Journal of Intelligent Manufacturing (2020) 31:1635–1648
https://doi.org/10.1007/s10845-018-1411-1

Optimizing human–robot task allocation using a simulation tool based on standardized work descriptions

Timo Bänziger[1] · Andreas Kunz[2] · Konrad Wegener[2]

Received: 29 May 2017 / Accepted: 6 March 2018 / Published online: 3 April 2018

Abstract

Human–robot collaboration is enabled by the digitization of production and has become a key technology for the factory of the future. It combines the strengths of both the human worker and the assistant robot and allows the implementation of an varying degree of automation in workplaces in order to meet the increasing demand of flexibility of manufacturing systems. Intelligent planning and control algorithms are needed for the organization of the work in hybrid teams of humans and robots. This paper introduces an approach to use standardized work description for automated procedure generation of mobile assistant robots. A simulation tool is developed that implements the procedure model and is therefore capable of calculating different objective parameters like production time or ergonomics during a production cycle as a function of the human–robot task allocation. The simulation is validated with an existing workplace in an assembly line at the Volkswagen plant in Wolfsburg, Germany. Furthermore, a new method is presented to optimize the task allocation in human–robot teams for a given workplace, using the simulation as fitness function in a genetic algorithm. The advantage of this new approach is the possibility to evaluate different distributions of the tasks, while considering the dynamics of the interaction between the worker and the robot in their shared workplace. Using the presented approach for a given workplace, an optimized human–robot task allocation is found, in which the tasks are allocated in an intelligent and comprehensible way.

Keywords Human–robot collaboration · Simulation · Task allocation · Optimization

Introduction

Market changes towards individualization of products are highly affecting production in the automotive industry. Workers are facing increasing complexity due to more variants with fewer equal parts being assembled on the same line (Koren 2010). At the same time the available amount of work force from the market is decreasing due to demographic changes. In order to compensate for these developments, assistance technologies and automation solutions have to be implemented in assembly lines, where the majority of the work is currently done manually as stated by Krüger et al. (2009). Human–robot collaboration is a new work concept that promises substantial improvements, for example in efficiency and ergonomics for the worker as shown by Angerer et al. (2012). In the fourth industrial revolution, the digitization of production enables the use of new technologies, connected sensors and learning algorithms as proposed by Evans and Annunziata (2012). Kusiak (2017) states that smart manufacturing must embrace big data and that current software for controlling production processes and resource planning will face challenges in a more dynamic and open manufacturing environment. In the factory of the future, mobile robotic assistant systems will collaborate with human workers as predicted by Teiwes et al. (2016). Hybrid teams of humans and robots that share the same workspace at the same time allow an individual degree of automation in each workplace. Therefore, robots need to be flexible and cannot be programmed by hand each time anymore. Job orders and control commands have to be generated locally, based on information from the production database.

✉ Timo Bänziger
timo.baenziger@volkswagen.de

Andreas Kunz
kunz@iwf.mavt.ethz.ch

Konrad Wegener
wegener@iwf.mavt.ethz.ch

1 Volkswagen AG, Smart Production Lab, Wolfsburg, Germany

2 Institute of Machine Tools and Manufacturing, ETH Zürich, Zurich, Switzerland

First industrial applications of basic human–robot collaboration are already implemented, for example at the Volkswagen plant in Wolfsburg (Volkswagen 2016). In these pilot applications, the main incentive is the implementation of safe human–robot interaction and the achievement of acceptance of this new technology (de Gea Fernández et al. 2017b). The implementation of intelligent assistant robots is of high complexity since different sensors, actors, and interfaces have to be integrated. Therefore, new methodologies for the implementation of mobile robots in adaptive manufacturing environments are needed, as concluded by Nielsen et al. (2017). With online planning and real-time decision making of robots, the distribution and allocation of tasks has to be performed in an intelligent way, as concluded by de Gea Fernández et al. (2017a). While maximizing objectives like time, costs, and quality, constraints like capabilities of the worker and the robot have to be considered. Furthermore, the workspace has to be modeled accurately and also social aspects, like preferences of workers, have to be taken into account.

In this work, a method to model the assembly procedure and a new approach to find an optimized human–robot work distribution for a given workplace in an automotive assembly line is presented. The procedure model is based on the information from the work description, the shopfloor layout, and the product database. It can therefore be used for the automated generation of instructions for human workers or commands for robots. A simulation tool is developed to implement the proposed procedure model. The advantage of the new tool is the ability to simulate different task allocations of the worker and the robot for a given workplace, based solely on the procedure model. Dependencies like collision avoidance movements between the human worker and the robot are taken into account. Additionally, the impact of path finding in different environments with containers and the use of tools and parts is simulated, which is a significant improvement in accuracy when compared to approaches, found in existing literature. After validating the simulation with an existing workplace at the Volkswagen plant, it is used to optimize the task allocation in a human–robot scenario. Therefore, a new approach is presented in which the human–robot task allocation is optimized using genetic algorithms. In this new approach, the optimization is not carried out using a simple fitness function, but with respect to a fitness value that is calculated in the developed simulation. In contrast to existing optimization methods for the human–robot task allocation, the method presented in this paper is able to consider the highly dynamic interaction between the worker and the robot in the same workspace. Using the presented approach, an optimized human–robot task allocation is found for an existing workplace at the Volkswagen plant in Wolfsburg.

The sections in this paper are structured as follows. In the state of the art, an overview on human–robot collaboration is given and related work is discussed. In the "Methodology" section, the functionality of the simulation is explained in detail, and in the "Experiment" section the simulation is validated. In the "Optimization" section, an approach for an intelligent human–robot task distribution is presented and analyzed in the "Result" section. Finally, a conclusion and an outlook on future work is given.

State of the art

The most important aspect in order to assess the potential of human–robot collaboration is the safety for the human worker as stated by Haddadin et al. (2011). The procedure to guarantee the safe operation for a human–robot shared workspace is regulated in several standards. According to the A-standard 'DIN EN ISO 12100' (German Organization for Standardization 2011), a risk analysis and risk minimization concept is needed for each human–robot shared workplace. In the 'IEC 61508' (International Electrotechnical Commission 2010), the functional safety is defined and further specified for different technologies in the B-standards 'DIN EN ISO 13849' (German Organization for Standardization 2008) and 'DIN EN ISO 13855' (German Organization for Standardization 2010). The C-standard 'DIN EN ISO 10218' (German Organization for Standardization 2012) defines four modes for safe human–robot interaction:

– Safe stop: no movement of the robot while a human is in the same workplace.
– Hand guiding: the robot moves only if it is directly guided by a human.
– Speed and distance surveillance: depending on the distance between the human and the robot, the moving speed of the robot is decreased to a safe value defined in the risk minimization concept.
– Force and power limitation: either by construction or by control, an injury of a human by the robot is excluded.

In the new 'ISO TS 15066' (International Organization for Standardization 2016) these concepts are described and biomechanical limits and measurement regulations are defined for the implementation of human–robot shared workplaces. Applications that implement one of the first three modes as safety strategy can already be found in industry. The inherent safety by force and power limiting, based on biomechanical limits and measurements, is still a field of research.

