: TASK5 - TASK1 - TASK6 is the optimal assembly job for human-robot collaboration assembly (Fig. 6). The system is dynamic and flexible which means we can add extra assembly parts and assembly tasks to perform in different directions.

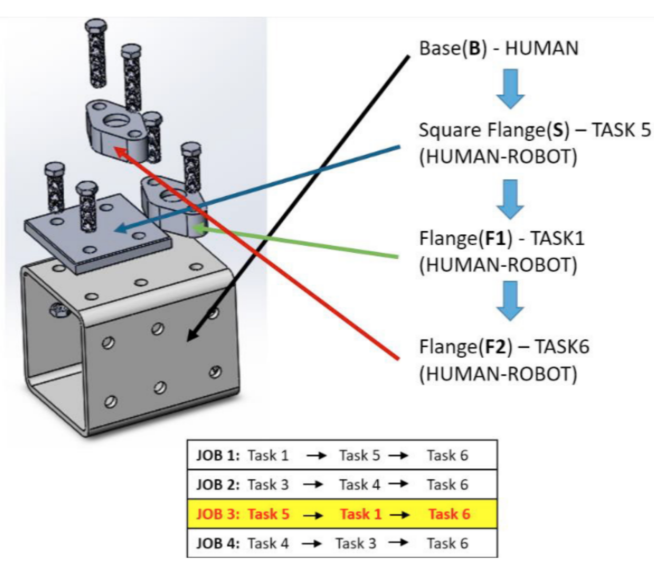

Fig. 6. Assembly sequence order after optimization.

4.3 GUI of HRIS.

Fig.7 shows the human robot collaboration with the developed GUI. A user-friendly application is useful for new unskilled operators. Usage of the application is very easy, first operator establishes communication by pressing Connect button after he/she chooses assembling direction and press predefined position number for particular flange types. By pressing sequence of number of position and types of flanges operator will be able to assemble the flanges on the base. After every operation the application guides human operator for the next operation using message boxes. Each time operator performs assembling task the application records tasks and checks for constraints. Moreover, application has emergency button which stops in emergency cases and TCP/IP disconnecting button.

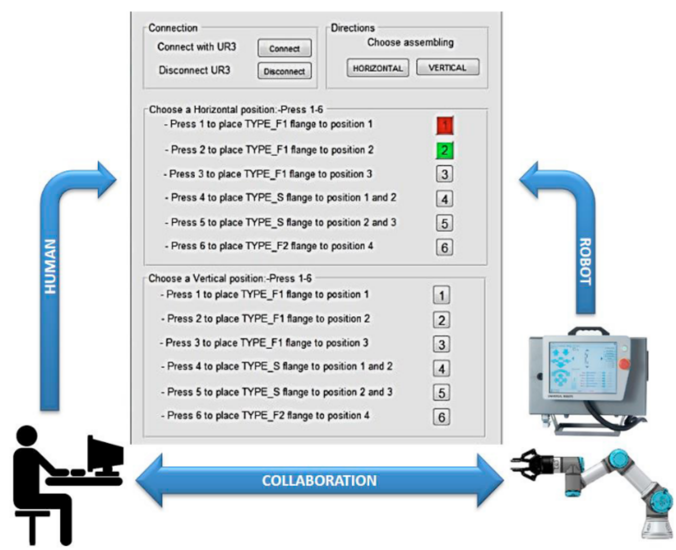

Fig. 7. GUI interaction with human and robot.

The safety in the collaboration area is another important factor that is managed by allowing an additional emergency button on the HRIS. This additional safety level is redundant with respect to built-in robot protections that guarantee human safety during the collaborative work. It is introduced specifically for allowing the operator to abort incorrect processes and avoid damaging the workpiece or the tools.

5. CONCLUSIONS

Optimization of assembly sequence is implemented for an industrial-like case study with a human operator and a robot that execute the process collaboratively.

The more important aspect of the study is that assembly sequence is not pre-optimized in the design phase of the process, but it is decided every time during the execution of work. As a consequence, there is not a robot program loaded in advance on the controller, but the program is generated in real time by composing a set of task-based subprograms after the assembly sequence is found. To find the assembly sequence an AST is generated from the CAD model of the part and a number of constraints are applied, deriving them from the same CAD model. Task assignment procedure, described in another study, allows to define which kind of collaboration to adopt for every assembly task. To demonstrate the concept with an experiment, a human robot interaction system has been developed, which compose human using her/his own brain, software developed on Matlab IDE running on an external PC and UR3 robot program on URScript running on the robot controller. The application of constraints leads to find a feasible assembly sequence in human robot collaboration.