The required safety mode also depends on the kind of human–robot interaction. For industrial applications, human–robot interaction can be divided in four concepts depending on the shared workspace and time, as shown in Table 1. There is no interaction between locally and chrono-

Table 1 Classification of human–robot interaction

Interaction	Locally separated	Locally identical
Chronologically separated	No interaction	Cooperation
Chronologically identical	Coexistence	Collaboration

logically separated humans and robots. If locally separated, but operating at the same time in a workstation, human and robot coexist, e.g. when they work on different sides of a vehicle like shown by Müller et al. (2014). In this case, a safety stop strategy is sufficient, because the workspace of human and robot do normally not overlap at any time. Cooperation is defined as performing different tasks to achieve a common goal, e.g. the transport and handling of a part by a robot and the following assembly of the part by the worker. In this concept, human and robot share their workspace, but not at the same time. This allows the implementation of speed and distance surveillance safety strategies. An example of human–robot cooperation is shown by Morioka and Sakakibara (2010). The highest degree of interaction is the collaboration, defined as the work on a common task to achieve a common goal. In this concept, humans and robots share the same workspace at the same time, which means a direct physical interaction is taking place. Therefore, either hand guiding or force and power limiting safety strategies must be implemented in order to achieve an efficient collaboration, like presented by Michalos et al. (2014) or Cherubini et al. (2016). It can be seen that these concepts define the possible degree of interaction and assistance between workers and robots and require different work distribution strategies.

Another important aspect in human–robot interaction is the description of interaction between partners in a team. Task allocation between multiple robots (MRTA: Multi-Robot-Task-Allocation) is a well-researched topic and presented optimization approaches can be adapted for human–robot teams. Modeling the interaction between robots is already a complex task. As proved by Dahl et al. (2009), significant performance effects from group dynamics in multi-robot task allocation result in a NP-complete problem. They present a vacancy chain scheduling model as a formal model of a restricted multi-robot task allocation problem. By dividing global system performance into individual robot contributions, the general scheduling problem in MRTA is simplified. Nagarajan and Thondiyath (2014) also developed a heuristic algorithm for cooperative task allocation in constrained multiple robot systems. A peer search method is used to optimize the task allocation between heterogeneous robots based on a cost function including execution, communication, and static costs. The results are tested in a simulation where interdependent sub-tasks with requirements are allocated to a number of robots with different capabilities. Instead of using requirements and capabilities, Shi et al. (2010) presented an approach to optimize MRTA using reputations gained from the evaluation of the completion of historical tasks. Based on a combination of the reputations between multiple robots and an overall reputation of robots and groups of robots, single tasks are allocated in an optimized way. Das et al. (2015) introduce a more flexible task allocation algorithm that uses parallel auction and execution. Depending on resources needed for certain operations, robots place bids on a number of tasks based on their hardware and software specification. It can be seen that for multi-robot task allocation, numerous different approaches have been researched. The general advantage is that robots follow predefined movements, which results in a good predictability of robotic behaviors. Therefore, accurate models can be developed to find well optimized multi-robot task allocations. Task scheduling for single robots can also be efficiently optimized using kinematic models combined with genetic algorithms, which is similar to the traveling salesman problem as shown by Zacharia and Aspragathos (2005). Loredo-Flores et al. (2008) additionally utilized a set of cameras to automate the detection of welding spots for the robot.

The distribution of tasks in human–robot teams (HRTA: Human–Robot-Task-Allocation) brings significantly greater complexity to a workplace as stated by Shen et al. (2015). It is clear that in human–robot teams the skills differ heavily between the worker and the robot compared to multi-human or multi-robot teams. The major difference is that in human–robot teams the worker always has the primacy over the robot, which is not the case if only robots or humans work in the same workspace. This means that the robot always has to react to the behavior of the worker, even if this results in a lower overall efficiency, which is usually not the case in multi-human or multi-robot teams. Shen et al. (2015) give a detailed overview on design principles for workspace-sharing concepts including task identification and coordination aspects for assembly systems. The difficulty for the robot to determine the active goal of the worker is shown. Since humans perform tasks in different non-deterministic ways, a robust task model and a detailed database are identified as requirements for successful task coordination. Also, Hengstebeck (2015) state that automation tasks need to be modeled using relevant parameters concerning product, process and work system environment. They propose a formal modeling framework for the description of manual work processes as base for the application of industrial assistance robotics. The modeling language is based on MTM-1 and offers a detailed description of assembly tasks for human–robot collaboration. However, detailed information of single tasks is usually not available for automotive assembly processes and must first be collected. Stenmark and Malec (2014) presented an algorithm to describe constraint-based assembly tasks in unstructured natural language, which brings improvements to the convenience and speed in programming human–robot applications. They state that the problem is the unavailability of easily parametrizable robotic skills, which shows the need for more detailed task description models. Ghosh and Helander (1986) state that the detailed analysis of manufacturing tasks and the development of an intelligent task model are crucial for a proper task allocation between humans and robots.

In workplaces where humans and robots are working in teams, the tasks have to be allocated in an intelligent way. Chen et al. (2014) show that task scheduling between humans and robots is very difficult because the two agents differ dramatically from each other. Humans and robots can be viewed as a set of heterogeneous processors with different capabilities, working both sequentially and in parallel. A multi-level task allocation model is presented by Malvankar-Mehta and Mehta (2015) to optimize the performance of a multi-robot multi-human system. Tasks are allocated to leaders that distribute the allocated tasks in their human–robot teams based on high-risk and low-risk information at multiple team levels. If such information is present, the model can be used to optimize the HRTA by backward induction. Another popular approach to optimize the human–robot task distribution is to define a cost function for the cost of each task for the worker and the robot and use any optimization function (e.g. a genetic algorithm). Chen et al. (2011, 2012) introduced a detailed cost function for the human and robot, including electricity cost for the robot and labor cost for programming. Using a genetic algorithm, the human–robot task distribution is optimized based on this cost function. Takata and Hirano (2011) present a similar approach, where the total production cost for a task distribution is calculated as the sum of the labor cost and investment cost of the robots. For the estimation of the operation time of each process, methods-time measurement (MTM) is used. They show that using their cost model, human–robot cooperation is superior especially for high probabilities in variant changes. However, estimating the time for collaboration and real interaction between the human and the robot remains difficult. Furthermore, the integration of different costs based on robot capabilities or cost of ergonomic impacts for the worker in human–robot task allocation have not been addressed in the literature. The main disadvantage of the optimization approaches discussed above is that the estimation of the execution time, which is the base for any cost calculation, is considered independent of the distribution and the sequence of the tasks themselves. Dependencies between the tasks, especially additional moving distances and waiting times, have a big influence on the total process time as stated by Chen and Hector (1992). Using a simulation for the evaluation of different task distributions over a whole production cycle therefore allows a better optimization.

As shown by Ding et al. (2014), the base of any optimization of the work organization between humans and robots is the modeling of jobs in a workplace and the estimation of execution times for each task. They present a heuristic method to optimize the human–robot task allocation considering the change of efficiency of a human worker if multiple tasks are executed in parallel. In task allocation for manual car assembly, it is generally assumed that an agent only executes one task at a time, but this highlights an important weakness of current approaches in HRTA: the change in the duration of tasks depends on the task distribution and sequence.

Methodology

In order to assess the suitability of a particular human–robot work distribution, a simulation tool is programmed in MATLAB. The tool uses an assembly procedure model that contains information on existing workplaces from the manual work description database, the shopfloor layout database and the product database to simulate the tasks and movements of the human and the robot in a workplace. Key objectives, such as manufacturing and idle times or walking distances during a production cycle, are calculated during a simulation run.

Procedure model

Based on MTM-UAS (The Industrial Engineer 2016), a new model to describe the assembly procedure is developed. As shown in Fig. 1, it contains all relevant information in order to generate the commands for the correct sequence of movements of the worker and the robot in the simulation. The data is structured hierarchically and combines information from different production databases such as the workplan, the hall layout, or the product database. The workplan database contains information encoded in MTM, where an assembly task consists of a sequence of movements that have to be performed in order to get a task done. Each movement is assigned a code representing a movement category combined with a standardized execution time.

The proposed assembly procedure model (see Fig. 1) consists of multiple tasks that can be executed either by a human or a robot. In order to complete a task, a series of of production steps (skills) must be performed at a particular mounting location of the car. The sequence of the skills within a task is fixed, while the sequence of the tasks is constrained by the precedence matrix in the model. The skills are categorized based on the type of action they represent (e.g. pick, place, handle, wait) and assigned the standard execution time from MTM, extended by the number of repetitions. For simplicity, it is assumed that robots require the same standardized duration as a human to perform an action. The model is extended by information on the ergonomic impact for a human worker, which is also stored in the workplan database. Using links to the tool and part database, information on the objects that are

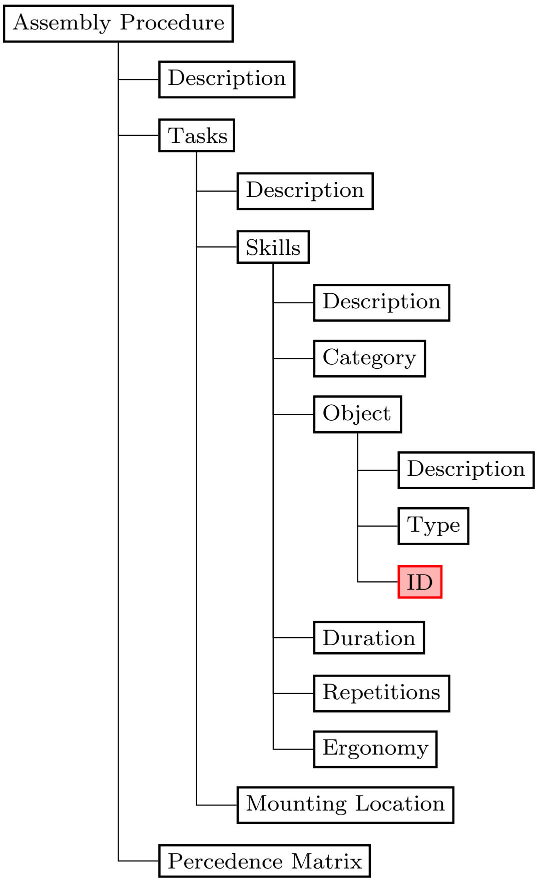

Fig. 1 Diagram showing the assembly procedure model used for the automated generation of action commands for the agents in the simulation. The data can directly imported from different production databases except for the object ID, representing the connection to the objects in the simulation, which has to be entered manually (marked red) (Color figure online)

handled during each skill is stored in the model. The ID of each object is used as a connection between the assembly procedure model and the simulation in order to find the objects in the simulated environment. The ID has to be set manually when initializing the simulation environment, which can be compared to teaching the robot the different positions of tools, parts, and containers when setting up a new workplace.

Simulation principle

A two dimensional representation of a workplace is used to simulate the tasks during a production cycle. Objects like the car, containers, parts, and tools and their interactions with agents (i.e. a worker or robot) are simulated. The simulation environment is set up by importing parameters like the size and position of static objects, the movement speed of the cars, and the dimensions of the workplace from the hall layout database. In the initialization phase, the tasks contained in the assembly procedure model are distributed to the robot and the worker. So each agent is assigned to a particular sequence of tasks that is repeated in each production cycle. During the simulation, the following functions are called in each simulation step:

1. Update environment.
2. Update action commands of agents.
3. Calculate waypoints for agents.
4. Update positions of agents.
5. Move attached objects of agents.
6. Update visualization.

In a first step the environment is updated, for example moving the cars in the direction of the production line, based on which the following calculations are performed. Then, the assembly procedure model is used to generate the action commands for the agents depending on the category, the handled object, and the progress of the active skill. The resulting target destination is then used to find a path and calculate waypoints for the agents. The position update of the agents includes a collision check and a calculation of evasion movements, if necessary. Finally, the attached objects of each agent have to be moved in the same direction as the agent and the visualization has to be updated.

Job execution logic

The job execution logic is the most important part of the simulation and describes the determination of the action commands of the worker and the robot. The basic idea is to use the information from the assembly procedure model and add a target position, a status, and a progress to the skill that is being executed. Based on this information, the destination of the agent and where it has to move in each simulation step is updated. The procedure of the action command generation is shown in Fig. 2.

– In a first step, it is checked if a task is currently allocated to the agent and which skill of the task is currently active. If all tasks allocated to the agent are finished, the agent moves to its initial position to wait for the next production cycle.
– For an active skill, the target position for execution must first be determined. The target position is the point where a skill actually can be performed and has to be calculated dynamically. To perform a pick skill for example, the agent has to be in reaching distance of the container, where the part or the tool has to be picked. For a place skill, it has to be distinguished between placing a part (i.e. assembling the part on the car) or placing a tool (i.e. place it back to the storage position). The calculation of the destination is therefore a function of the skill category