Task-based programming of robot and optimal assembly sequence design are complex problems that hardly allow a definitive and complete solution universally applicable and foster the application of evolved artificial intelligence. Thus, several industrial assembly processes have strong similarities allowing for simplified but effective solutions.

REFERENCES

Antonelli D., Astanin S., Bruno G. (2016) Applicability of human-robot collaboration to small batch production. IFIP Advances in Information and Communication Technology, 480, 24-32.

Bahubalendruni, M. V. A., Biswal, B. B., & Khanolkar, G. R. (2015). A review on graphical assembly sequence representation methods and their advancements. Journal of Mechatronics and Automation, 1(2), 16-26.

Bänziger, T., Kunz, A., Wegener, K. (2018) Optimizing human–robot task allocation using a simulation tool based on standardized work descriptions. Journal of Intelligent Manufacturing, 1-14.

Bruno, G., Antonelli, D. (2018). Dynamic task classification and assignment for the management of human-robot collaborative teams in workcells. The International Journal of Advanced Manufacturing Technology, 98(9-12), 2415-2427.

Homem de Mello, L.S.,Sanderson, A.C. (1986). AND/OR Graph representation of assembly plans. Tech. Report, Robotics Institute, Carnegie Mellon University.

Ding, H., Schipper, M., Bjoern, M. (2014) Optimized task distribution for industrial assembly in mixed human robot environments case study on IO module assembly, IEEE International Conference on Automation Science and Engineering

Goodrich, M.A., Schultz, A.C. (2007) Human-robot interaction: a survey, Found Trends Hum-Comput Interact, 1, 20375.

Gottipolu, R.B. and Ghosh, K. (2003) A simplified and efficient representation for evaluation and selection of assembly sequences. Computers in Industry, 50(3), pp.251-264.

Hsu, Y.Y., Tai, P.H., Wang, M.W. and Chen, W.C. (2011) A knowledge-based engineering system for assembly sequence planning. The International Journal of Advanced Manufacturing Technology, 55(5-8), pp.763-782.

Krüger, J., Lien, T. K., Verl, A. (2009) Cooperation of human and machines in assembly lines. CIRP Annals-Manufacturing Technology, 58(2), 628-646.

Makris, S., Tsarouchi, P., Surdilovic, D. and Krüger, J. (2014) Intuitive dual arm robot programming for assembly operations. CIRP Annals of Manufacturing Technology 63(1), pp.13-16.

Massa, D., Callegari, M., Cristalli, C. (2015) Manual guidance for industrial robot programming. Industrial Robot: An International Journal, 42(5), 457-465.

Michalos, G., Makris, S., Papakostas, N., Mourtzis, D., Chryssolouris, G. (2010) Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach. CIRP Journal of Manufacturing Science and Technology, 2(2), 81-91.

Michalos, G., Spiliotopoulos, J., Makris, S., Chryssolouris, G. (2018) A method for planning human robot shared tasks. CIRP Journal of Manufacturing Science and Technology, 22, 76-90.

Mosemann, H. and Wahl, F.M. (2001) Automatic decomposition of planned assembly sequences into skill primitives. IEEE transactions on Robotics and Automation, 17(5), pp.709-718.

Pan, Z., Polden, J., Larkin, N., Van Duin, S., & Norrish, J. (2010) Recent progress on programming methods for industrial robots. In Robotics, 41st International Symposium on Robotics (ROBOTIK) pp. 1-8. VDE.

Sinanoğlu, C. and Rıza Börklü, H. (2005) An assembly sequence-planning system for mechanical parts using neural network. Assembly Automation, 25(1), pp.38-52.

Sanderson, A.C., de Mello, L.S.H. and Zhang, H., 1990. Assembly sequence planning. AI Magazine, 11(1), p.62.

Tan, J. T. C., Duan, F., Zhang, Y., Arai, T. (2009) Extending Task Analysis in HTA to Model Man-Machine Collaboration in Cell Production, Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics Bangkok, Thailand

Thomas, G., Chien, M., Tamar, A., Ojea, J. A., Abbeel, P. (2018) Learning Robotic Assembly from CAD. arXiv preprint arXiv:1803.07635.

Tsarouchi, P., Matthaiakis, A.S., Makris, S. and Chryssolouris, G. (2017) On a human-robot collaboration in an assembly cell. International Journal of Computer Integrated Manufacturing, 30(6), pp.580-589.

Villani, V., Pini, F., Leali, F., & Secchi, C. (2018) Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Mechatronics.