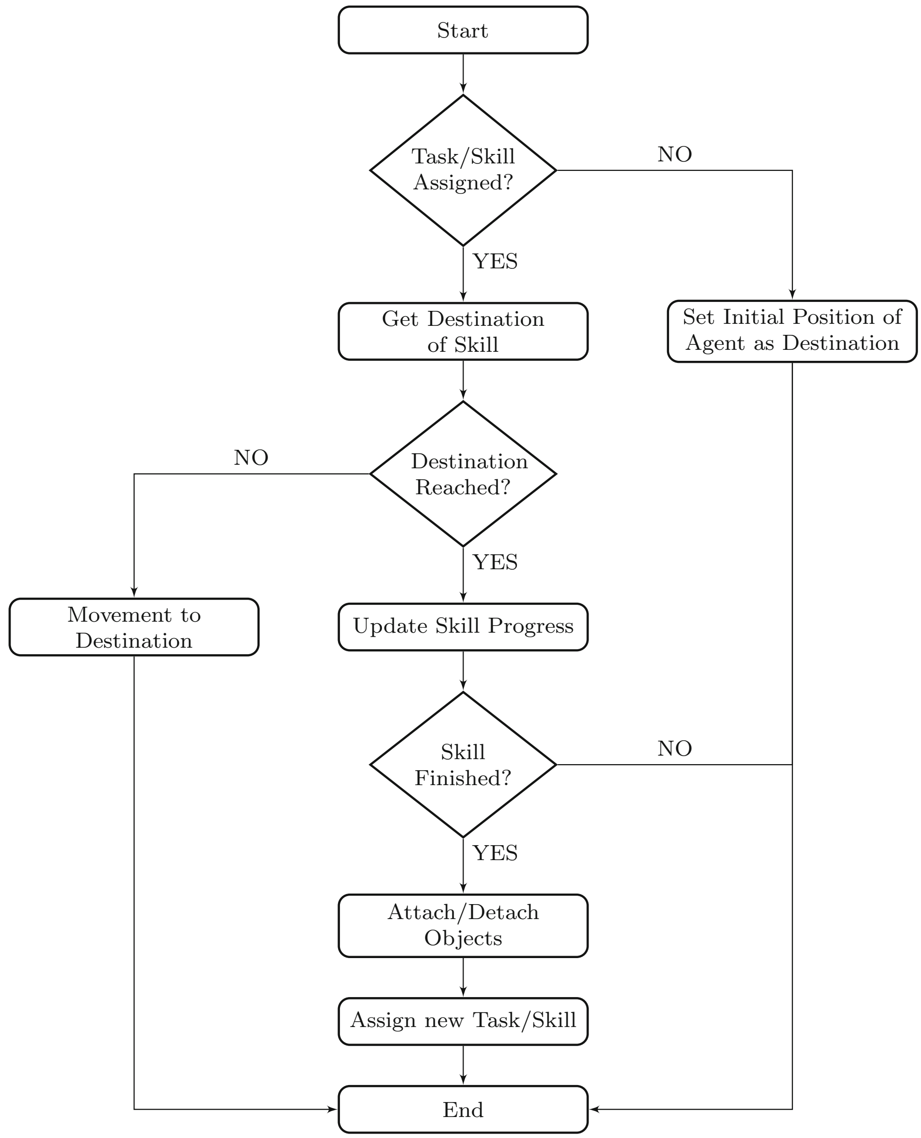

Fig. 2 Diagram showing the job execution logic for an agent

and the affected object type of the skill. It can either be the mounting location or a storage location. The mounting location of a skill is stored in the task properties, since there is only one mounting location for a whole task. The storage position of an object (i.e. a tool, part) can be found by checking the contents of all containers in the layout recursively and returning the position of the container with the required object of the active skill. It can be seen that using the proposed assembly procedure model, the determination of the target position becomes a simple combination of the category and the object of the skill. For the three most important skills (i.e. pick, place, handle) in combination with the objects tool, part, and container, the target positions are shown in Table 2.

– After updating the target position of the agent, the distance to the target position is calculated and the skill status is set accordingly. In order to perform a skill, the agent has to be in reaching distance of the object. If the target position is not yet reached, the job execution is terminated and the movement calculation is started with the newly set destination. If the target position is reached, the progress of the skill is increased relative to the duration of the skill. The duration of a skill represents the time needed to perform the movements of a skill after the destination has been reached.
– If a skill is completed, the affected object has to be attached or detached, depending on the skill category. For a pick skill, the object has to be detached from the container and attached to the agent. The same holds true for a place skill, where the affected object has to be detached from the agent and attached to the car or the container, depending on where the object was placed.

Table 2 Skill destination depending on the skill category and affected object

Destination		Affected object of skill		
		Tool	Part	Container
Category of skill	Pick	Position of corresponding container	Position of corresponding container	Position of container
	Place	Position of corresponding container	Mounting position on car	Initial position of container
	Handle	Mounting position on car	Mounting position on car	Mounting position on car

Movement calculation

After the job execution has been performed and the destination for the agents are set, the actual movements must be calculated and the position of the agents must be updated. This allows the consideration of dependencies between the worker and the robot in their movement paths, like collision avoidance movements. The calculation of the movements of the worker and the robot is divided in two parts: a global path finding and a local collision avoidance. Based on the layout, waypoints are generated for the agent to reach the target position defined by the previous action command generation. In each simulation step, the planned movement to the next waypoint is checked for collisions and updated, based on the current position and speed of the agent.

The generation of waypoints to reach the defined destination in minimal time is done via the generation of a navigation mesh from the simulation layout and a shortest path search from the agent's distance to the skill destination, as described by Cui and Shi (2012). After the waypoint generation, the actual movements of the worker and the robot have to be calculated. The generation of realistic movements in the simulation is of high importance, since they have a big influence on the whole production cycle. Therefore, the movement in route to the next waypoint, implemented by a proportional controller, is combined with a collision avoidance movement, implemented by a potential fields controller taking into account all other objects in the layout, as presented by Khatib (1986). The implementation of well-established methods from the field of robotics for the movement calculation (navigation and collision avoidance) allows the generation of realistic behaviors of the worker and the robot.

Visualization

In order to follow and interpret the results from the simulation, a visualization is built that shows the layout with the objects and the movements of the agents (i.e. the worker and the robot), as shown in Fig. 3.

The cars are represented by black polygons that move in the direction of the line during a cycle. The start and end of a cycle are indicated by vertical dashed lines in the layout. The containers, where tools, parts, and smaller containers are stored, are indicated by gray rectangles in the layout. The worker is represented as a blue circle with the reaction distance indicated as a dashed line, whereas the robot is shown in green. Additional information, such as the planned waypoints of the worker and the robot or collision avoidance movements, can be displayed.

Experiment

In the previous section, the main functionalities of the simulation were explained in detail. Physical objects and dependencies are modeled so that the worker and the robot show a realistic behavior. In order to get reliable results, the simulation tool has to be validated with a real workplace.

Workplace

An actual workplace of the final assembly line of the Golf 7 at the Volkswagen plant in Wolfsburg (Germany) is taken for the validation of the simulation tool. At the selected workplace, three tasks are performed during a cycle time of one minute:

1. Screw and check a fixation in the motor area
2. Assemble a body protection part with screws
3. Mount multiple clips in the tire area

In addition to the information from the hall layout database, measures are taken from the workplace in the assembly line to further specify the layout from the database and build an exact model of the real workplace. The updated layout is shown in Fig. 3.

Measurements

For the validation of the simulation tool, the workplace is simulated without a robot so that the worker has to perform all tasks, which represents the actual state in the assembly line. In order to analyze the movements of the worker under real

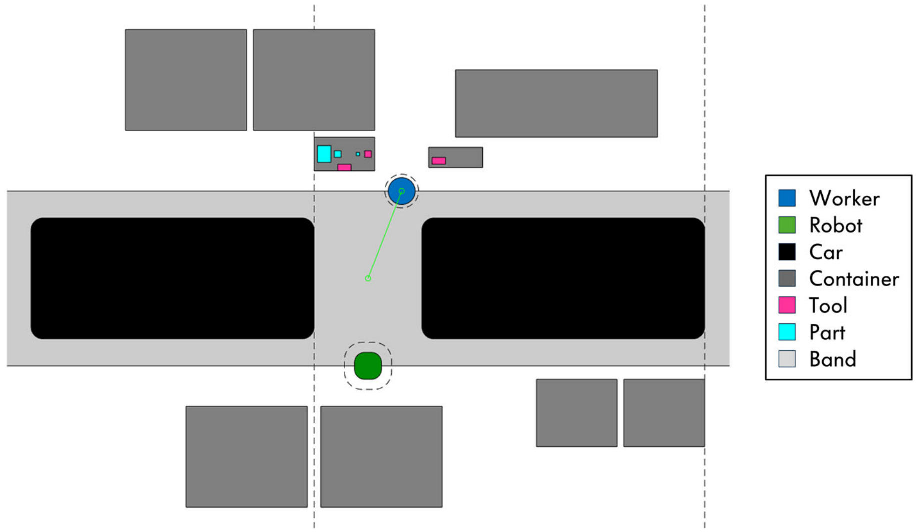

Fig. 3 Visualization of a workplace in the simulation (Bänziger et al. 2018)

circumstances, the workplace is rebuilt in a test environment. The time needed for the completion of each movement is measured during three production cycles in order to be able to calculate a mean time and a standard deviation for each measurement. The time for the individual skills and also the total time needed for the completion of the three tasks are then compared to the results from the simulation. Additionally, the positions of the worker are marked on the floor and the total walking distance is compared to the path generated by the simulation.

Optimization

As explained in the introduction, tasks can be allocated either to the human or the robot and can be executed in different sequences, which results in a high number of different work organizations for a workplace with human–robot cooperation. For example a workplace with six tasks ($n_t = 6$) leads to

$$n_d = 2^{n_t} = 2^6 = 64 \tag{1}$$

possible work distributions, which then have to be ordered in a sequence. For the case that two tasks are distributed to the worker ($n_{t,w} = 2$) and four tasks to the robot ($n_{t,r} = 4$),

$$n_{seq} = n_{t,w}! \cdot n_{t,r}! = 2! \cdot 4! = 48 \tag{2}$$

different sequences can be evaluated. Summing up all the possible sequences for the 64 different work distributions, theoretically 5040 scenarios have to be evaluated in total.

It is clear that the work organization highly influences the whole production cycle, since there are a lot of dependencies and interactions between the agents and the objects in the workplace. Therefore, different work organizations have to be analyzed and suitable task distributions and sequences must be identified. In order to maximize the efficiency (i.e. time and cost) and minimize ergonomic impacts for the worker, the simulation tool is used in a two-layered genetic algorithm (GA) to find an optimized work organization between a worker and a robot, as shown in Fig. 4. The optimization principle allows the separation of the two optimization problems: the task distribution and the task sequence optimization. Both optimization problems can therefore be solved independently, which increases transparency and makes the results more interpretable.

Design of genetic algorithm

A genetic algorithm is used in both optimization problems, the task distribution and the task sequencing. For the task distribution, a chromosome is represented by a binary vector (D) with length equal to the number of tasks of the workplace,

$$D = [1\ 0\ 1\ 0\ 0\ 1],$$

where 1 means the task is assigned to the worker and 0 to the robot. After the assignment, the robot tasks are checked if they match the robot capabilities. Some tasks (e.g. visual quality check) are referred to the human or can not be executed by a robot (e.g. handling limp cables). If a skill or task cannot be executed by the robot, zero is returned as fitness for the given distribution.

The optimization of the sequence is solved as sub-problem for a given task distribution. From the distribution, all possible sequences for the worker and the robot are calculated and

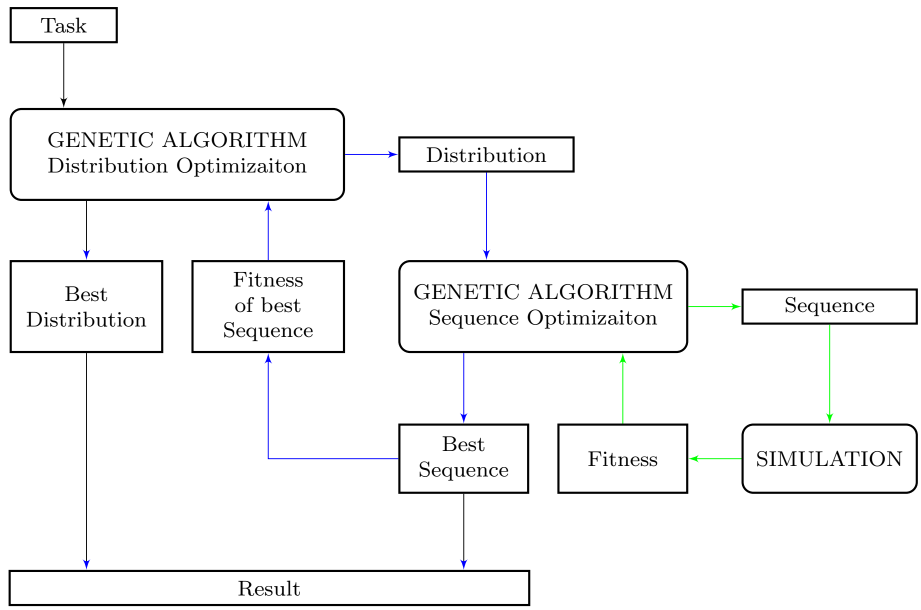

Fig. 4 Principle for the optimization of the work organization. Independent optimization of the task distribution (blue) and the task sequence (green) in two layers. Function blocks are shown in rounded boxes, parameters are shown in square boxes (Color figure online)

Table 3 Generation of all possible task sequences

Worker		Robot	
i_w	Sequence	i_r	Sequence
1	[1 3 6]	1	[2 4 5]
2	[1 6 3]	2	[2 5 4]
…	…	…	…
$n_{seq,w}$	[6 3 1]	$n_{seq,r}$	[5 4 2]

numbered, as indicated in Table 3. The chromosome for the sequence optimization (S) is defined as a two dimensional vector with integer values (i),

$$S = [i_w \quad i_r],$$

with conditions

$$\{i_w \in \aleph | 1 \leq i \leq n_{s,w}\} \quad and \quad \{i_r \in \aleph | 1 \leq i \leq n_{s,r}\},$$

where the first value (i_w) represents the sequence number for the worker and the second value (i_r) the sequence number for the robot of a total number of sequences $(n_{seq,w}, n_{seq,r})$. The fitness of the best sequence for a given task distribution is then returned as fitness to the task distribution optimization algorithm. The implementation of a genetic algorithm in MATLAB is used for both optimization problems, the task distribution and the task sequencing between human and robot. For the optimization of the task distribution, the population size is set to 6 and the crossover fraction is set to 0.65. For the task sequence optimization, the population size is set to 8 and the crossover fraction is set to 0.4.

Definition of fitness function

The simulation tool is used to evaluate different task sequences of a given distribution. The fitness function is therefore defined as

$$f = w_p \cdot p_n + w_t \cdot t_n + w_s \cdot s_n, \tag{3}$$

with the normalized key objectives task progress (p_n), waiting time (t_n), traveled distance (s_n), which are calculated in the simulation, and the corresponding weights (w). The weights are introduced to allow the user to influence the optimization of the task allocation. They can vary depending on the type of production system the workplace is planned for. For example in automotive assembly lines, the task progress within a sequence (e.g. 60 s) is very important, whereas in island production the walking distance is of higher importance.

The normalized task progress,

$$p_n = \frac{n_{sk,cpl}}{n_{sk,tot}}, \tag{4}$$

is defined as the ratio of the number of completed skills $(n_{sk,cpl})$ to the total number of skills in a workplace $(n_{sk,tot})$. The distance is calculated as the sum of the traveled distance of the worker and the robot (s_w, s_r) and normalized by the length of a cycle (s_c),

$$s_n = \frac{s_c}{s_c + s_w + s_r}. \tag{5}$$

The waiting time is calculated as the difference between the times the worker and the robot need to finish their tasks (t_w, t_r), normalized by the half of the cycle time (t_c),

$$t_n = \frac{t_c}{t_c + |t_w - t_r|}. \tag{6}$$

For the optimization of the workplace described in "Optimization of the human–robot work distribution" section, the weights are chosen at

$$w_p = 0.5, \quad w_t = 0.1, \quad \text{and} \quad w_s = 0.4$$

in order to make sure that all tasks are finished in one assembly cycle. Furthermore, the traveled distance is weighted higher than the waiting time in order to prevent large collision avoidance movements.

Results

Validation of the simulation tool

The comparison of the results from the simulation experiment and the recorded data from the assembly line setup can be found in Tables 4 and 5. It can be seen that the difference of total time needed for the execution of the tasks is less than a second, while the difference in the total walking distance is 3.8 m. The prediction of the total assembly time is very accurate with a deviation of only 0.7% for the tested workplace. It can be seen that the highest errors in time occur in so-called value adding actions of the worker, the screwing in the first two tasks and the mounting of the clips in the third task. The positioning of the screws and the screwdriver and the actual screwing of the part takes less time in reality than predicted by the simulation based on the MTM data. One explanation for this effect is that there are no standard times in MTM for processes like screwing. For the workplan, an average execution time is measured in test environments, but in reality the worker is able to perform the action much faster using his or her sensory and multitasking capabilities. While walking to the assembly location, the worker can already preposition the screws on the part. On the other side, the picking up of tools and parts at the beginning of the tasks takes longer in real assembly lines than in the simulation so that a good approximation of the real assembly time for a production cycle is achieved.

The estimation of the walking distance is not as accurate as the time prediction. For the given workplace, the relative error of the calculated walking distance is 37.2%. It can be seen that for longer walking paths like in the first task, the distance prediction is still accurate with a relative error of 4.1%. For tasks with very short walking distances, the error becomes larger. Since the second and third task require only short displacements of the worker, there is a big difference in the predicted distance and the actual walked distance of the worker in the experiment. This can be explained by additional steps the worker has to perform in order to adjust his

Table 4 Difference between the time measurement and the simulation result in seconds

Task	Skill	Terminated skill	Meas. mean		Meas. σ		Simulation		Difference	
			Incr.	Total	Incr.	Total	Incr.	Total	Incr.	Total
0	1	Initial position	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	1	Pick tool	1.76	1.76	0.23	0.23	1.88	1.88	0.12	0.12
1	2	Walk to mounting location	3.63	5.39	0.21	0.27	2.95	4.83	−0.68	−0.56
1	3	Screw	6.76	12.15	0.47	0.38	8.74	13.57	1.98	1.42
1	4	Check	2.02	14.17	0.73	0.38	0.61	14.18	−1.41	0.01
1	5	Put back tool	3.10	17.27	0.82	0.45	2.88	17.06	−0.22	−0.21
2	1	Walk to container	3.15	20.42	0.41	0.68	2.63	19.69	−0.52	−0.73
2	2	Pick tool and parts	2.92	23.34	0.14	0.68	3.63	23.32	0.71	−0.02
2	3	Walk to mounting location	3.15	26.50	0.48	0.89	2.47	25.79	−0.68	−0.71
2	4	Screw	9.02	35.52	0.57	0.59	15.63	41.42	6.61	5.90
2	5	Put back tool	3.41	38.93	0.57	0.93	1.25	42.67	−2.16	3.74
3	1	Pick up tool and parts	4.05	42.98	1.63	1.80	2.51	45.18	−1.54	2.20
3	2	Walk to mounting location	3.01	45.99	0.67	2.45	0.43	45.61	−2.58	−0.38
3	3	Mount clips	7.49	53.48	0.32	2.61	10.09	55.70	2.60	2.22
3	4	Put back tool	2.77	56.24	0.09	2.54	1.29	56.99	−1.48	0.75
4	1	Initial position	1.61	57.85	0.39	2.36	0.51	57.50	−1.10	−0.35

Table 5 Difference between the time measurement and the simulation result in meters

Task	Skill	Terminated skill	Measurement		Simulation		Difference	
			Incr.	Total	Incr.	Total	Incr.	Total
0	1	Initial position	0.00	0.00	0.00	0.00	0.00	0.00
1	1	Pick tool	1.00	1.00	0.51	0.51	− 0.49	− 0.49
1	2	Walk to mounting location	1.00	2.00	1.48	1.99	0.48	− 0.01
1	3	Screw	0.65	2.65	0.81	2.80	0.16	0.15
1	4	Check	0.10	2.75	0.06	2.86	− 0.04	0.11
1	5	Put back tool	1.30	4.05	1.03	3.89	− 0.27	− 0.16
2	1	Walk to container	1.70	5.75	1.33	5.22	− 0.37	− 0.53
2	2	Pick tool and parts	0.00	5.75	0.29	5.51	0.29	− 0.24
2	3	Walk to mounting location	2.00	7.75	1.17	6.68	− 0.83	− 1.07
2	4	Screw	1.10	8.85	1.24	7.92	0.14	− 0.93
2	5	Put back tool	1.10	9.95	0.18	8.10	− 0.92	− 1.85
3	1	Pick up tool and parts	0.00	9.95	0.13	8.23	0.13	− 1.72
3	2	Walk to mounting location	0.75	10.70	0.08	8.31	− 0.67	− 2.39
3	3	Mount clips	0.95	11.65	0.81	9.12	− 0.14	− 2.53
3	4	Put back tool	1.20	12.85	0.31	9.43	− 0.89	− 3.42
4	1	Initial position	1.00	13.85	0.67	10.10	− 0.33	− 3.75

orientation relative to the car for different assembly steps, which are not modeled in the simulation.

Optimization of the human–robot work distribution

The same workplace is used in the optimization part to find an optimized human–robot work organization for a potential implementation of a mobile assistance robot. However, the tasks are slightly modified in order to increase the distribution and sequence flexibility. The three tasks are split up in five smaller sub-tasks in order to allow human–robot interaction, which means the worker and the robot are working on the same main task. Finally, an additional task is added to allow the small storage table to be moved, like it is originally scheduled in the workplan.

1. Handle the movable table.
2. Screw the fixation of an air pipe in the motor area.
3. Visually check the fixation.
4. Position the body protection part in the tire area.
5. Fix the body protection part with screws.
6. Mount three clips in tire area using a hammer.

An optimized work organization that is found using the simulation tool in the presented genetic algorithm is

$$W = [1\ 6\ 3] \quad \text{and} \quad R = [5\ 2\ 4].$$

In this constellation, the worker is guiding the movable table in order to minimize his or her walking distance. The robot starts by picking the body protection part from a container to position it in the tire area. The worker then picks the screws from the movable table to fix the prepositioned body protection part. In this time, the robot is screwing the air pipe fixation in the motorblock area. As soon as the robot has finished the task, the worker can check the fixation, while the robot collects three clips from the table to assemble them to the car.

The total time for the worker and the robot to complete all tasks is 43.0 s. When compared to the first case, in which the worker is performing all tasks on his or her own in 57.5 s, the task distribution and sequence is optimized. Waiting times and walking distance are minimized while constraints, like the check task being reserved for the human, are fulfilled.

The fitness score for the presented solution is 0.733. In total 566 function calls were executed during the evaluation of 3 generations. The optimization terminated after a change of less than 0.005 in the last two generations of the task distribution. The majority of the sequence optimizations also converged after three generations.

Conclusion and outlook

In this paper, a new approach to optimize the human–robot work organization, based on genetic algorithms, is presented. A detailed simulation tool is developed and first used as an objective function for the human–robot task allocation problem.

In the first part, it is shown that a simulation based on workplan and layout data can be used to model complex dependencies between objects like tools, parts, containers, and products in automotive assembly lines. The integration of a physical model of a human worker and a robot in the simulation allows the accurate prediction of total manufacturing times. This also shows that the MTM-based times can be used for the accurate prediction of assembly steps and therefore suitable as base for the simulation and optimization of workplaces. Since every workplace is already described in MTM at the Volkswagen plant, no additional data must be collected. However, errors in the prediction of process times of single skills still exist. Therefore, the standardized times for each skill have to be adjusted for the simulation by intelligent algorithms, which is part of future work. The standardized times will be adjusted for robotic skills, since robots have different capabilities than human workers. For the more accurate prediction of the distances, the pose of the worker and the approach direction for different tasks have to be included in the simulation in the future.

In the second part of the paper, a new method to optimize the human–robot task allocation is presented. The method uses a two layered genetic algorithm and a human–robot simulation to optimize the task distribution and the task sequence of a worker and a robot in the same workplace. The advantage of using a two-layered optimization principle is that the standard implementation of the genetic algorithm in MATLAB can be used. It is shown that using the proposed method, the work organization for an existing workplace can be optimized and an intelligent task allocation can be found. In contrast to methods known from the literature, the simulation based calculation of the fitness value in the optimization leads to more realistic results and is more flexible since different objectives can be considered. Previous methods, such as presented by Chen et al. (2012), use a simple cost function to calculate the assembly cost, based on an average labor time of the worker and the needed time to program the robot, which is not capable of considering interactions between the robot and the worker in the same workspace. Furthermore, the use of a simulation allows the modeling of the effects of the environment on the walking distances and the execution times, as shown in Fig. 3. Methods, such as presented by Ding et al. (2014), mainly depend on the precedence graph, so the influence of the layout of the workplace and the influence of the task order on the execution times is not considered. In the simulation, these effects are modeled by the realistic movement of the worker and the robot, which leads to a more accurate estimation of the assembly time, as shown in Table 4.

It is clear that the presented optimization approach cannot guarantee optimality, but for the presented workplace an intuitive solution was found with a significantly reduced number of function calls. Also for large scale scenarios, an optimized task allocation can be generated in reasonable time. This shows that a two-layered optimization approach is suitable for the described problem of human–robot task allocation in automotive assembly lines, but there is also a lot of potential for improvement. In future works, more data will be used for the calculation of the fitness function, for example the ergonomic points for single skills that are allocated to the worker or the detailed automation potential of skills that are allocated to the robot. Additionally, more sophisticated implementation of genetic algorithms will be used, especially for the task scheduling problem that is very similar to the traveling salesman problem. Since the two optimization problems are separated in the presented approach, as shown in Fig. 4, different optimization functions can be used in the future to overcome the reduced local search capabilities of genetic algorithms.

In todays automotive assembly lines, human–robot cooperation is implemented in already existing workplaces. In order to assess the whole potential of humans and robots working in teams, multiple workplaces have to be combined and the task allocation has to be optimized globally, as stated by Bochmann et al. (2017). Therefore, the simulation has also to be tested with more complex workplaces with multiple robots and workers in the same workspace for future applications.

As a next step towards smart manufacturing, the simulation tool needs to be integrated in a service-oriented production system, as proposed by Tao and Qi (2017). Using new information technologies and integration frameworks, such as presented by Tao et al. (2018), the information of the workplan and the hall layout can directly be used as input for the optimization of the task allocation. The output of the simulation, the skills allocated to the robot, will be used in a skill based control framework to automatically program mobile assistant robots. This allows the flexible use of smart assistant systems in assembly workplaces. Since the presented procedure model depends solely on information from the production database, which is available for every workplace in the factory, data-driven manufacturing is enabled by connecting the presented planning and optimization tool with the production database and the robots on the shopfloor.

References

Angerer, S., Strassmair, C., Staehr, M., Roettenbacher, M., & Robertson, N. M. (2012). Give me a hand: The potential of mobile assistive robots in automotive logistics and assembly applications. In *Proceedings of the IEEE conference on technologies for practical robot applications* (pp. 111–116). https://doi.org/10.1109/TePRA.2012.6215663.

Bänziger, T., Kunz, A., & Wegener, K. (2018). *Simulation of the human–robot cooperation in automotive assembly lines based on MTM*. https://doi.org/10.6084/m9.figshare.5743314. https://figshare.com/s/35b5a5d311f6930ec3f3. Accessed January 1, 2018.

Bochmann, L., Bänziger, T., Kunz, A., & Wegener, K. (2017). Human–robot collaboration in decentralized manufacturing systems: An approach for simulation-based evaluation of future intelligent production. In *CIRP conference on intelligent computation in manufacturing engineering* (Vol. 62, pp. 624–629). https://doi.org/10.1016/j.procir.2016.06.021.

Chen, F., Sekiyama, K., Cannella, F., & Fukuda, T. (2014). Optimal subtask allocation for human and robot collaboration within hybrid assembly system. In *Proceedings of the IEEE transactions on automation science and engineering* (pp. 1065–1075). https://doi.org/10.1109/TASE.2013.2274099.

Chen, F., Sekiyama, K., & Fukuda, T. (2012). A genetic algorithm for subtask allocation within human and robot coordinated assembly. In *International symposium on micro-nanomechatronics and human science* (MHS) (pp. 507–511). https://doi.org/10.1109/MHS.2012.6492504.

Chen, F., Sekiyama, K., Sasaki, H., Huang, J., Sun, B., & Fukuda, T. (2011). Assembly strategy modeling and selection for human and robot coordinated cell assembly. In *IEEE/RSJ international conference on intelligent robots and systems* (IROS) (pp. 4670–4675). https://doi.org/10.1109/IROS.2011.6094715.

Chen, H. G., & Hector, G. (1992). A rule-based robot scheduling system for flexible manufacturing cells. *Journal of Intelligent Manufacturing*, *3*, 285–296. https://doi.org/10.1007/BF01577270.

Cherubini, A., Passama, R., Crosnier, A., Lasnier, A., & Fraisse, P. (2016). Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, *40*, 1–13. https://doi.org/10.1016/j.rcim.2015.12.007.

Cui, X., & Shi, H. (2012). An overview of pathfinding innavigation mesh. *International Journal of Computer Science and Network Security*, *12*(12), 48–51.

Dahl, T. S., Matarić, M., & Sukhatme, G. S. (2009). Multi-robot task allocation through vacancy chain scheduling. *Robotics and Autonomous Systems*, *57*(6–7), 674–687. https://doi.org/10.1016/j.robot.2008.12.001.

Das, G. P., McGinnity, T. M., Coleman, S. A., & Behera, L. (2015). A distributed task allocation algorithm for a multi-robot system in healthcare facilities. *Journal of Intelligent & Robotic Systems*, *80*(1), 33–58. https://doi.org/10.1007/s10846-014-0154-2.

de Gea Fernández, J., Mronga, D., Günther, M., Knobloch, T., Wirkus, M., Schröer, M., et al. (2017). Multimodal sensor-based whole-body control for human–robot collaboration in industrial settings. *Robotics and Autonomous Systems*, *94*, 102–119. https://doi.org/10.1016/j.robot.2017.04.007.

de Gea Fernández, J., Mronga, D., Günther, M., Wirkus, M., Schröer, M., Stiene, S., et al. (2017). iMRK: Demonstrator for intelligent and intuitive human-robot collaboration in industrial manufacturing. *KI: Künstliche Intelligenz*, *31*(2), 203–207. https://doi.org/10.1007/s13218-016-0481-5.

Ding, H., Schipper, M., & Matthias, B. (2014). Optimized task distribution for industrial assembly in mixed human-robot environments—case study on IO module assembly. In *Proceedings of the IEEE international conference on automation science and engineering* (pp. 19–24). https://doi.org/10.1109/CoASE.2014.6899298.

Evans, P., & Annunziata, M. (2012). *Industrial internet: Pushing the boundaries of minds and machines*. https://www.ge.com/docs/chapters/Industrial_Internet.pdf. Accessed November 26, 2012.

German Organization for Standardization. (2008). *DIN EN ISO 13849: Safety of machinery—Safety-related parts of control systems—Part 1: General principles for design*.

German Organization for Standardization. (2010). *DIN EN ISO 13855: Safety of machinery—Positioning of safeguards with respect to the approach speeds of parts of the human body*.

German Organization for Standardization. (2011). *DIN EN ISO 12100: Safety of machinery—General principles for design—Risk assessment and risk reduction*.

German Organization for Standardization. (2012). *DIN EN ISO 10218: Robots and robotic devices—Safety requirements for industrial robots*.

Ghosh, B. K., & Helander, M. G. (1986). A systems approach to task allocation of human–robot interaction in manufacturing. *Journal of Manufacturing Systems*, *5*(1), 41–49. https://doi.org/10.1016/0278-6125(86)90066-X.

Haddadin, S., Parusel, S., Belder, R., Albu-Schaffer, A., & Hirzinger, G. (2011). Safe acting and manipulation in human environments: A key concept for robots in our society. In *Proceedings of the IEEE workshop on advanced robotics and its social impacts* (pp. 72–75). https://doi.org/10.1109/ARSO.2011.6301962.

Hengstebeck, A. (2015). Formal modelling of manual work processes for the application of industrial service robotics. In *Proceedings of the CIRP conference on manufacturing systems* (pp. 364–369). https://doi.org/10.1016/j.procir.2015.12.013.

International Electrotechnical Commission. (2010). *IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems—Part 1: General requirements*.

International Organization for Standardization. (2016). *ISO TS 15066: Robots and robotic devices—Collaborative robots*.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, *5*, 90–98. https://doi.org/10.1109/ROBOT.1985.1087247.

Koren, Y. (2010). *The global manufacturing revolution: Product–process–business integration and reconfigurable systems*. Hoboken, NJ: Wiley.

Krüger, J., Lien, T. K., & Verl, A. (2009). Cooperation of human and machines in assembly lines. *CIRP Annals: Manufacturing Technology*, *58*(2), 628–646. https://doi.org/10.1016/j.cirp.2009.09.009.

Kusiak, A. (2017). Smart manufacturing must embrace big data. *Nature*, *544*, 23–25. https://doi.org/10.1038/544023a.

Loredo-Flores, A., Gonzalez-Galvan, E. J., Cervantes-Sanchez, J. J., & Martinez-Soto, A. (2008). Optimization of industrial, vision-based, intuitively generated robot point-allocating tasks using genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(4), 600–608. https://doi.org/10.1109/TSMCC.2008.923886.

Malvankar-Mehta, M. S., & Mehta, S. S. (2015). Optimal task allocation in multi-human multi-robot interaction. *Optimization Letters*, *9*(8), 1787–1803. https://doi.org/10.1007/s11590-015-0890-7.

Michalos, G., Makris, S., Spiliotopoulos, J., Misios, I., Tsarouchi, P., & Chryssolouris, G. (2014). ROBO-PARTNER: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future. *Procedia CIRP*, *23*, 71–76. https://doi.org/10.1016/j.procir.2014.10.079.

Morioka, M., & Sakakibara, S. (2010). A new cell production assembly system with human–robot cooperation. *CIRP Annals: Manufacturing Technology*, *59*(1), 9–12. https://doi.org/10.1016/j.cirp.2010.03.044.

Müller, R., Vette, M., & Scholer, M. (2014). Inspector robot—A new collaborative testing system designed for the automotive final assembly line. *Procedia CIRP*, *23*, 59–64. https://doi.org/10.1016/j.procir.2014.10.093.

Nagarajan, T., & Thondiyath, A. (2014). An algorithm for cooperative task allocation in scalable, constrained multiple robot systems. *Intelligent Service Robotics*, *7*(4), 221–233. https://doi.org/10.1007/s11370-014-0154-x.

Nielsen, I., Dang, Q. V., Bocewicz, G., & Banaszak, Z. (2017). A methodology for implementation of mobile robot in adaptive manufacturing environments. *Journal of Intelligent Manufacturing*, *28*(5), 1171–1188. https://doi.org/10.1007/s10845-015-1072-2.

Shen, Y., Reinhart, G., & Tseng, M. M. (2015). A design approach for incorporating task coordination for human-robot-coexistence within assembly systems. In *Proceedings of the annual IEEE inter-*

national systems conference (Vol. 9, pp. 426–431). https://doi.org/10.1109/SYSCON.2015.7116788.

Shi, Z., Wei, J., Wei, X., Tan, K., & Wang, Z. (2010). Task allocation model based on reputation for the heterogeneous multi-robot collaboration system. *World Congress on Intelligent Control and Automation (WCICA)*, *8*, 6642–6647. https://doi.org/10.1109/WCICA.2010.5554165.

Stenmark, M., & Malec, J. (2014). Describing constraint-based assembly tasks in unstructured natural language. In *Proceedings of the 19th world congress of the international federation of automatic control* (Vol. 19, pp. 3056–3061). https://doi.org/10.3182/20140824-6-ZA-1003.02062.

Takata, S., & Hirano, T. (2011). Human and robot allocation method for hybrid assembly systems. *CIRP Annals: Manufacturing Technology*, *60*(1), 9–12. https://doi.org/10.1016/j.cirp.2011.03.128.

Tao, F., & Qi, Q. (2017). New it driven service-oriented smart manufacturing: Framework and characteristics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *99*, 1–11. https://doi.org/10.1109/TSMC.2017.2723764.

Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*,. https://doi.org/10.1016/j.jmsy.2018.01.006.

Teiwes, J., Bänziger, T., Kunz, A., & Wegener, K. (2016). Identifying the potential of human–robot collaboration in automotive assembly lines using a standardised work description. *IEEE International Conference on Automation and Computing (ICAC)*, *22*, 78–83. https://doi.org/10.1109/IConAC.2016.7604898.

The Industrial Engineer: MTM—Universal Analysis System (UAS). (2016). http://www.der-wirtschaftsingenieur.de/index.php/mtm-universelles-analysier-system/. Accessed January 7, 2016.

Volkswagen, A. G. (2016). *Golf production at Wolfsburg plant: People and robots work hand-in-hand*. http://www.volkswagen-media-services.com/en/detailpage/-/detail/Golf-production-at-Wolfsburg-plant-people-and-robots-work-hand-in-hand/view/3613525/. Accessed August 1, 2016.

Zacharia, P., & Aspragathos, N. A. (2005). Optimal robot task scheduling based on genetic algorithms. *Robotics and Computer-Integrated Manufacturing*, *21*(1), 67–79. https://doi.org/10.1016/j.rcim.2004.04.003.